

Scilab Textbook Companion for  
Power System Engineering  
by S. Chakraborty, Gupta and Bhatnagar<sup>1</sup>

Created by  
Kavan A B  
B.E

Electrical Engineering

SRI JAYACHAMARAJENDRA COLLEGE OF ENGINEERING  
College Teacher

None

Cross-Checked by  
Reshma

July 31, 2019

<sup>1</sup>Funded by a grant from the National Mission on Education through ICT,  
<http://spoken-tutorial.org/NMEICT-Intro>. This Textbook Companion and Scilab  
codes written in it can be downloaded from the "Textbook Companion Project"  
section at the website <http://scilab.in>

# **Book Description**

**Title:** Power System Engineering

**Author:** S. Chakraborty, Gupta and Bhatnagar

**Publisher:** D. Rai

**Edition:** 2

**Year:** 2013

**ISBN:** 978-8177000207

Scilab numbering policy used in this document and the relation to the above book.

**Exa** Example (Solved example)

**Eqn** Equation (Particular equation of the above book)

**AP** Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

# Contents

List of Scilab Codes	4
<b>2 THERMAL STATIONS</b>	<b>5</b>
<b>3 HYDRO ELECTRIC STATIONS</b>	<b>10</b>
<b>7 TARIFFS AND ECONOMIC ASPECTS IN POWER GENERATION</b>	<b>14</b>
<b>9 CONSTANTS OF OVERHEAD TRANSMISSION LINES</b>	<b>72</b>
<b>10 STEADY STATE CHARACTERISTICS AND PERFORMANCE OF TRANSMISSION LINES</b>	<b>110</b>
<b>11 OVERHEAD LINE INSULATORS</b>	<b>169</b>
<b>12 MECHANICAL DESIGN OF OVERHEAD LINES</b>	<b>181</b>
<b>13 INTERFERENCE OF POWER LINES WITH NEIGHBOURING COMMUNICATION CIRCUITS</b>	<b>197</b>
<b>14 UNDERGROUND CABLES</b>	<b>202</b>
<b>15 CORONA</b>	<b>220</b>

<b>16 LOAD FLOW STUDY USING COMPUTER TECHNIQUES</b>	<b>233</b>
<b>17 POWER SYSTEM STABILITY</b>	<b>247</b>
<b>18 LOAD FREQUENCY CONTROL AND LOAD SHARING OF POWER GENERATING SOURCES</b>	<b>287</b>
<b>20 WAVE PROPAGATION ON TRANSMISSION LINES</b>	<b>312</b>
<b>21 LIGHTNING AND PROTECTION AGAINST OVERVOLTAGES DUE TO LIGHTNING</b>	<b>317</b>
<b>22 INSULATION COORDINATION</b>	<b>322</b>
<b>23 POWER SYSTEM GROUNDING</b>	<b>326</b>
<b>24 ELECTRIC POWER SUPPLY SYSTEMS</b>	<b>328</b>
<b>25 POWER DISTRIBUTION SYSTEMS</b>	<b>341</b>
<b>27 SYMMETRICAL SHORT CIRCUIT CAPACITY CALCULATIONS</b>	<b>358</b>
<b>28 FAULT LIMITING REACTORS</b>	<b>384</b>
<b>29 SYMMETRICAL COMPONENTS ANALYSIS</b>	<b>391</b>
<b>30 UNSYMMETRICAL FAULTS IN POWER SYSTEMS</b>	<b>408</b>
<b>32 CIRCUIT BREAKER</b>	<b>444</b>
<b>33 PROTECTIVE RELAYS</b>	<b>451</b>
<b>34 PROTECTION OF ALTERNATORS AND AC MOTORS</b>	<b>458</b>
<b>35 PROTECTION OF TRANSFORMERS</b>	<b>469</b>
<b>36 PROTECTION OF TRANSMISSION LINE SHUNT INDUCTORS AND CAPACITORS</b>	<b>475</b>
<b>39 INDUSTRIAL APPLICATIONS OF ELECTRIC MOTORS</b>	<b>480</b>

<b>40 HEATING AND WELDING</b>	<b>524</b>
<b>41 ELECTROLYTIC AND ELECTRO METALLURGICAL PROCESSES</b>	<b>539</b>
<b>42 ILLUMINATION</b>	<b>544</b>
<b>43 ELECTRIC TRACTION SPEED TIME CURVES AND MECHANICS OF TRAIN MOVEMENT</b>	<b>556</b>
<b>44 MOTORS FOR ELECTRIC TRACTION</b>	<b>573</b>
<b>45 CONTROL OF MOTORS</b>	<b>582</b>
<b>46 BRAKING</b>	<b>588</b>
<b>47 ELECTRIC TRACTION SYSTEMS AND POWER SUPPLY</b>	<b>595</b>

# List of Scilab Codes

Exa 2.1	Limiting value and Coal per hour . . . . .	5
Exa 2.2	Average load on power plant . . . . .	6
Exa 2.3	Heat balance sheet . . . . .	7
Exa 3.1	Firm capacity and Yearly gross output . . .	10
Exa 3.3	Available continuous power . . . . .	11
Exa 3.4	Minimum flow of river water to operate the plant . . . . .	12
Exa 7.1	Demand factor and Load factor . . . . .	14
Exa 7.2	Total energy generated annually . . . . .	15
Exa 7.3	Annual load factors and Capacity factors of two power stations . . . . .	16
Exa 7.4	Reserve capacity of plant . . . . .	18
Exa 7.5	Number of units supplied annually Diversity factor and Demand factor . . . . .	19
Exa 7.6	Annual load factor . . . . .	21
Exa 7.7	Diversity factor and Annual load factor . . .	22
Exa 7.8	Maximum demand and Connected load of each type . . . . .	24
Exa 7.9	Size and number of generator units Reserve plant capacity Load factor Plant factor and Plant use factor . . . . .	26
Exa 7.10	Cost of generation per kWh at 100 and 50 percent load factor . . . . .	28
Exa 7.11	Cost per unit generated . . . . .	30
Exa 7.12	Minimum reserve capacity of station and Cost per kWh generated . . . . .	31
Exa 7.13	Two part tariff to be charged from consumers	33
Exa 7.14	Generation cost in two part form . . . . .	35

Exa 7.15	Overall generating cost per unit at 50 and 100 percent capacity factor . . . . .	37
Exa 7.16	Yearly cost per kW demand and Cost per kWh supplied at substations and Consumer premises . . . . .	40
Exa 7.17	Number of working hours per week above which the HV supply is cheaper . . . . .	42
Exa 7.18	Cheaper alternative to adopt and by how much	44
Exa 7.19	Valuation halfway based on Straight line Reducing balance and Sinking fund depreciation method . . . . .	
Exa 7.20	Type and hp ratings of two turbines for the station . . . . .	47
Exa 7.21	Plot of chronological load curve and Load duration curve . . . . .	48
Exa 7.22	Daily energy produced Reserve capacity and Maximum energy produced at all time and fully loaded . . . . .	50
Exa 7.23	Rating Annual energy produced Total fixed and variable cost Cost per kWh generated Overall efficiency and Quantity of cooling water required . . . . .	54
Exa 7.24	Turbine rating Energy produced Average steam consumption Evaporation capacity Total fixed cost and variable cost and Cost per kWh generated . . . . .	56
Exa 7.25	Plot of hydrograph and Average discharge available . . . . .	60
Exa 7.26	Plot of flow duration curve Maximum power Average power developed and Capacity of proposed station . . . . .	63
Exa 9.1	Loop inductance and Reactance of transmission line . . . . .	67
Exa 9.2	Inductance per phase of the system . . . . .	72
Exa 9.3	Loop inductance of line per km . . . . .	73
Exa 9.4	Inductance per phase of the system . . . . .	74
Exa 9.5	Total inductance of the line . . . . .	75
Exa 9.6	Inductance of the line . . . . .	76
		78

Exa 9.7	Inductance per km of the double circuit line	79
Exa 9.8	Geometric mean radius of the conductor and Ratio of GMR to overall conductor radius .	81
Exa 9.9	Inductance of the line per phase . . . . .	82
Exa 9.10	Inductance per km of 3 phase transmission line	84
Exa 9.11	Inductance of each conductor per phase per km . . . . .	85
Exa 9.12	Inductance of each conductor and Average inductance of each phase . . . . .	86
Exa 9.13	Inductance per phase . . . . .	88
Exa 9.14	Inductance per phase of double circuit . . .	90
Exa 9.15	Spacing between adjacent conductor to keep same inductance . . . . .	92
Exa 9.16	Capacitance of line neglecting and taking presence of ground . . . . .	94
Exa 9.17	Capacitance of conductor . . . . .	95
Exa 9.18	New value of capacitance . . . . .	97
Exa 9.19	Capacitance per phase to neutral of a line .	98
Exa 9.20	Phase to neutral capacitance . . . . .	100
Exa 9.21	Capacitance per phase to neutral . . . . .	101
Exa 9.22	Capacitive reactance to neutral and Charging current per phase . . . . .	102
Exa 9.23	Inductive reactance Capacitance and Capacitive reactance of the line . . . . .	104
Exa 9.24	Capacitance of the line and Charging current	106
Exa 9.25	Capacitance of the line . . . . .	107
Exa 9.26	Capacitance of each line conductor . . . . .	109
Exa 10.1	Voltage regulation Sending end power factor and Transmission efficiency . . . . .	110
Exa 10.2	Line current Receiving end voltage and Efficiency of transmission . . . . .	112
Exa 10.3	Sending end voltage . . . . .	114
Exa 10.4	Distance over which load is delivered . . . .	115
Exa 10.5	Sending end voltage Voltage regulation Value of capacitors and Transmission efficiency . .	116
Exa 10.6	Voltage regulation Sending end voltage Line loss and Sending end power factor . . . . .	119

Exa 10.7	Nominal pi equivalent circuit parameters and Receiving end voltage . . . . .	121
Exa 10.8	Voltage Current and Power factor at sending end . . . . .	122
Exa 10.9	Sending end voltage Current and Transmission efficiency . . . . .	125
Exa 10.10	Line to line voltage and Power factor at sending end . . . . .	127
Exa 10.11	Voltage Current Power factor at sending end Regulation and Transmission efficiency by Nominal T and Pi method . . . . .	130
Exa 10.12	Receiving end Voltage Load and Nature of compensation required . . . . .	134
Exa 10.13	Sending end voltage and Current . . . . .	136
Exa 10.14	Incident voltage and Reflected voltage at receiving end and 200 km from receiving end .	137
Exa 10.15	A B C D constants . . . . .	139
Exa 10.16	Sending end voltage Current Power factor and Efficiency . . . . .	140
Exa 10.17	Values of auxiliary constants A B C D . . .	143
Exa 10.18	Sending end voltage and Current using convergent series method . . . . .	145
Exa 10.19	Sending end voltage and Current using nominal pi and nominal T method . . . . .	147
Exa 10.20	Sending end voltage Voltage regulation Transmission efficiency and A B C D constants by Short line Nominal T Nominal pi and Long line approximation . . . . .	150
Exa 10.21	Sending end voltage Current Power factor and Efficiency of transmission . . . . .	160
Exa 10.23	Overall constants A B C D . . . . .	163
Exa 10.24	Values of constants A0 B0 C0 D0 . . . . .	164
Exa 10.25	Maximum power transmitted Receiving end power factor and Total line loss . . . . .	165
Exa 10.26	Maximum power that can be transferred to the load . . . . .	167
Exa 11.1	Ratio of capacitance Line voltage and String efficiency . . . . .	169

Exa 11.2	Mutual capacitance of each unit in terms of C	170
Exa 11.3	Voltage distribution over a string of three suspension insulators and String efficiency . . . . .	171
Exa 11.4	Line to neutral voltage and String efficiency	172
Exa 11.5	Value of line to pin capacitance . . . . .	173
Exa 11.6	Voltage distribution as a percentage of voltage of conductor to earth and String efficiency	174
Exa 11.7	Voltage across each insulator as a percentage of line voltage to earth and String efficiency With and Without guard ring . . . . .	176
Exa 11.8	Voltage across each insulator as a percentage of line voltage to earth and String efficiency	178
Exa 11.9	Voltage on the line end unit and Value of capacitance required . . . . .	179
Exa 12.1	Weight of conductor . . . . .	181
Exa 12.2	Point of maximum sag at the lower support	182
Exa 12.3	Vertical sag . . . . .	183
Exa 12.4	Height above ground at which the conductors should be supported . . . . .	184
Exa 12.5	Permissible span between two supports . . . . .	186
Exa 12.6	Maximum sag of line due to weight of conductor Additional weight of ice Plus wind and Vertical sag . . . . .	187
Exa 12.7	Point of minimum sag . . . . .	189
Exa 12.8	Clearance between conductor and water at a point midway between towers . . . . .	190
Exa 12.9	Sag at erection and Tension of the line . . . . .	191
Exa 12.10	Sag in inclined direction and Vertical direction	193
Exa 12.11	Sag in still air Wind pressure Ice coating and Vertical sag . . . . .	194
Exa 13.1	Mutual inductance between the circuits and Voltage induced in the telephone line . . . . .	197
Exa 13.2	Induced voltage at fundamental frequency and Potential of telephone conductor . . . . .	198
Exa 14.1	Insulation resistance per km . . . . .	202
Exa 14.2	Insulation thickness . . . . .	203
Exa 14.3	Capacitance and Charging current of single core cable . . . . .	204

Exa 14.4	Most economical diameter of a single core cable and Overall diameter of the insulation . . . . .	205
Exa 14.6	Conductor radius and Electric field strength that must be withstood . . . . .	206
Exa 14.7	Location of intersheath and Ratio of maximum electric field strength with and without intersheath . . . . .	207
Exa 14.8	Maximum and Minimum stress in the insulation . . . . .	208
Exa 14.9	Maximum stress with and without intersheath . . . . .	209
Exa 14.10	Best position and Voltage on each intersheath . . . . .	211
Exa 14.11	Maximum stress in the two dielectrics . . . . .	212
Exa 14.12	Diameter and Voltage of intersheath Conductor and Outside diameter of graded cable and Ungraded cable . . . . .	213
Exa 14.13	Equivalent star connected capacity and kVA required . . . . .	214
Exa 14.14	Charging current drawn by a cable with three cores . . . . .	215
Exa 14.15	Capacitance between any two conductors Two bounded conductors Capacitance to neutral and Charging current taken by cable . . . . .	216
Exa 14.16	Charging current drawn by cable . . . . .	217
Exa 15.1	Capacitance of the cable Charging current Total charging kVAR Dielectric loss per phase and Maximum stress in the cable . . . . .	218
Exa 15.2	Minimum spacing between conductors . . . . .	219
Exa 15.3	Critical disruptive voltage and Corona loss . . . . .	220
Exa 15.4	Corona loss in fair weather and Foul weather . . . . .	221
Exa 15.5	Corona characteristics . . . . .	222
Exa 15.6	Spacing between the conductors . . . . .	223
Exa 15.7	Disruptive critical voltage and Corona loss . . . . .	224
Exa 15.8	Corona will be present in the air space or not . . . . .	225
Exa 16.1	Line voltage for commencing of corona . . . . .	226
Exa 16.3	Bus admittance matrix Ybus . . . . .	227
Exa 16.4	Voltage values at different buses . . . . .	228
Exa 16.5	New bus admittance matrix Ybus . . . . .	229
	Bus admittance matrix V1 and V2 . . . . .	230

Exa 16.6	Bus impedance matrix Zbus . . . . .	242
Exa 16.7	Power flow expressions . . . . .	243
Exa 16.8	Voltage V2 by GS method . . . . .	244
Exa 17.1	Operating power angle and Magnitude of P0	247
Exa 17.2	Minimum value of E and VL Maximum power limit and Steady state stability margin . . .	248
Exa 17.3	Maximum power transfer if shunt inductor and Shunt capacitor is connected at bus 2 .	249
Exa 17.4	Maximum power transfer and Stability margin	251
Exa 17.5	QgB Phase angle of VB and What happens if QgB is made zero . . . . .	252
Exa 17.6	Steady state stability limit with two terminal voltages constant and If shunt admittance is zero and series resistance neglected . . . .	254
Exa 17.8	Power angle diagram Maximum power the line is capable of transmitting and Power transmitted with equal voltage at both ends . . .	255
Exa 17.9	Maximum steady state power that can be transmitted over the line . . . . .	259
Exa 17.10	Maximum steady state power Value of P and Q if static capacitor is connected and Replaced by an inductive reactor . . . . .	260
Exa 17.11	Kinetic energy stored in the rotor at synchronous speed and Acceleration . . . . .	263
Exa 17.12	Kinetic energy stored in the rotor at synchronous speed and Acceleration . . . . .	264
Exa 17.13	Change in torque angle in that period and RPM at the end of 10 cycles . . . . .	265
Exa 17.14	Accelerating torque at the time the fault occurs . . . . .	266
Exa 17.16	Value of H and in 100 MVA base . . . . .	267
Exa 17.17	Equivalent H for the two to common 100 MVA base . . . . .	268
Exa 17.18	Energy stored in the rotor at the rated speed Value of H and Angular momentum . . . . .	269
Exa 17.19	Acceleration of the rotor . . . . .	270
Exa 17.20	Accelerating power and New power angle after 10 cycles . . . . .	271

Exa 17.21	Kinetic energy stored by rotor at synchronous speed and Acceleration in . . . . .	272
Exa 17.22	Change in torque angle and Speed in rpm at the end of 10 cycles . . . . .	274
Exa 17.23	Accelerating torque at the time of fault occurrence . . . . .	275
Exa 17.24	Swing equation . . . . .	276
Exa 17.26	Critical clearing angle . . . . .	279
Exa 17.27	Critical angle using equal area criterion . . . . .	281
Exa 17.28	Critical clearing angle . . . . .	282
Exa 17.30	Power angle and Swing curve data . . . . .	283
Exa 18.1	Load shared by two machines and Load at which one machine ceases to supply any portion of load . . . . .	287
Exa 18.2	Synchronizing power and Synchronizing torque for no load and full load . . . . .	289
Exa 18.3	Armature current EMF and PF of the other alternator . . . . .	292
Exa 18.4	New value of machine current and PF Power output Current and PF corresponding to maximum load . . . . .	293
Exa 18.5	Phase angle between busbar sections . . . . .	295
Exa 18.6	Voltage and Power factor at this latter station	296
Exa 18.7	Load received Power factor and Phase difference between voltage . . . . .	298
Exa 18.8	Percentage increase in voltage and Phase angle difference between the two busbar voltages	300
Exa 18.9	Station power factors and Phase angle between two busbar voltages . . . . .	302
Exa 18.10	Constants of the second feeder . . . . .	304
Exa 18.11	Necessary booster voltages . . . . .	305
Exa 18.12	Load on C at two different conditions of load in A and B . . . . .	307
Exa 18.13	Loss in the interconnector as a percentage of power received and Required voltage of the booster . . . . .	309
Exa 20.4	Reflected and Transmitted wave of Voltage and Current at the junction . . . . .	312

Exa 20.5	First and Second voltages impressed on C . . . . .	313
Exa 20.6	Voltage and Current in the cable and Open wire lines . . . . .	315
Exa 21.1	Ratio of voltages appearing at the end of a line when line is open circuited and Terminate d by arrester . . . . .	317
Exa 21.2	Choosing suitable arrester rating . . . . .	318
Exa 22.1	Highest voltage to which the transformer is subjected . . . . .	322
Exa 22.2	Rating of LA and Location with respect to transformer . . . . .	323
Exa 23.1	Inductance and Rating of arc suppression coil . . . . .	326
Exa 24.1	Weight of copper required for a three phase transmission system and DC transmission sys tem . . . . .	328
Exa 24.2	Percentage increase in power transmitted . . . . .	330
Exa 24.3	Percentage additional balanced load . . . . .	330
Exa 24.4	Amount of copper required for 3 phase 4 wire system with that needed for 2 wire dc system . . . . .	331
Exa 24.5	Weight of copper required and Reduction of weight of copper possible . . . . .	332
Exa 24.6	Economical cross section of a 3 core distribu tor cable . . . . .	334
Exa 24.7	Most economical cross section . . . . .	335
Exa 24.8	Most economical current density for the trans mission line . . . . .	337
Exa 24.9	Most economical cross section of the conductor . . . . .	338
Exa 25.1	Potential of O and Current leaving each sup ply point . . . . .	341
Exa 25.2	Point of minimum potential along the track and Currents supplied by two substations . . . . .	342
Exa 25.3	Position of lowest run lamp and its Voltage . . . . .	344
Exa 25.4	Point of minimum potential and its Potential . . . . .	346
Exa 25.6	Ratio of weight of copper with and without interconnector . . . . .	348
Exa 25.7	Potential difference at each load point . . . . .	350
Exa 25.8	Load on the main generators and On each balancer machine . . . . .	353

Exa 25.9	Currents in various sections and Voltage at load point C . . . . .	354
Exa 27.1	Per unit current . . . . .	358
Exa 27.2	kVA at a short circuit fault between phases at the HV terminal of transformers and Load end of transmission line . . . . .	360
Exa 27.3	Transient short circuit current and Sustained short circuit current at X . . . . .	362
Exa 27.4	Current in the short circuit . . . . .	367
Exa 27.5	Per unit values of the single line diagram . .	369
Exa 27.6	Actual fault current using per unit method . .	372
Exa 27.7	Sub transient fault current . . . . .	374
Exa 27.8	Voltage behind the respective reactances . .	376
Exa 27.9	Initial symmetrical rms current in the hv side and lv side . . . . .	377
Exa 27.10	Initial symmetrical rms current at the generator terminal . . . . .	378
Exa 27.11	Sub transient current in the fault in generator and Motor . . . . .	380
Exa 27.12	Sub transient fault current Fault current rating of generator breaker and Each motor breaker	381
Exa 28.1	Reactance necessary to protect the switchgear	384
Exa 28.2	kVA developed under short circuit when reactors are in circuit and Short circuited . . .	386
Exa 28.4	Reactance of each reactor . . . . .	387
Exa 28.5	Instantaneous symmetrical short circuit MVA for a fault at X . . . . .	389
Exa 29.1	Positive Negative and Zero sequence currents	391
Exa 29.4	Sequence components of currents in the resistors and Supply lines . . . . .	392
Exa 29.5	Magnitude of positive and Negative sequence components of the delta and Star voltages .	394
Exa 29.6	Current in each line by the method of symmetrical components . . . . .	396
Exa 29.7	Symmetrical components of line current if phase 3 is only switched off . . . . .	398
Exa 29.8	Positive Negative and Zero sequence components of currents for all phases . . . . .	400

Exa 29.9	Currents in all the lines and their symmetrical components . . . . .	402
Exa 29.10	Radius of voltmeter connected to the yellow line and Current through the voltmeter . . . . .	404
Exa 29.11	Three line currents and Wattmeter reading . . . . .	406
Exa 30.1	Initial symmetrical rms line currents Ground wire currents and Line to neutral voltages involving ground and Solidly grounded fault . . . . .	408
Exa 30.2	Current in the line with two lines short circuited . . . . .	412
Exa 30.3	Fault current Sequence component of current and Voltages of the sound line to earth at fault . . . . .	415
Exa 30.4	Fault currents in each line and Potential above earth attained by the alternator neutrals . . . . .	418
Exa 30.5	Fault currents . . . . .	420
Exa 30.6	Fault current for line fault and Line to ground fault . . . . .	422
Exa 30.7	Fault current for a LG fault at C . . . . .	425
Exa 30.8	Fault current when a single phase to earth fault occurs . . . . .	430
Exa 30.9	Fault currents in the lines . . . . .	432
Exa 30.10	Currents in the faulted phase Current through ground and Voltage of healthy phase to neutral . . . . .	433
Exa 30.11	Fault currents . . . . .	435
Exa 30.12	Fault current if all 3 phases short circuited If single line is grounded and Short circuit between two lines . . . . .	437
Exa 30.13	Sub transient current in the faulty phase . . . . .	440
Exa 30.14	Initial symmetrical rms current in all phases of generator . . . . .	441
Exa 32.1	Maximum restriking voltage Frequency of transient oscillation and Average rate of rise of voltage upto first peak of oscillation . . . . .	444
Exa 32.3	Rate of rise of restriking voltage . . . . .	445
Exa 32.5	Voltage across the pole of a CB and Resistance to be used across the contacts . . . . .	446
Exa 32.6	Rated normal current Breaking current Making current and Short time rating . . . . .	448

Exa 32.8	Sustained short circuit Initial symmetrical rms current Maximum possible dc component of the short circuit Momentary current rating Current to be interrupted and Interrupting kVA . . . . .	449
Exa 33.1	Time of operation of the relay . . . . .	451
Exa 33.2	Time of operation of the relay . . . . .	452
Exa 33.3	Operating time of feeder relay Minimum plug setting of transformer relay and Time setting of transformer . . . . .	453
Exa 33.4	Time of operation of the two relays . . . . .	455
Exa 33.6	Will the relay operate the trip of the breaker	456
Exa 34.1	Neutral earthing reactance . . . . .	458
Exa 34.2	Unprotected portion of each phase of the stator winding against earth fault and Effect of varying neutral earthing resistance . . . . .	459
Exa 34.3	Portion of alternator winding unprotected . .	461
Exa 34.4	Will the relay trip the generator CB . . . . .	462
Exa 34.5	Winding of each phase unprotected against earth when machine operates at nominal voltage . . . . .	463
Exa 34.6	Portion of winding unprotected . . . . .	464
Exa 34.7	Percentage of winding that is protected against earth faults . . . . .	465
Exa 34.8	Magnitude of neutral earthing resistance . .	467
Exa 35.2	Ratio of CTs . . . . .	469
Exa 35.3	Ratio of CTs on high voltage side . . . . .	470
Exa 35.4	Ratio of protective CTs . . . . .	471
Exa 35.5	CT ratios on high voltage side . . . . .	472
Exa 35.6	Suitable CT ratios . . . . .	473
Exa 36.1	First Second and Third zone relay setting Without infeed and With infeed . . . . .	475
Exa 36.2	Impedance seen by relay and Relay setting for high speed backup protection . . . . .	478
Exa 39.1	Total annual cost of group drive and Individual drive . . . . .	480

Exa 39.2	Starting torque in terms of full load torque with star delta starter and with Auto transformer starter . . . . .	482
Exa 39.3	Tapping to be provided on an auto transformer Starting torque in terms of full load torque and with Resistor used . . . . .	483
Exa 39.4	Starting torque and Starting current if motor started by Direct switching Star delta starter Star connected auto transformer and Series parallel switch . . . . .	484
Exa 39.5	Motor current per phase Current from the supply Starting torque Voltage to be applied and Line current . . . . .	487
Exa 39.6	Ratio of starting current to full load current	488
Exa 39.7	Resistance to be placed in series with shunt field . . . . .	489
Exa 39.9	Speed and Current when field winding is shunted by a diverter . . . . .	491
Exa 39.10	Additional resistance to be inserted in the field circuit to raise the speed . . . . .	492
Exa 39.11	Speed of motor with a diverter connected in parallel with series field . . . . .	493
Exa 39.12	Diverter resistance as a percentage of field resistance . . . . .	494
Exa 39.13	Additional resistance to be placed in the armature circuit . . . . .	495
Exa 39.14	Resistance to be connected in series with armature to reduce speed . . . . .	496
Exa 39.15	Ohmic value of resistor connected in the armature circuit . . . . .	497
Exa 39.16	External resistance per phase added in rotor circuit to reduce speed . . . . .	499
Exa 39.17	Braking torque and Torque when motor speed has fallen . . . . .	500
Exa 39.18	Initial plugging torque and Torque at standstill	501
Exa 39.19	Value of resistance to be connected in motor circuit . . . . .	503

Exa 39.20	Current drawn by the motor from supply and Resistance required in the armature circuit for rheostatic braking . . . . .	504
Exa 39.21	One hour rating of motor . . . . .	505
Exa 39.22	Final temperature rise and Thermal time constant of the motor . . . . .	506
Exa 39.23	Half hour rating of motor . . . . .	508
Exa 39.24	Time for which the motor can run at twice the continuously rated output without overheating	509
Exa 39.25	Maximum overload that can be carried by the motor . . . . .	510
Exa 39.26	Required size of continuously rated motor .	511
Exa 39.27	Suitable size of the motor . . . . .	512
Exa 39.28	Time taken to accelerate the motor to rated speed against full load torque . . . . .	514
Exa 39.29	Time taken to accelerate the motor to rated speed . . . . .	515
Exa 39.30	Time taken to accelerate a fly wheel . . . .	516
Exa 39.31	Time taken for dc shunt motor to fall in speed with constant excitation and Time for the same fall if frictional torque exists . . . . .	517
Exa 39.32	Time taken and Number of revolutions made to come to standstill by Plugging and Rheostatic braking . . . . .	518
Exa 39.33	Inertia of flywheel required . . . . .	520
Exa 39.34	Moment of inertia of the flywheel . . . . .	522
Exa 40.1	Diameter Length and Temperature of the wire	524
Exa 40.2	Width and Length of nickel chrome strip .	526
Exa 40.3	Power drawn under various connections . .	527
Exa 40.4	Amount of energy required to melt brass . .	530
Exa 40.5	Height up to which the crucible should be filled to obtain maximum heating effect . .	531
Exa 40.6	Voltage necessary for heating and Current flowing in the material . . . . .	532
Exa 40.7	Voltage applied across electrodes and Current through the material . . . . .	534
Exa 40.8	Time taken to melt Power factor and Electrical efficiency of the furnace . . . . .	536

Exa 41.1	Quantity of electricity and Time taken for the process . . . . .	539
Exa 41.2	Annual output of refined copper and Energy consumption . . . . .	540
Exa 41.3	Weight of aluminium produced from aluminium oxide . . . . .	541
Exa 42.2	mscp of lamp Illumination on the surface when it is normal Inclined to 45 degree and Parallel to rays . . . . .	544
Exa 42.3	Illumination at the centre Edge of surface with and Without reflector and Average illumination over the area without reflector . . . . .	545
Exa 42.5	cp of the globe and Percentage of light emitted by lamp that is absorbed by the globe .	547
Exa 42.6	Curve showing illumination on a horizontal line below lamp . . . . .	548
Exa 42.7	Maximum and Minimum illumination on the floor along the centre line . . . . .	551
Exa 42.8	Illumination on the working plane . . . . .	553
Exa 42.9	Suitable scheme of illumination and Saving in power consumption . . . . .	554
Exa 43.1	Maximum speed over the run . . . . .	556
Exa 43.2	Value of retardation . . . . .	557
Exa 43.3	Rate of acceleration required to operate service	558
Exa 43.4	Duration of acceleration Coasting and Braking periods . . . . .	560
Exa 43.5	Tractive resistance . . . . .	561
Exa 43.6	Torque developed by each motor . . . . .	562
Exa 43.7	Time taken by train to attain speed . . . . .	563
Exa 43.8	Speed Time curve for the run and Energy consumption at the axles of train . . . . .	565
Exa 43.9	Acceleration Coasting retardation and Scheduled speed . . . . .	567
Exa 43.10	Minimum adhesive weight of the locomotive	569
Exa 43.11	Energy usefully employed in attaining speed and Specific energy consumption at steady state speed . . . . .	570
Exa 43.12	Minimum adhesive weight of a locomotive .	571

Exa 44.1	Speed current of the motor . . . . .	573
Exa 44.2	Speed torque for motor . . . . .	575
Exa 44.3	Speed of motors when connected in series . .	577
Exa 44.4	HP delivered by the locomotive when dc series motor and Induction motor is used . . . . .	578
Exa 44.5	New characteristics of motor . . . . .	579
Exa 45.1	Approximate loss of energy in starting rheostats	582
Exa 45.2	Energy supplied during the starting period Energy lost in the starting resistance and Use- ful energy supplied to the train . . . . .	583
Exa 45.3	Duration of starting period Speed of train at transition Rheostatic losses during series and Parallel steps of starting . . . . .	585
Exa 46.1	Braking torque . . . . .	588
Exa 46.2	Current delivered when motor works as gen- erator . . . . .	589
Exa 46.3	Energy returned to lines . . . . .	590
Exa 46.4	Energy returned to the line . . . . .	592
Exa 46.5	Braking effect and Rate of retardation pro- duced by this braking effect . . . . .	593
Exa 47.1	Maximum potential difference between any two points of the rails and Rating of the booster	595
Exa 47.2	Maximum sag and Length of wire required .	596

# Chapter 2

## THERMAL STATIONS

Scilab code Exa 2.1 Limiting value and Coal per hour

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 2: THERMAL STATIONS
8
9 // EXAMPLE : 2.1 :
10 // Page number 25–26
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 //Given data
14 M = 15000.0+10.0          // Water evaporated(kg)
15 C = 5000.0+5.0           // Coal consumption(kg)
16 time = 8.0                // Generation shift time(
    hours)
17
18 //Calculations
19 //Case(a)
```

```

20 M1 = M-15000.0
21 C1 = C-5000.0
22 M_C = M1/C1
    // Limiting value of water evaporation(kg)
23 //Case(b)
24 kWh = 0
    // Station output at no load
25 consumption_noload = 5000+5*kWh
    // Coal consumption at no load(kg)
26 consumption_noload_hr = consumption_noload/time
    // Coal consumption per hour(kg)
27
28 //Results
29 disp("PART I - EXAMPLE : 2.1 : SOLUTION :-")
30 printf("\nCase(a): Limiting value of water
        evaporation per kg of coal consumed , M/C = %.f kg
        ", M_C)
31 printf("\nCase(b): Coal per hour for running station
        at no load = %.f kg\n", consumption_noload_hr)

```

---

### Scilab code Exa 2.2 Average load on power plant

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 2: THERMAL STATIONS
8
9 // EXAMPLE : 2.2 :
10 // Page number 26
11 clear ; clc ; close ; // Clear the work space and
    console
12

```

```

13 // Given data
14 amount = 25.0*10**5           // Amount spent in 1
   year(Rs)
15 value_heat = 5000.0          // Heating value(kcal/
   kg)
16 cost = 500.0                // Cost of coal per
   ton(Rs)
17 n_ther = 0.35               // Thermal efficiency
18 n_elec = 0.9                // Electrical
   efficiency
19
20 // Calculations
21 n = n_ther*n_elec          // Overall efficiency
22 consumption = amount/cost*1000 // Coal
   consumption in 1 year(kg)
23 combustion = consumption*value_heat // Heat
   of combustion(kcal)
24 output = n*combustion        // Heat
   output(kcal)
25 unit_gen = output/860.0      // Annual
   heat generated(kWh). 1 kWh = 860 kcal
26 hours_year = 365*24.0        // Total
   time in a year(hour)
27 load_average = unit_gen/hours_year // Average load on the power plant(kW)
28
29 // Result
30 disp("PART I - EXAMPLE : 2.2 : SOLUTION :-")
31 printf("\nAverage load on power plant = %.2f kW\n",
   load_average)
32 printf("\nNOTE: ERROR: Calculation mistake in the
   final answer in the textbook")

```

---

### Scilab code Exa 2.3 Heat balance sheet

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 2: THERMAL STATIONS
8
9 // EXAMPLE : 2.3 :
10 // Page number 26
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 //Given data
14 consumption = 0.5          // Coal consumption per kWh
    output(kg)
15 cal_value = 5000.0          // Calorific value(kcal/kg)
16 n_boiler = 0.8              // Boiler efficiency
17 n_elec = 0.9                // Electrical efficiency
18
19 //Calculations
20 input_heat = consumption*cal_value
    // Heat input(kcal)
21 input_elec = input_heat/860.0
    // Equivalent electrical energy(kWh). 1 kWh = 860
    kcal
22 loss_boiler = input_elec*(1-n_boiler)
    // Boiler loss(kWh)
23 input_steam = input_elec-loss_boiler
    // Heat input to steam(kWh)
24 input_alter = 1/n_elec
    // Alternator input(kWh)
25 loss_alter = input_alter*(1-n_elec)
    // Alternate loss(kWh)
26 loss_turbine = input_steam-input_alter
    // Loss in turbine(kWh)
27 loss_total = loss_boiler+loss_alter+loss_turbine
    // Total loss(kWh)

```

```

28 output = 1.0
    // Output(kWh)
29 Input = output+loss_total
    // Input(kWh)
30
31 // Results
32 disp("PART I - EXAMPLE : 2.3 : SOLUTION :-")
33 printf("\nHeat Balance Sheet")
34 printf("\nLOSSES: Boiler loss      = %.3f kWh",
        loss_boiler)
35 printf("\n          Alternator loss = %.2f kWh",
        loss_alter)
36 printf("\n          Turbine loss   = %.3f kWh",
        loss_turbine)
37 printf("\n          Total loss     = %.2f kWh",
        loss_total)
38 printf("\nOUTPUT: %.1f kWh", output)
39 printf("\nINPUT: %.2f kWh\n", Input)

```

---

# Chapter 3

## HYDRO ELECTRIC STATIONS

Scilab code Exa 3.1 Firm capacity and Yearly gross output

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 3: HYDRO-ELECTRIC STATIONS
8
9 // EXAMPLE : 3.1 :
10 // Page number 41
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 Q = 95.0          // Minimum run-off ( $m^3/sec$ )
15 h = 40.0          // Head (m)
16
17 // Calculations
18 w = 1000.0        // Density of water (kg/m3)
```

```

        ^3)
19 weight = Q*w                         // Weight of water per
   sec(kg)
20 work_done = weight*h                  // Work done in one
   second(kg-mt)
21 kW_1 = 75.0/0.746                     // 1 kW(kg-mt/sec)
22 power = work_done/kW_1                // Power production(kW)
23 hours_year = 365.0*24                 // Total hours in a year
24 output = power*365*24.0               // Yearly gross output(
   kWhr)
25
26 // Results
27 disp("PART I - EXAMPLE : 3.1 : SOLUTION :-")
28 printf("\nFirm capacity = %.f kW", power)
29 printf("\nYearly gross output = %.2e kWhr.", output)

```

---

### Scilab code Exa 3.3 Available continuous power

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 3: HYDRO-ELECTRIC STATIONS
8
9 // EXAMPLE : 3.3 :
10 // Page number 41
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 A = 200.0                      // Catchment area(Sq.km)
15 F = 1000.0                      // Annual rainfall(mm)
16 H = 200.0                       // Effective head(m)

```

```

17 K = 0.5           // Yield factor
18 n = 0.8           // Plant efficiency
19
20 // Calculations
21 P = 3.14*n*K*A*F*H*10**-4      // Available continuous
   power (kW)
22
23 // Results
24 disp("PART I - EXAMPLE : 3.3 : SOLUTION :-")
25 printf("\nAvailable continuous power of hydro-
   electric station , P = %.f kW" , P)

```

---

#### Scilab code Exa 3.4 Minimum flow of river water to operate the plant

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 3: HYDRO-ELECTRIC STATIONS
8
9 // EXAMPLE : 3.4 :
10 // Page number 41-42
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 load_factor = 0.15      // Load factor
15 P = 10.0*10**3          // Rated installed capacity (kW
   )
16 H = 50.0                 // Head of plant (m)
17 n = 0.8                  // Efficiency of plant
18
19 // Calculation

```

```
20 units_day = P*load_factor      // Total units
     generated daily on basis of load factor(kWhr)
21 units_week = units_day*24.0*7   // Total units
     generated for one week(kWhr)
22 Q = units_week/(9.81*H*n*24*7) // Minimum flow of
     water(cubic mt/sec)
23
24 // Result
25 disp("PART I - EXAMPLE : 3.4 : SOLUTION :-")
26 printf("\nMinimum flow of river water to operate the
     plant , Q = %.3f cubic mt/sec" , Q)
```

---

# Chapter 7

## TARIFFS AND ECONOMIC ASPECTS IN POWER GENERATION

Scilab code Exa 7.1 Demand factor and Load factor

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
    GENERATION
8
9 // EXAMPLE : 7.1 :
10 // Page number 73
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 connected_load = 450.0*10**3           // Connected load
    (kW)
```

```

15 maximum_demand = 250.0*10**3           // Maximum demand
   (kW)
16 units_generated = 615.0*10**6          // Units
   generated per annum(kWh)
17
18 // Calculations
19 // Case(i)
20 demand_factor = maximum_demand/connected_load
   // Demand factor
21 // Case(ii)
22 hours_year = 365.0*24
   // Total hours in
   a year
23 average_demand = units_generated/hours_year
   // Average demand(kW)
24 load_factor = average_demand/maximum_demand*100
   // Load factor(%)
25
26 // Results
27 disp("PART I - EXAMPLE : 7.1 : SOLUTION :-")
28 printf("\nCase(i) : Demand factor = %.3f ", 
   demand_factor)
29 printf("\nCase(ii) : Load factor = %.1f percent",
   load_factor)

```

---

### Scilab code Exa 7.2 Total energy generated annually

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
   GENERATION

```

```

8
9 // EXAMPLE : 7.2 :
10 // Page number 73
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 maximum_demand = 480.0*10**3           // Maximum demand
   (kW)
15 LF = 0.4                                // Annual load
   factor
16
17 // Calculation
18 hours_year = 365.0*24                   //
   Total hours in a year
19 energy_gen = maximum_demand*LF*hours_year // 
   Total energy generated annually (kWh)
20
21 // Results
22 disp("PART I – EXAMPLE : 7.2 : SOLUTION :-")
23 printf("\nTotal energy generated annually = %.5e kWh
   ", energy_gen)

```

---

### Scilab code Exa 7.3 Annual load factors and Capacity factors of two power stations

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
   GENERATION
8
9 // EXAMPLE : 7.3 :

```

```

10 // Page number 73
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 cap_baseload = 400.0*10**3           // Installed
   capacity of base load plant(kW)
15 cap_standby = 50.0*10**3            // Installed
   capacity of standby unit(kW)
16 output_baseload = 101.0*10**6        // Annual baseload
   station output(kWh)
17 output_standby = 87.35*10**6         // Annual standby
   station output(kWh)
18 peakload_standby = 120.0*10**3       // Peak load on
   standby station(kW)
19 hours_use = 3000.0                  // Hours of standby
   station use/year(hrs)
20
21 // Calculations
22 // Case(i)
23 LF_1 = output_standby*100/(peakload_standby*
   hours_use) // Annual load factor(%)
24 hours_year = 365.0*24                // Total
   hours in a year
25 CF_1 = output_standby*100/(cap_standby*hours_year)
   // Annual capacity factor(%)
26 // Case(ii)
27 peakload_baseload = peakload_standby
   // Peak load on baseload
   station(kW)
28 LF_2 = output_baseload*100/(peakload_baseload*
   hours_use) // Annual load factor on baseload
   station(%)
29 hours_year = 365.0*24                // Total
   hours in a year
30 CF_2 = output_baseload*100/(cap_baseload*hours_year)

```

```

                // Annual capacity factor on baseload
station(%)
```

31

32 // Results

33 **disp**("PART I – EXAMPLE : 7.3 : SOLUTION :-")

34 **printf**("\\nCase(i) : Standby Station")

35 **printf**("\\n Annual load factor = %.2f  
 percent", LF\_1)

36 **printf**("\\n Annual capacity factor = %.2f  
 percent\\n", CF\_1)

37 **printf**("\\nCase(ii) : Base load Station")

38 **printf**("\\n Annual load factor = %.2f  
 percent", LF\_2)

39 **printf**("\\n Annual capacity factor = %.2f  
 percent\\n", CF\_2)

40 **printf**("\\nNOTE: Incomplete solution in the textbook"  
 ) ;

---

#### Scilab code Exa 7.4 Reserve capacity of plant

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
GENERATION
8
9 // EXAMPLE : 7.4 :
10 // Page number 74
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
```

```

14 MD = 500.0          // Maximum demand (MW)
15 LF = 0.5            // Annual load factor
16 CF = 0.4            // Annual capacity factor
17
18 // Calculations
19 hours_year = 365.0*24           // Total
   hours in a year
20 energy_gen = MD*LF*hours_year      // Energy
   generated/annum (MWh)
21 plant_cap = energy_gen/(CF*hours_year) // Plant
   capacity (MW)
22 reserve_cap = plant_cap-MD        // Reserve
   capacity of plant (MW)
23
24 // Results
25 disp("PART I – EXAMPLE : 7.4 : SOLUTION :-")
26 printf("\nReserve capacity of plant = %.f MW",
   reserve_cap)

```

---

### Scilab code Exa 7.5 Number of units supplied annually Diversity factor and Demand

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
   GENERATION
8
9 // EXAMPLE : 7.5 :
10 // Page number 74
11 clear ; clc ; close ; // Clear the work space and
   console
12

```

```

13 // Given data
14 load_1 = 150.0           // Load supplied by station(
    MW)
15 load_2 = 120.0           // Load supplied by station(
    MW)
16 load_3 = 85.0            // Load supplied by station(
    MW)
17 load_4 = 60.0             // Load supplied by station(
    MW)
18 load_5 = 5.0              // Load supplied by station(
    MW)
19 MD = 220.0                // Maximum demand(MW)
20 LF = 0.48                  // Annual load factor
21
22 // Calculations
23 // Case(a)
24 hours_year = 365.0*24          //
    Total hours in a year
25 units = LF*MD*hours_year          //
    Number of units supplied annually
26 // Case(b)
27 sum_demand = load_1+load_2+load_3+load_4+load_5      //
    Sum of maximum demand of individual consumers(MW
)
28 diversity_factor = sum_demand/MD          //
    Diversity factor
29 // Case(c)
30 DF = MD/sum_demand          //
    Demand factor
31
32 // Results
33 disp("PART I – EXAMPLE : 7.5 : SOLUTION :-")
34 printf("\nCase(a): Number of units supplied annually
    = %.2e units", units)
35 printf("\nCase(b): Diversity factor = %.3f ,
    diversity_factor)
36 printf("\nCase(c): Demand factor = %.3f = %.1f
    percent", DF,DF*100)

```

---

### Scilab code Exa 7.6 Annual load factor

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
    GENERATION
8
9 // EXAMPLE : 7.6 :
10 // Page number 74
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 power_del_1 = 1000.0      // Power delivered by
    station (MW)
15 time_1 = 2.0              // Time for which power is
    delivered (hours)
16 power_del_2 = 500.0      // Power delivered by
    station (MW)
17 time_2 = 6.0              // Time for which power is
    delivered (hours)
18 days_maint = 60.0         // Maintenance days
19 max_gen_cap = 1000.0     // Maximum generating
    capacity (MW)
20
21 // Calculations
22 energy_sup_day = (power_del_1*time_1)+(power_del_2*
    time_2) // Energy supplied for each working day
    (MWh)
23 days_total = 365.0
```

```

// Total days in a year
24 days_op = days_total-days_maint // Operating days of
station in a year
25 energy_sup_year = energy_sup_day*days_op // Energy supplied per year(
MWh)
26 hours_day = 24.0 // Total hours in a day
27 working_hours = days_op*hours_day // Hour of working in
a year
28 LF = energy_sup_year*100/(max_gen_cap*working_hours) // Annual load factor(%)
29
30 // Results
31 disp("PART I - EXAMPLE : 7.6 : SOLUTION :-")
32 printf("\nAnnual load factor = %.1f percent", LF)

```

---

### Scilab code Exa 7.7 Diversity factor and Annual load factor

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
GENERATION
8
9 // EXAMPLE : 7.7 :
10 // Page number 74
11 clear ; clc ; close ; // Clear the work space and

```

```

        console
12
13 // Given data
14 load_industry = 750.0           // Industrial
   consumer load supplied by station (MW)
15 load_commercial = 350.0         // Commercial
   establishment load supplied by station (MW)
16 load_power = 10.0              // Domestic power
   load supplied by station (MW)
17 load_light = 50.0               // Domestic light
   load supplied by station (MW)
18 MD = 1000.0                   // Maximum demand (MW)
19 kWh_gen = 50.0*10**5          // Number of kWh
   generated per year
20
21 // Calculations
22 // Case(i)
23 sum_demand = load_industry+load_commercial+
   load_power+load_light // Sum of max demand of
   individual consumers (MW)
24 diversity_factor = sum_demand/MD
                           // Diversity
   factor
25 // Case(ii)
26 hours_year = 365.0*24
                           //
   Total hours in a year
27 average_demand = kWh_gen/hours_year
                           // Average demand(
   MW)
28 LF = average_demand/MD*100
                           // Load
   factor (%)
29
30 // Results
31 disp("PART I - EXAMPLE : 7.7 : SOLUTION :-")
32 printf("\nCase(i) : Diversity factor = %.2f ", 
   diversity_factor)

```

```
33 printf("\nCase( ii ) : Annual load factor = %.f percent  
" , LF)
```

---

Scilab code Exa 7.8 Maximum demand and Connected load of each type

```
1 // A Texbook on POWER SYSTEM ENGINEERING  
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  
3 // DHANPAT RAI & Co.  
4 // SECOND EDITION  
5  
6 // PART I : GENERATION  
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER  
// GENERATION  
8  
9 // EXAMPLE : 7.8 :  
10 // Page number 74–75  
11 clear ; clc ; close ; // Clear the work space and  
// console  
12  
13 // Given data  
14 load Domestic = 15000.0 // Domestic  
// load supplied by station (kW)  
15 diversity Domestic = 1.25 // Diversity  
// factor of domestic load  
16 DF Domestic = 0.7 // Demand  
// factor of domestic load  
17 load Commercial = 25000.0 // Commercial  
// load supplied by station (kW)  
18 diversity Commercial = 1.2 // Diversity  
// factor of commercial load  
19 DF Commercial = 0.9 // Demand  
// factor of commercial load  
20 load Industry = 50000.0 // Industrial  
// load supplied by station (kW)  
21 diversity Industry = 1.3 // Diversity
```

```

        factor of industrial load
22 DF_industry = 0.98                                // Demand
        factor of industrial load
23 diversity_factor = 1.5                            // Overall
        system diversity factor
24
25 // Calculations
26 // Case(a)
27 sum_demand = load Domestic+load Commercial+
    load Industry // Sum of max demand of
    individual consumers (MW)
28 MD = sum_demand/diversity_factor
                    // Maximum demand
29 // Case(b)
30 MD Domestic = load Domestic*diversity Domestic
                    // Maximum domestic load demand (kW)
31 connected Domestic = MD Domestic/DF Domestic
                    // Connected domestic load (kW)
32 MD Commercial = load Commercial*diversity Commercial
                    // Maximum commercial load demand (kW)
33 connected Commercial = MD Commercial/DF Commercial
                    // Connected commercial load (kW)
34 MD Industry = load Industry*diversity Industry
                    // Maximum industrial load demand (kW)
35 connected Industry = MD Industry/DF Industry
                    // Connected industrial load (kW)
36
37 // Results
38 disp("PART I - EXAMPLE : 7.8 : SOLUTION :-")
39 printf("\nCase(a): Maximum demand = %.f kW", MD)
40 printf("\nCase(b): Connected domestic load = %.1f kW
        ", connected Domestic)
41 printf("\n            Connected commercial load = %.1f
        kW", connected Commercial)
42 printf("\n            Connected industrial load = %.1f
        kW", connected Industry)

```

---

### Scilab code Exa 7.9 Size and number of generator units Reserve plant capacity Load

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
     GENERATION
8
9 // EXAMPLE : 7.9 :
10 // Page number 75–76
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 MD = 10000.0          // Maximum demand(kW)
15 load_1 = 2000.0        // Load from 11 PM–6 AM(kW)
16 t_1 = 7.0              // Time from 11 PM–6 AM(hour)
17 load_2 = 3500.0        // Load from 6 AM–8 AM(kW)
18 t_2 = 2.0              // Time from 6 AM–8 AM(hour)
19 load_3 = 8000.0        // Load from 8 AM–12 Noon(kW)
20 t_3 = 4.0              // Time from 8 AM–12 Noon(hour)
21 load_4 = 3000.0        // Load from 12 Noon–1 PM(kW)
22 t_4 = 1.0              // Time from 12 Noon–1 PM(hour)
23 load_5 = 7500.0        // Load from 1 PM–5 PM(kW)
24 t_5 = 4.0              // Time from 1 PM–5 PM(hour)
25 load_6 = 8500.0        // Load from 5 PM–7 PM(kW)
26 t_6 = 2.0              // Time from 5 PM–7 PM(hour)
27 load_7 = 10000.0       // Load from 7 PM–9 PM(kW)
28 t_7 = 2.0              // Time from 7 PM–9 PM(hour)
29 load_8 = 4500.0        // Load from 9 PM–11 PM(kW)
30 t_8 = 2.0              // Time from 9 PM–11 PM(hour)
```

```

31
32 // Calculations
33 energy_gen = (load_1*t_1)+(load_2*t_2)+(load_3*t_3)
   +(load_4*t_4)+(load_5*t_5)+(load_6*t_6)+(load_7*
   t_7)+(load_8*t_8) // Energy generated during 24
   hours (kWh)
34 LF = energy_gen/(MD*24.0)                                // Load factor
35 no_units = 3.0                                            // Number
   of generating set
36 cap_1 = 5000.0                                           // Capacity
   of first generating unit (kW)
37 cap_2 = 3000.0                                           // Capacity
   of second generating unit (kW)
38 cap_3 = 2000.0                                           // Capacity
   of third generating unit (kW)
39 cap_reserve = cap_1                                      // Reserve
   capacity (kW) i.e largest size of generating unit
40 cap_installed = cap_1+cap_2+cap_3+cap_reserve
   // Installed capacity (kW)
41 cap_factor = energy_gen/(cap_installed*24.0)
   // Plant capacity factor
42 cap_plant = cap_3*t_1+(cap_3+cap_2)*t_2+(cap_2+cap_1)
   *t_3+cap_2*t_4+(cap_2+cap_1)*t_5+(cap_3+cap_2+
   cap_1)*t_6+(cap_3+cap_2+cap_1)*t_7+cap_1*t_8 // Capacity of plant running actually (kWh)
43 use_factor = energy_gen/cap_plant
   // Plant use factor
44
45 // Results
46 disp("PART I - EXAMPLE : 7.9 : SOLUTION :-")
47 printf("\nNumber of generator units = %.f", no_units
   )

```

```

48 printf("\nSize of generator units required are %.f
      kW, %.f kW and %.f kW", cap_1, cap_2, cap_3)
49 printf("\nReserve plant capacity = %.f kW",
      cap_reserve)
50 printf("\nLoad factor = %.2f = %.f percent", LF, LF
      *100)
51 printf("\nPlant capacity factor = %.4f = %.2f
      percent", cap_factor, cap_factor*100)
52 printf("\nPlant use factor = %.3f = %.1f percent",
      use_factor, use_factor*100)
53 printf("\n\nNOTE: Capacity of plant is directly
      taken & operating schedule is not displayed here"
      )

```

---

### Scilab code Exa 7.10 Cost of generation per kWh at 100 and 50 percent load factor

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
   GENERATION
8
9 // EXAMPLE : 7.10 :
10 // Page number 76
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 cap_installed = 210.0*10**3           // Installed
   capacity of the station(kW)
15 capital_cost_kW = 1000.0              // Capital cost of
   station(Rs/kW)

```

```

16 fixed_cost_per = 0.13           // Fixed cost = 13
   % * cost of investment
17 variable_cost_per = 1.3         // Variable cost =
   1.3*fixed cost
18 LF_1 = 1.0                     // Load factor
19 LF_2 = 0.5                     // Load factor
20
21 // Calculations
22 MD = cap_installed
                           //
   Maximum demand (kW)
23 hours_year = 365.0*24
                           // Total
   hours in a year
24 capital_cost = capital_cost_kW*cap_installed
                           // Capital cost of station (Rs)
25 // Case(i) At 100% load factor
26 fixed_cost_1 = capital_cost*fixed_cost_per
                           // Fixed cost (Rs)
27 variable_cost_1 = variable_cost_per*fixed_cost_1
                           // Variable cost (Rs)
28 operating_cost_1 = fixed_cost_1+variable_cost_1
                           // Operating cost per annum (Rs)
29 units_gen_1 = LF_1*MD*hours_year
                           // Total units
   generated (kWh)
30 cost_gen_1 = operating_cost_1*100/units_gen_1
                           // Cost of generation per kWh (Paise
   )
31 // Case(ii) At 50% load factor
32 fixed_cost_2 = capital_cost*fixed_cost_per
                           // Fixed cost (Rs)
33 units_gen_2 = LF_2*MD*hours_year
                           // Total units
   generated (kWh)
34 variable_cost_2 = variable_cost_1*units_gen_2/
   units_gen_1 // Variable cost (Rs)
35 operating_cost_2 = fixed_cost_2+variable_cost_2

```

```

            // Operating cost per annum(Rs)
36 cost_gen_2 = operating_cost_2*100/units_gen_2
                // Cost of generation per kWh(Paise
)
37
38 // Results
39 disp("PART I - EXAMPLE : 7.10 : SOLUTION :-")
40 printf("\nCost of generation per kWh at 100 percent
    load factor = %.2f paise", cost_gen_1)
41 printf("\nCost of generation per kWh at 50 percent
    load factor = %.1f paise", cost_gen_2)
42 printf("\nComment: As the load factor is reduced ,
    cost of generation is increased\n")
43 printf("\nNOTE: ERROR: (1) In problem statement ,
    Capital cost of station must be Rs. 1000/kW, not
    Rs. 1000/MW")
44 printf("\n          (2) Calculation mistake in
    Total units generated in Case(i) in textbook")

```

---

### Scilab code Exa 7.11 Cost per unit generated

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
    GENERATION
8
9 // EXAMPLE : 7.11 :
10 // Page number 76
11 clear ; clc ; close ; // Clear the work space and
    console
12

```

```

13 // Given data
14 MD = 100.0*10**3 // Maximum
   demand(kW)
15 capital_cost = 200.0*10**6 // Capital cost(
   Rs)
16 LF = 0.4 // Annual load
   factor
17 cost_fueloil = 15.0*10**6 // Annual cost
   of fuel and oil(Rs)
18 cost_tax = 10.0*10**6 // Cost of taxes
   , wages and salaries(Rs)
19 interest = 0.15 // Interest and
   depreciation
20
21 // Calculations
22 hours_year = 365.0*24 // Total hours in a year
23 units_gen = MD*LF*hours_year // Units generated per annum(kWh)
24 fixed_charge = interest*capital_cost // Annual fixed charges(Rs)
25 running_charge = cost_fueloil+cost_tax // Annual running charges(Rs)
26 annual_charge = fixed_charge+running_charge // Total annual charges(Rs)
27 cost_unit = annual_charge*100/units_gen // Cost per unit(Paise)
28
29 // Results
30 disp("PART I - EXAMPLE : 7.11 : SOLUTION :-")
31 printf("\nCost per unit generated = %.f paise",
   cost_unit)

```

---

**Scilab code Exa 7.12 Minimum reserve capacity of station and Cost per kWh generated**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
     GENERATION
8
9 // EXAMPLE : 7.12 :
10 // Page number 76-77
11 clear ; clc ; close ; // Clear the work space and
      console
12
13 // Given data
14 cap_installed = 500.0          // Installed
      capacity of the station (MW)
15 CF = 0.45                     // Capacity factor
16 LF = 0.6                       // Annual laod
      factor
17 cost_fueloil = 10.0*10**7      // Annual cost of
      fuel , oil etc(Rs)
18 capital_cost = 10**9           // Capital cost (Rs)
19 interest = 0.15                // Interest and
      depreciation
20
21 // Calculations
22 // Case(i)
23 MD = cap_installed*CF/LF      // Maximum
      demand(MW)
24 cap_reserve = cap_installed-MD // Reserve capacity(
      MW)
25 // Case(ii)
26 hours_year = 365.0*24         // Total
      hours in a year

```

```

27 units_gen = MD*10**3*LF*hours_year
                           // Units generated per
                           annum(kWh)
28 fixed_charge = interest*capital_cost
                           // Annual fixed charges(Rs
                           )
29 running_charge = cost_fueloil
                           // Annual running
                           charges(Rs)
30 annual_charge = fixed_charge+running_charge
                           // Total annual charges(Rs)
31 cost_unit = annual_charge*100/units_gen
                           // Cost per kWh generated(
                           Paise)
32
33 // Results
34 disp("PART I - EXAMPLE : 7.12 : SOLUTION :-")
35 printf("\nCase(i) : Minimum reserve capacity of
                           station = %.f MW", cap_reserve)
36 printf("\nCase(ii) : Cost per kWh generated = %.f
                           paise", cost_unit)

```

---

**Scilab code Exa 7.13 Two part tariff to be charged from consumers**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
                           GENERATION
8
9 // EXAMPLE : 7.13 :
10 // Page number 77

```

```

11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 gen_expense = 850000.0           // Annual
    generation expense(Rs)
15 fuel_expense = 2800000.0         // Annual
    fuel expense(Rs)
16 trans_expense = 345000.0         // Annual
    transmission expense(Rs)
17 dist_expense = 2750000.0         // Annual
    distribution expense(Rs)
18 repair_expense = 300000.0        // Annual
    repairs , etc expense(Rs)
19 unit_gen = 600.0*10**6           // Number of
    units generated per year(kWh)
20 MD = 75.0*10**3                 // Maximum
    demand(kW)
21 gen = 0.9                        // Fixed
    charges for generation
22 fuel = 0.15                      // Fixed
    charges for fuel
23 transm = 0.85                    // Fixed
    charges for transmission
24 dist = 0.95                      // Fixed
    charges for distribution
25 repair = 0.5                     // Fixed
    charges for repairs , etc
26 loss_dist = 0.2                  // Losses in
    transmission and distribution
27
28 // Calculations
29 fixed_gen = gen_expense*gen      //
    Fixed charge on generation(Rs)
30 running_gen = gen_expense*(1-gen) //
    Running charge on generation(Rs)
31 fixed_fuel = fuel_expense*fuel   //
    Fixed charge on fuel(Rs)

```

```

32 running_fuel = fuel_expense*(1-fuel) //  

    Running charge on fuel(Rs)  

33 fixed_trans = trans_expense*transm //  

    Fixed charge on transmission(Rs)  

34 running_trans = trans_expense*(1-transm) //  

    Running charge on transmission(Rs)  

35 fixed_dist = dist_expense*dist //  

    Fixed charge on distribution(Rs)  

36 running_dist = dist_expense*(1-dist) //  

    Running charge on distribution(Rs)  

37 fixed_repair = repair_expense*repair //  

    Fixed charge on repairs ,etc(Rs)  

38 running_repair = repair_expense*(1-repair) //  

    Running charge on repairs ,etc(Rs)  

39 fixed_charge = fixed_gen+fixed_fuel+fixed_trans+  

    fixed_dist+fixed_repair // Total  

    fixed charges(Rs)  

40 running_charge = running_gen+running_fuel+  

    running_trans+running_dist+running_repair //  

    Total running charges(Rs)  

41 fixed_unit = fixed_charge/MD //  

    Fixed charges per unit(Rs)  

42 units_dist = unit_gen*(1-loss_dist) //  

    Total number of units distributed(kWh)  

43 running_unit = running_charge*100/units_dist //  

    Running charges per unit(Paise)  

44  

45 // Results  

46 disp("PART I - EXAMPLE : 7.13 : SOLUTION :-")  

47 printf("\nTwo part tariff is Rs %.3f per kW of  

    maximum demand plus %.3f paise per kWh",  

    fixed_unit,running_unit)

```

---

Scilab code Exa 7.14 Generation cost in two part form

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
     GENERATION
8
9 // EXAMPLE : 7.14 :
10 // Page number 77
11 clear ; clc ; close ; // Clear the work space and
      console
12
13 // Given data
14 cap_installed = 100.0*10**3           // Installed
      capacity of the station (kW)
15 capital_cost_kW = 1000.0              // Capital
      cost(Rs/kW)
16 depreciation = 0.15                  // Annual
      depreciation charge
17 royalty_kW = 2.0                     // Royalty per
      kW per year(Rs)
18 royalty_kWh = 0.03                   // Royalty per
      kWh per year(Rs)
19 MD = 70.0*10**3                     // Maximum
      demand(kW)
20 LF = 0.6                           // Annual load
      factor
21 cost_salary = 1000000.0             // Annual cost
      of salaries , maintenance charges etc(Rs)
22 cost_salary_per = 0.2               // Annual cost
      of salaries , maintenance charges etc charged as
      fixed charges
23
24 // Calculations
25 hours_year = 365.0*24
      //

```

```

    Total hours in a year
26 unit_gen = MD*LF*hours_year
                                // Units
        generated/annum(kWh)
27 capital_cost = cap_installed*capital_cost_kW
                                // Capital cost of plant(Rs)
28 depreciation_charge = depreciation*capital_cost
                                // Depreciation charges(Rs)
29 salary_charge = cost_salary_per*cost_salary
                                // Cost on salaries ,
        maintenance etc(Rs)
30 fixed_charge = depreciation_charge+salary_charge
                                // Total annual fixed charges(Rs)
31 cost_kW_fixed = (fixed_charge/MD)+royalty_kW
                                // Cost per kW(Rs)
32 salary_charge_running = (1-cost_salary_per)*
        cost_salary           // Annual running charge on
        salaries , maintenance etc(Rs)
33 cost_kWh_running = (salary_charge_running/unit_gen)+
        royalty_kWh // Cost per kWh(Rs)
34
35 // Results
36 disp("PART I - EXAMPLE : 7.14 : SOLUTION :-")
37 printf("\nGeneration cost in two part form is given
        by, Rs. (%.2f*kW + %.3f*kWh) ", cost_kW_fixed,
        cost_kWh_running)

```

---

**Scilab code Exa 7.15 Overall generating cost per unit at 50 and 100 percent capaci**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION

```

```

7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
     GENERATION
8
9 // EXAMPLE : 7.15 :
10 // Page number 78
11 clear ; clc ; close ; // Clear the work space and
     console
12
13 // Given data
14 cap_installed = 100.0*10**3 // Installed capacity
     of station(kW)
15 cost_gen = 30.0             // Generating cost per
     annum(Rs/kW)
16 cost_fixed = 4000000.0      // Fixed cost per annum
     (Rs)
17 cost_fuel = 60.0            // Cost of fuel(Rs/
     tonne)
18 calorific = 5700.0          // Calorific value of
     fuel(kcal/kg)
19 rate_heat_1 = 2900.0         // Plant heat rate at
     100% capacity factor(kcal/kWh)
20 CF_1 = 1.0                  // Capacity factor
21 rate_heat_2 = 4050.0         // Plant heat rate at
     50% capacity factor(kcal/kWh)
22 CF_2 = 0.5                  // Capacity factor
23
24 // Calculations
25 cost_fixed_kW = cost_fixed/cap_installed
     // Fixed cost per kW(Rs)
26 cost_fixed_total = cost_gen+cost_fixed_kW
     // Fixed cost per kW capacity(Rs)
27 average_demand_1 = CF_1*cap_installed
     // Average demand at 100% capacity factor(kW)
28 average_demand_2 = CF_2*cap_installed
     // Average demand at 50% capacity factor(kW)
29 hours_year = 365.0*24
     // Total hours in a year
30 unit_gen_1 = CF_1*hours_year

```

```

    // Energy generated per annum with average demand
    // of 1 kW(kWh)
31 unit_gen_2 = CF_2*hours_year
    // Energy generated per annum with average demand
    // of 0.5 kW(kWh)
32 cost_kWh_fixed_1 = cost_fixed_total*100/unit_gen_1
    // Cost per kWh due to fixed charge with 100% CF(
    // Paise)
33 cost_kWh_fixed_2 = cost_fixed_total*100/unit_gen_2
    // Cost per kWh due to fixed charge with 50% CF(
    // Paise)
34 kg_kWh_1 = rate_heat_1/calorific
    // Weight(kg)
35 kg_kWh_2 = rate_heat_2/calorific
    // Weight(kg)
36 cost_coal_1 = kg_kWh_1*cost_fuel*100/1000.0
    // Cost due to coal at 100% CF(Paise/kWh)
37 cost_coal_2 = kg_kWh_2*cost_fuel*100/1000.0
    // Cost due to coal at 50% CF(Paise/kWh)
38 cost_total_1 = cost_kWh_fixed_1+cost_coal_1
    // Total cost per unit with 100% CF(Paise)
39 cost_total_2 = cost_kWh_fixed_2+cost_coal_2
    // Total cost per unit with 50% CF(Paise)
40
41 // Results
42 disp("PART I - EXAMPLE : 7.15 : SOLUTION :-")
43 printf("\nOverall generating cost per unit at 100
    percent capacity factor = %.3f paise",
    cost_total_1)
44 printf("\nOverall generating cost per unit at 50
    percent capacity factor = %.3f paise\n",
    cost_total_2)
45 printf("\nNOTE: Slight changes in obtained answer
    from that of textbook answer is due to more
    precision here")

```

---

**Scilab code Exa 7.16 Yearly cost per kW demand and Cost per kWh supplied at substa**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
    GENERATION
8
9 // EXAMPLE : 7.16 :
10 // Page number 78
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 MD = 75.0*10**3 // Maximum
    demand(kW)
15 LF = 0.4 // Yearly load
    factor
16 cost_capital = 60.0 // Capital
    cost(Rs/annum/kW)
17 cost_kWh = 1.0 // Cost per
    kWh transmitted(Paise)
18 charge_trans = 2000000.0 // Annual
    capital charge for transmission(Rs)
19 charge_dist = 1500000.0 // Annual
    capital charge for distribution(Rs)
20 diversity_trans = 1.2 // Diversity
    factor for transmission
21 diversity_dist = 1.25 // Diversity
    factor for distribution
22 n_trans = 0.9 // Efficiency
```

```

        of transmission system
23 n_dist = 0.85                                // Efficiency
        of distribution system
24
25 // Calculations
26 // Case(a)
27 capital_cost = cost_capital*MD
                // Annual capital
                cost(Rs)
28 fixed_charge_sub = capital_cost+charge_trans
                // Total fixed charges for supply
                to substation per annum(Rs)
29 sum_MD_sub = MD*diversity_trans
                // Sum of all maximum
                demand of substation(kW)
30 cost_kW_sub = fixed_charge_sub/sum_MD_sub
                // Yearly cost per kW demand at
                substation(Rs)
31 running_cost_unit_sub = 1/n_trans
                // Running cost per
                unit supplied at substation(Paise)
32 // Case(b)
33 sum_MD_con = sum_MD_sub*diversity_dist
                // Sum of all maximum demand
                of consumer(kW)
34 fixed_charge_con = capital_cost+charge_trans+
                charge_dist // Total fixed charges for supply
                to consumers(Rs)
35 cost_kW_con = fixed_charge_con/sum_MD_con
                // Yearly cost per kW demand on
                consumer premises(Rs)
36 running_cost_unit_con = running_cost_unit_sub/n_dist
                // Running cost per unit supplied to
                consumer(Paise)
37
38 // Results
39 disp("PART I - EXAMPLE : 7.16 : SOLUTION :-")
40 printf("\nCase(a): Yearly cost per kW demand at the

```

```

        substations = Rs. %.2f ", cost_kW_sub)
41 printf("\n          Cost per kWh supplied at the
        substations = %.2f paise\n",
        running_cost_unit_sub)
42 printf("\nCase(b): Yearly cost per kW demand at the
        consumer premises = Rs. %.2f ", cost_kW_con)
43 printf("\n          Cost per kWh supplied at the
        consumer premises = %.3f paise",
        running_cost_unit_con)

```

---

**Scilab code Exa 7.17 Number of working hours per week above which the HV supply is**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
    GENERATION
8
9 // EXAMPLE : 7.17 :
10 // Page number 79
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 kVA_tariff_hv = 60.0           // HV supply per kVA per
    annum(Rs)
15 kWh_tariff_hv = 3.0/100        // HV supply per kWh
    annum(Rs)
16 kVA_tariff_lv = 65.0           // LV supply per kVA per
    annum(Rs)
17 kWh_tariff_lv = 3.3/100        // LV supply per kWh
    annum(Rs)

```

```

18 cost_equip_kVA = 50.0           // Cost of transformers
   and switchgear per kVA(Rs)
19 loss_full_load = 0.02          // Full load
   transformation loss
20 fixed_charge_per = 0.2         // Fixed charges per
   annum
21 no_week = 50.0                 // Number of working
   weeks in a year
22
23 // Calculations
24 rating_equip = 1000/(1-loss_full_load)           //
   Rating of transformer and switchgear(kVA)
25 cost_equip = cost_equip_kVA*rating_equip           //
   Cost of transformers and switchgear(Rs)
26 fixed_charge = fixed_charge_per*cost_equip           //
   Fixed charges per annum on HV plant(Rs)
27 X = poly(0,"X")                                     //
   Number of working hours per week
28 units_consumed = (no_week*X)*1000.0                //
   Yearly units consumed by load
29 total_units = units_consumed/(1-loss_full_load)  //
   Total units to be paid on HV supply
30 // Case(a)
31 annual_cost_hv = (kVA_tariff_hv*rating_equip)+(    // Annual cost(Rs)
   kWh_tariff_hv*cost_equip*X)+fixed_charge
32 // Case(b)
33 annual_cost_lv = (kVA_tariff_lv*1000.0)+(        // Annual cost(Rs)
   kWh_tariff_lv*units_consumed)
34 p = annual_cost_hv-annual_cost_lv                  //
   Finding unknown value i.e working hours in terms
   of X
35 x = roots(p)                                      //
   Finding unknown value i.e working hours
36
37 // Results

```

```
38 disp("PART I – EXAMPLE : 7.17 : SOLUTION :-")
39 printf("\nAbove %.1f working hours per week the H.V
    supply is cheaper ", x)
```

---

**Scilab code Exa 7.18** Cheaper alternative to adopt and by how much

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
    GENERATION
8
9 // EXAMPLE : 7.18 :
10 // Page number 79–80
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 load_1 = 10.0*10***3          // Load per annum(kVA)
15 time_1 = 1800.0                // Time(hours)
16 load_2 = 6.0*10***3           // Load per annum(kVA)
17 time_2 = 600.0                 // Time(hours)
18 load_3 = 0.25*10***3          // Load per annum(kVA)
19 time_3 = 400.0                 // Time(hours)
20 rating_trans = 10.0*10***3    // Transformer rating(kVA
    )
21 pf = 0.8                      // Lagging power factor
22 n_f1_A = 98.3/100.0           // Full load efficiency
    of transformer A
23 n_f1_B = 98.8/100.0           // Full load efficiency
    of transformer B
24 loss_A = 70.0                  // Core loss at rated
```

```

        voltage of transformer A(kW)
25 loss_B = 40.0           // Core loss at rated
        voltage of transformer B(kW)
26 cost_A = 250000.0       // Cost of transformer A(
        Rs)
27 cost_B = 280000.0       // Cost of transformer B(
        Rs)
28 interest_per = 0.1      // Interest and
        depreciation charges
29 cost_energy_unit = 3.0   // Energy costs per unit(
        Paise)
30
31 // Calculations
32 // Transformer A
33 output_A = rating_trans*pf
                           // kW output at full
        load(kW)
34 input_A = output_A/n_fl_A
                           // Input at full
        load(kW)
35 cu_loss_fl_A = input_A-output_A-loss_A
                           // Copper loss at full load(kW)
36 cu_loss_2_A = (load_2/load_1)**2*cu_loss_fl_A
                           // Copper loss at 6 MVA output(kW)
37 cu_loss_3_A = (load_3/load_1)**2*cu_loss_fl_A
                           // Copper loss at 0.25 MVA output(kW)
38 ene_iron_loss_A = loss_A*(time_1+time_2+time_3)
                           // Energy consumed due to iron losses(kWh)
39 ene_cu_loss_A = time_1*cu_loss_fl_A+time_2*
        cu_loss_2_A+time_3*cu_loss_3_A // Energy
        consumed due to copper losses(kWh)
40 total_loss_A = ene_iron_loss_A+ene_cu_loss_A
                           // Total loss per annum(kWh)
41 cost_energy_A = cost_energy_unit/100*total_loss_A
                           // Energy cost per annum due to losses(Rs)
42 // Transformer B
43 output_B = rating_trans*pf
                           // kW output at full

```

```

        load (kW)
44 input_B = output_B/n_fl_B
                           // Input at full
        load (kW)
45 cu_loss_fl_B = input_B-output_B-loss_B
                           // Copper loss at full load (kW)
46 cu_loss_2_B = (load_2/load_1)**2*cu_loss_fl_B
                           // Copper loss at 6 MVA output (kW)
47 cu_loss_3_B = (load_3/load_1)**2*cu_loss_fl_B
                           // Copper loss at 0.25 MVA output (kW)
48 ene_iron_loss_B = loss_B*(time_1+time_2+time_3)
                           // Energy consumed due to iron losses (kWh)
49 ene_cu_loss_B = time_1*cu_loss_fl_B+time_2*
                           cu_loss_2_B+time_3*cu_loss_3_B // Energy
                           consumed due to copper losses (kWh)
50 total_loss_B = ene_iron_loss_B+ene_cu_loss_B
                           // Total loss per annum (kWh)
51 cost_energy_B = cost_energy_unit/100*total_loss_B
                           // Energy cost per annum due to losses (Rs)
52 diff_capital = cost_B-cost_A
                           // Difference in
                           capital costs (Rs)
53 annual_charge = interest_per*diff_capital
                           // Annual charge due to this amount (
                           Rs)
54 diff_cost_energy = cost_energy_A-cost_energy_B
                           // Difference in energy cost per annum (Rs
                           )
55 cheap = diff_cost_energy-annual_charge
                           // Cheaper in cost (Rs)
56
57 // Results
58 disp("PART I – EXAMPLE : 7.18 : SOLUTION :-")
59 printf("\nTransformer B is cheaper by Rs. %.f per
                           year \n", cheap)
60 printf("\nNOTE: ERROR: Full load efficiency for
                           transformer B is 98.8 percent , not 98.3 percent
                           as given in problem statement")

```

```
61 printf("\n      Changes in obtained answer from that  
      of textbook answer is due to more precision")
```

---

Scilab code Exa 7.19 Valuation halfway based on Straight line Reducing balance and

```
1 // A Texbook on POWER SYSTEM ENGINEERING  
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  
3 // DHANPAT RAI & Co.  
4 // SECOND EDITION  
5  
6 // PART I : GENERATION  
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER  
    GENERATION  
8  
9 // EXAMPLE : 7.19 :  
10 // Page number 80–81  
11 clear ; clc ; close ; // Clear the work space and  
    console  
12  
13 // Given data  
14 fixed_cost = 4.0*10**4 // Fixed cost of  
    plant(Rs)  
15 salvage_value = 4.0*10**3 // Salvage value(Rs)  
16 n = 20.0 // Useful life (years)  
17 r = 0.06 // Sinking fund  
    depreciation compounded annually  
18  
19 // Calculations  
20 n_2 = n/2 //  
    Halfway of useful life (years)  
21 // Case(a)  
22 total_dep_A = fixed_cost-salvage_value //  
    Total depreciation in 20 years(Rs)  
23 dep_10_A = total_dep_A/2 //  
    Depreciation in 10 years(Rs)
```

```

24 value_10_A = fixed_cost-dep_10_A           //  

    Value at the end of 10 years(Rs)  

25 // Case(b)  

26 P_B = fixed_cost                         //  

    Capital outlay(Rs)  

27 q_B = (salvage_value/fixed_cost)**(1/n)    // q =  

    (1-p)  

28 value_10_B = P_B*(q_B)**n_2              //  

    Value at the end of 10 years(Rs)  

29 // Case(c)  

30 P_C = fixed_cost                         //  

    Capital cost of plant(Rs)  

31 P_C = salvage_value                      //  

    Scrap value(Rs)  

32 Q_C = P_C-P__C                          // Cost  

    of replacement(Rs)  

33 q_C = Q_C/(((1+r)**n-1)/r)               //  

    Yearly charge(Rs)  

34 amount_dep = q_C*((1+r)**n_2-1)/r        //  

    Amount deposited at end of 10 years(Rs)  

35 value_10_C = P_C-amount_dep             //  

    Value at the end of 10 years(Rs)  

36  

37 // Results  

38 disp("PART I - EXAMPLE : 7.19 : SOLUTION :-")  

39 printf("\nCase(a): Valuation halfway through its  

    life based on Straight line depreciation method =  

    Rs %.1e ", value_10_A)  

40 printf("\nCase(b): Valuation halfway through its  

    life based on Reducing balance depreciation  

    method = Rs %.2e ", value_10_B)  

41 printf("\nCase(c): Valuation halfway through its  

    life based on Sinking fund depreciation method =  

    Rs %.2e ", value_10_C)

```

---

**Scilab code Exa 7.20** Type and hp ratings of two turbines for the station

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
GENERATION
8
9 // EXAMPLE : 7.20 :
10 // Page number 81
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 h = 30.0 // Mean head(m)
15 area_catch = 250.0 // Catchment area(Square km
)
16 average_rain = 1.25 // Average rainfall per
annum(m)
17 utilized_rain = 0.7 // Rainfall utilized
18 LF = 0.8 // Expected load factor
19 n_turbine = 0.9 // Mechanical efficiency of
turbine
20 n_gen = 0.95 // Efficiency of generator
21
22 // Calculations
23 water_avail = utilized_rain*area_catch*10**6*
average_rain // Water available(m^3)
24 sec_year = 365.0*24*60*60 // Total
seconds in a year
25 Q = water_avail/sec_year // Quantity
available per second(m^3) i.e Discharge(m^3/sec)
26 w = 1000.0
```

```

    // Density of water(kg/m^3)
27 n = n_turbine*n_gen
                                //
                                Overall efficiency
28 P = 0.736/75*Q*w*h*n
                                //
                                Average output of generator units(kW)
29 rating_gen = P/LF
                                //
                                Rating of generator(kW)
30 rating_gen_each = rating_gen/2.0
                                // Rating of each
                                generator(kW)
31 rating_turbine = rating_gen/2*(1/(0.736*n_gen))
                                // Rating of each turbine(metric hp
)
32
33 // Results
34 disp("PART I - EXAMPLE : 7.20 : SOLUTION :-")
35 printf("\nChoice of units are:")
36 printf("\n 2 generators each having maximum rating
      of %.f kW ", rating_gen_each)
37 printf("\n 2 propeller turbines each having maximum
      rating of %.f metric hp \n", rating_turbine)
38 printf("\nNOTE: Changes in obtained answer from that
      of textbook answer is due to more precision here
      ")

```

---

**Scilab code Exa 7.21 Plot of chronological load curve and Load duration curve**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar

```

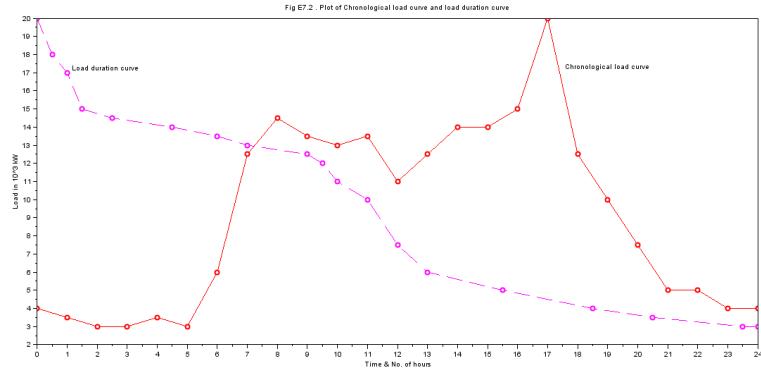


Figure 7.1: Plot of chronological load curve and Load duration curve

```

3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
     GENERATION
8
9 // EXAMPLE : 7.21 :
10 // Page number 81–82
11 clear ; clc ; close ; // Clear the work space and
     console
12
13 // Given data
14 t0 = 0.0                      // Time 12 morning
15 l0 = 4.0                       // Load at 12 morning(kW
     *1000)
16 t1 = 1.0                       // Time 1 a.m
17 l1 = 3.5                       // Load at 1 a.m(kW*1000)
18 t2 = 2.0                       // Time 2 a.m
19 l2 = 3.0                       // Load at 2 a.m(kW*1000)
20 t3 = 3.0                       // Time 3 a.m
21 l3 = 3.0                       // Load at 3 a.m(kW*1000)
22 t4 = 4.0                       // Time 4 a.m

```

```

23 t4 = 3.5 // Load at 4 a.m(kW*1000)
24 t5 = 5.0 // Time 5 a.m
25 t5 = 3.0 // Load at 5 a.m(kW*1000)
26 t6 = 6.0 // Time 6 a.m
27 t6 = 6.0 // Load at 6 a.m(kW*1000)
28 t7 = 7.0 // Time 7 a.m
29 t7 = 12.5 // Load at 7 a.m(kW*1000)
30 t8 = 8.0 // Time 8 a.m
31 t8 = 14.5 // Load at 8 a.m(kW*1000)
32 t9 = 9.0 // Time 9 a.m
33 t9 = 13.5 // Load at 9 a.m(kW*1000)
34 t10 = 10.0 // Time 10 a.m
35 t10 = 13.0 // Load at 10 a.m(kW*1000)
36 t11 = 11.0 // Time 11 a.m
37 t11 = 13.5 // Load at 11 a.m(kW*1000)
38 t113 = 11.50 // Time 11.30 a.m
39 t113 = 12.0 // Load at 11.30 am(kW
                  *1000)
40 t12 = 12.0 // Time 12 noon
41 t12 = 11.0 // Load at 12 noon(kW*1000)
42 t123 = 12.50 // Time 12.30 noon
43 t123 = 5.0 // Load at 12.30 noon(kW
                  *1000)
44 t13 = 13.0 // Time 1 p.m
45 t13 = 12.5 // Load at 1 p.m(kW*1000)
46 t133 = 13.50 // Time 1.30 p.m
47 t133 = 13.5 // Load at 1.30 p.m(kW
                  *1000)
48 t14 = 14.0 // Time 2 p.m
49 t14 = 14.0 // Load at 2 p.m(kW*1000)
50 t15 = 15.0 // Time 3 p.m
51 t15 = 14.0 // Load at 3 p.m(kW*1000)
52 t16 = 16.0 // Time 4 p.m
53 t16 = 15.0 // Load at 4 p.m(kW*1000)
54 t163 = 16.50 // Time 4.30 p.m
55 t163 = 18.0 // Load at 4.30 p.m(kW
                  *1000)
56 t17 = 17.0 // Time 5 p.m

```

```

57 l17 = 20.0 // Load at 5 p.m(kW*1000)
58 t173 = 17.50 // Time 5.30 p.m
59 l173 = 17.0 // Load at 5.30 p.m(kW
    *1000)
60 t18 = 18.0 // Time 6 p.m
61 l18 = 12.5 // Load at 6 p.m(kW*1000)
62 t19 = 19.0 // Time 7 p.m
63 l19 = 10.0 // Load at 7 p.m(kW*1000)
64 t20 = 20.0 // Time 8 p.m
65 l20 = 7.5 // Load at 8 p.m(kW*1000)
66 t21 = 21.0 // Time 9 p.m
67 l21 = 5.0 // Load at 9 p.m(kW*1000)
68 t22 = 22.0 // Time 10 p.m
69 l22 = 5.0 // Load at 10 p.m(kW*1000)
70 t23 = 23.0 // Time 11 p.m
71 l23 = 4.0 // Load at 11 p.m(kW*1000)
72 t24 = 24.0 // Time 12 morning
73 l24 = 4.0 // Load at 12 morning (kW
    *1000)
74
75 // Calculations
76 t = [t0,t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,
    t14,t15,t16,t17,t18,t19,t20,t21,t22,t23,t24]
77 l = [10,11,12,13,14,15,16,17,18,19,110,111,112,113,
    114,115,116,117,118,119,120,121,122,123,124]
78 a = gca(); // sets thickness of plot
79 a.thickness = 2
80 plot(t,l,'ro-') // Plot of Chronological load curve
81 T =
    [0,0.5,1,1.5,2.5,4.5,6,7,9,9.5,10,11,12,13,15.5,18.5,20.5,23.5,24]
    // Solved time
82 L =
    [20,18,17,15,14.5,14,13.5,13,12.5,12,11,10,7.5,6,5,4,3.5,3,3]
    // Solved load
83 plot(T,L,'--mo') // Plot of load duration curve

```

```

84 a.x_label.text = 'Time & No. of hours '
    // labels x-axis
85 a.y_label.text = 'Load in 10^3 kW'
    // labels y-axis
86 xtitle("Fig E7.2 . Plot of Chronological load curve
        and load duration curve")
87 xset('thickness',2)
    // sets thickness of axes
88 xstring(17.5,17,'Chronological load curve')
89 xstring(1.1,17,'Load duration curve')
90
91 // Results
92 disp("PART I – EXAMPLE : 7.21 : SOLUTION :-")
93 printf("\nThe chronological load curve and the load
        duration curve is shown in the Figure E7.2\n")
94 printf("\nNOTE: The time is plotted in 24 hours
        format")

```

---

### Scilab code Exa 7.22 Daily energy produced Reserve capacity and Maximum energy pro

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
        GENERATION
8
9 // EXAMPLE : 7.22 :
10 // Page number 82
11 clear ; clc ; close ; // Clear the work space and
        console
12
13 // Given data

```

```

14 MD = 20.0*10**3 // Maximum demand (kW)
15 LF = 0.6 // Load factor
16 CF = 0.48 // Plant capacity factor
17 UF = 0.8 // Plant use factor
18
19 // Calculations
20 // Case(a)
21 avg_demand = LF*MD // Average demand (kW)
22 ene_daily = avg_demand*24.0 // Daily energy produced (kWh)
23 // Case(b)
24 cap_installed = avg_demand/CF // Installed capacity (kW)
25 cap_reserve = cap_installed-MD // Reserve capacity (kW)
26 // Case(c)
27 max_ene_C = cap_installed*24.0 // Maximum energy that could be produced daily (kWh)
28 // Case(d)
29 max_ene_D = ene_daily/UF // Maximum energy that could be produced daily as per schedule (kWh)
30
31 // Results
32 disp("PART I - EXAMPLE : 7.22 : SOLUTION :-")
33 printf("\nCase(a): Daily energy produced = %.f kWh" , ene_daily)
34 printf("\nCase(b): Reserve capacity of plant = %.f kW" , cap_reserve)
35 printf("\nCase(c): Maximum energy that could be produced daily when plant runs at all time = %.f kWh" , max_ene_C)
36 printf("\nCase(d): Maximum energy that could be produced daily when plant runs fully loaded = %.f

```

kWh" , max\_ene\_D)

---

**Scilab code Exa 7.23** Rating Annual energy produced Total fixed and variable cost C

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
GENERATION
8
9 // EXAMPLE : 7.23 :
10 // Page number 83–84
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 cap_3sets = 600.0 // Capacity of 3
generators (kW)
15 no_3 = 3.0 // Number of sets
of 600 kW
16 cap_4thset = 400.0 // Capacity of 4th
generator set (kW)
17 no_4 = 1.0 // Number of sets
of 400 kW
18 MD = 1600.0 // Maximum demand (
kW)
19 LF = 0.45 // Load factor
20 cost_capital_kW = 1000.0 // Capital cost
per kW installed capacity (Rs)
21 cost_annual_per = 0.15 // Annual cost =
15% of capital cost
22 cost_operation = 60000.0 // Annual
```

```

        operation cost (Rs)
23 cost_maintenance = 30000.0           // Annual
        maintenance cost (Rs)
24 fixed_maintenance = 1.0/3            // Fixed cost
25 variable_maintenance = 2.0/3          // Variable cost
26 cost_fuel_kg = 40.0/100             // Cost of fuel
        oil (Rs/kg)
27 cost_oil_kg = 1.25                  // Cost of
        lubricating oil (Rs/kg)
28 calorific = 10000.0                  // Calorific value
        of fuel (kcal/kg)
29 oil_consum = 1.0/400                // Consumption of
        lubricating oil. 1kg for every 400kWh generated
30 fuel_consum = 1.0/2                 // Consumption of
        fuel. 1kg for every 2kWh generated
31 n_gen = 0.92                      // Generator
        efficiency
32 heat_lost = 1.0/3                  // Heat lost in
        the fuel to cooling water
33 theta = 11.0                       // Difference of
        temperature between inlet and outlet( C )
34
35 // Calculations
36 // Case(a)
37 rating_3set_A = cap_3sets/n_gen
        //
        Rating of first 3 sets (kW)
38 rating_4th_A = cap_4thset/n_gen
        //
        Rating of 4th set (kW)
39 // Case(b)
40 avg_demand_B = LF*MD
        //
        Average demand (kW)
41 hours_year = 365.0*24
        //
        Total hours in a year
42 energy_B = avg_demand_B*hours_year

```

```

        // Annual energy produced (kWh)
43 // Case(c)
44 total_invest = (no_3*cap_3sets+cap_4thset*no_4)*
    cost_capital_kW // Total
    investment(Rs)
45 annual_cost = cost_annual_per*total_invest
    // Annual
    cost(Rs)
46 maintenance_cost = fixed_maintenance*
    cost_maintenance // Maintenance cost(Rs)
47 fixed_cost_total = annual_cost+maintenance_cost
    // Total fixed
    cost per annum(Rs)
48 fuel_consumption = energy_B*fuel_consum
    // Fuel
    consumption(Kg)
49 cost_fuel = fuel_consumption*cost_fuel_kg
    // Cost of
    fuel(Rs)
50 oil_consumption = energy_B*oil_consum
    // Lubrication oil consumption(Kg)
51 cost_oil = oil_consumption*cost_oil_kg
    // Cost
    of Lubrication oil(Rs)
52 var_maintenance_cost = variable_maintenance*
    cost_maintenance // Variable
    part of maintenance cost(Rs)
53 variable_cost_total = cost_fuel+cost_oil+
    var_maintenance_cost+cost_operation // Total
    variable cost per annum(Rs)
54 cost_total_D = fixed_cost_total+variable_cost_total
    // Total cost per
    annum(Rs)
55 cost_kWh_gen = cost_total_D/energy_B*100
    // Cost per

```

```

        kWh generated(Paise)
56 // Case(c)
57 n_overall = energy_B*860/(fuel_consumption*calorific
    )*100                                // Overall efficiency (
    %)
58 // Case(d)
59 weight_water_hr = heat_lost*fuel_consumption/(
    hours_year*theta)*calorific           // Weight of
    cooling water required(kg/hr)
60 weight_water_min = weight_water_hr/60.0
                                         // Weight
    of cooling water required(kg/min)
61 capacity_pump = weight_water_min*MD/avg_demand_B
                                         // Capacity of
    cooling water pump(kg/min)
62
63 // Results
64 disp("PART I - EXAMPLE : 7.23 : SOLUTION :-")
65 printf("\nCase(a): Rating of first 3 sets of diesel
    engine = %.f kW", rating_3set_A)
66 printf("\n          Rating of 4th set of diesel
    engine = %.f kW", rating_4th_A)
67 printf("\nCase(b): Annual energy produced = %.1e kWh
    ", energy_B)
68 printf("\nCase(c): Total fixed cost = Rs %.f ",
    fixed_cost_total)
69 printf("\n          Total variable cost = Rs %.f ",
    variable_cost_total)
70 printf("\n          Cost per kWh generated = %.f
    paise", cost_kWh_gen)
71 printf("\nCase(d): Overall efficiency of the diesel
    plant = %.1f percent", n_overall)
72 printf("\nCase(e): Quantity of cooling water
    required per round = %.2e kg/hr = %.f kg/min",
    weight_water_hr, weight_water_min)
73 printf("\n          Capacity of cooling-water pumps
    under maximum load = %.f kg/min \n",
    capacity_pump)

```

```
74 printf("\nNOTE: Changes in obtained answer from that  
      of textbook answer is due to more precision here  
'')
```

---

### Scilab code Exa 7.24 Turbine rating Energy produced Average steam consumption Evap

```
1 // A Texbook on POWER SYSTEM ENGINEERING  
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  
3 // DHANPAT RAI & Co.  
4 // SECOND EDITION  
5  
6 // PART I : GENERATION  
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER  
    GENERATION  
8  
9 // EXAMPLE : 7.24 :  
10 // Page number 84  
11 clear ; clc ; close ; // Clear the work space and  
    console  
12  
13 // Given data  
14 cap_installed = 30.0*10***3           // Rating of each  
    generators (kW)  
15 no = 4.0                                // Number of  
    installed generators  
16 MD = 100.0*10***3                      // Maximum demand (kW  
    )  
17 LF = 0.8                                // Load factor  
18 cost_capital_kW = 800.0                  // Capital cost per  
    kW installed capacity (Rs)  
19 depreciation_per = 0.125                 // Depreciation , etc  
    = 12.5% of capital cost  
20 cost_operation = 1.2*10***6            // Annual operation  
    cost (Rs)  
21 cost_maintenance = 600000.0           // Annual
```

```

        maintenance cost (Rs)
22 fixed_maintenance = 1.0/3           // Fixed cost
23 variable_maintenance = 2.0/3         // Variable cost
24 cost_miscellaneous = 100000.0
    cost (Rs)                         // Miscellaneous
25 cost_fuel_kg = 32.0/1000          // Cost of fuel oil (
    Rs/kg)                           // Rs/kg
26 calorific = 6400.0
    of fuel (kcal/kg)                // Calorific value
27 n_gen = 0.96
    efficiency                      // Generator
28 n_thermal = 0.28
    efficiency of turbine           // Thermal
29 n_boiler = 0.75
30 n_overall = 0.2
    efficiency                       // Boiler efficiency
                                    // Overall thermal
efficiency
31
32 // Calculations
33 // Case(a)
34 rating_turbine = cap_installed/(n_gen*0.736)
    // Rating of each steam
    turbine (metric hp)
35 // Case(b)
36 avg_demand_B = LF*MD
    // Average demand (kW)
37 hours_year = 365.0*24
    // Total hours in a year
38 energy_B = avg_demand_B*hours_year
    // Annual energy
    produced (kWh)
39 // Case(c)
40 steam_consumption_C = (0.8+3.5*LF)/LF
    // Average steam
    consumption (kg/kWh)
41 // Case(d)
42 LF_D = 1.0

```

```

        // Assumption that Load factor for boiler
43 steam_consumption_D = (0.8+3.5*LF_D)/LF_D
                           // Steam consumption(kg/kWh)
        )
44 energy_D = cap_installed*1.0
                           // Energy
        output per hour per set(kWh)
45 evaporation_cap = steam_consumption_D*energy_D
                           // Evaporation capacity of
        boiler(kg/hr)
46 // Case(e)
47 total_invest = no*cap_installed*cost_capital_kW
                           // Total investment(Rs)
48 capital_cost = depreciation_per*total_invest
                           // Capital cost(Rs)
49 maintenance_cost = fixed_maintenance*
        cost_maintenance           // Maintenance cost(Rs)
        )
50 fixed_cost_total = capital_cost+maintenance_cost
                           // Total fixed cost per annum(Rs)
51 var_maintenance_cost = variable_maintenance*
        cost_maintenance           // Variable part of
        maintenance cost(Rs)
52 input_E = energy_B/n_overall
                           // Input into
        system per annum(kWh)
53 weight_fuel = input_E*860/calorific
                           // Weight of fuel(kg)
54 cost_fuel = weight_fuel*cost_fuel_kg
                           // Cost of fuel(Rs)
55 variable_cost_total = cost_operation+
        var_maintenance_cost+cost_miscellaneous+cost_fuel
                           // Total variable cost per annum(Rs)
56 cost_total_E = fixed_cost_total+variable_cost_total
                           // Total cost per annum(Rs)
57 cost_kWh_gen = cost_total_E/energy_B*100
                           // Cost per kWh generated(

```

```

        Paise)
58
59 // Results
60 disp("PART I - EXAMPLE : 7.24 : SOLUTION :-")
61 printf("\nCase(a): Rating of each steam turbine = %.
          f metric hp", rating_turbine)
62 printf("\nCase(b): Energy produced per annum = %.3e
          kWh", energy_B)
63 printf("\nCase(c): Average steam consumption per kWh
          = %.1f kg/kWh", steam_consumption_C)
64 printf("\nCase(d): Evaporation capacity of boiler =
          %.f kg/hr", evaporation_cap)
65 printf("\nCase(e): Total fixed cost = Rs %.2e ,
          fixed_cost_total)
66 printf("\n          Total variable cost = Rs %.2e ,
          variable_cost_total)
67 printf("\n          Cost per kWh generated = %.2f
          paise\n", cost_kWh_gen)
68 printf("\nNOTE: Changes in obtained answer from that
          of textbook answer is due to more precision here
          ' ')

```

---

### Scilab code Exa 7.25 Plot of hydrograph and Average discharge available

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION
7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
          GENERATION
8

```

Fig E7.4 . Plot of Hydrograph

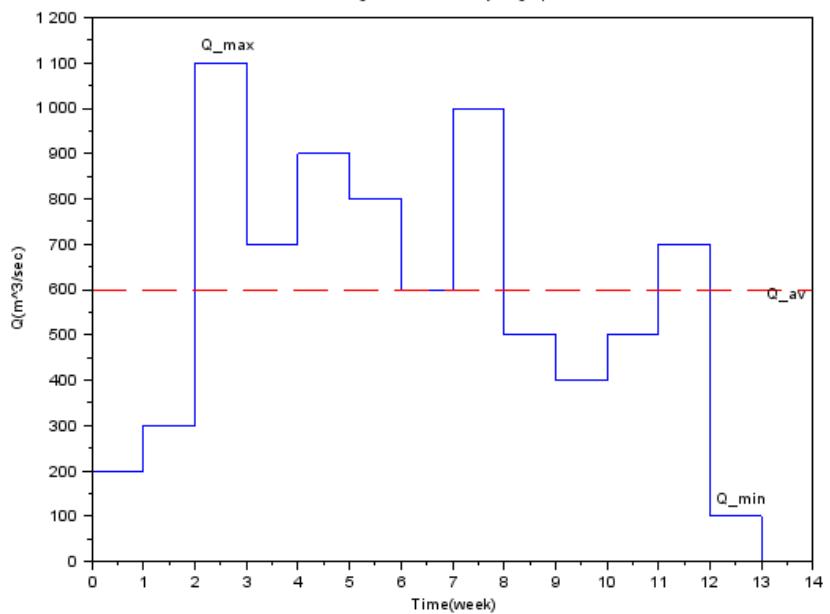


Figure 7.2: Plot of hydrograph and Average discharge available

```

9 // EXAMPLE : 7.25 :
10 // Page number 85
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 w1 = 1.0          // Week 1
15 Q1 = 200.0        // Discharge during week 1(m^2/sec)
16 w2 = 2.0          // Week 2
17 Q2 = 300.0        // Discharge during week 2(m^2/sec)
18 w3 = 3.0          // Week 3
19 Q3 = 1100.0       // Discharge during week 3(m^2/sec)
20 w4 = 4.0          // Week 4
21 Q4 = 700.0        // Discharge during week 4(m^2/sec)
22 w5 = 5.0          // Week 5
23 Q5 = 900.0        // Discharge during week 5(m^2/sec)
24 w6 = 6.0          // Week 6
25 Q6 = 800.0        // Discharge during week 6(m^2/sec)
26 w7 = 7.0          // Week 7
27 Q7 = 600.0        // Discharge during week 7(m^2/sec)
28 w8 = 8.0          // Week 8
29 Q8 = 1000.0       // Discharge during week 8(m^2/sec)
30 w9 = 9.0          // Week 9
31 Q9 = 500.0        // Discharge during week 9(m^2/sec)
32 w10 = 10.0         // Week 10
33 Q10 = 400.0       // Discharge during week 10(m^2/sec)
34 w11 = 11.0         // Week 11
35 Q11 = 500.0       // Discharge during week 11(m^2/sec)
36 w12 = 12.0         // Week 12
37 Q12 = 700.0       // Discharge during week 12(m^2/sec)
38 w13 = 13.0         // Week 13
39 Q13 = 100.0        // Discharge during week 13(m^2/sec)
40 no_week = 13.0      // Total weeks of discharge
41
42 // Calculations
43 Q_average = (Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10+Q11+Q12+
    Q13)/no_week      // Average weekly discharge(m
    ^3/sec)

```

```

44 // Hydrograph
45 W = [0,w1,w1,w2,w2,w3,w3,w4,w4,w5,w5,w6,w6,w7,w7,w8,
      w8,w9,w9,w10,w10,w11,w11,w12,w12,w13,w13,w13]
46 Q = [200,Q1,Q2,Q2,Q3,Q3,Q4,Q4,Q5,Q5,Q6,Q6,Q7,Q7,Q8,
      Q8,Q9,Q9,Q10,Q10,Q11,Q11,Q12,Q12,Q13,Q13,Q13,0]
47 a = gca()
48 a.thickness = 2

          // sets thickness of plot
49 plot(W,Q)

          // Plotting hydrograph
50 q = Q_average
51 w = [0,w1,w2,w3,w4,w5,w6,w7,w8,w9,w10,w11,w12,w13
      ,14]
52 q_dash = [q,q,q,q,q,q,q,q,q,q,q,q,q,q,q,q]
          // Plotting average
          weekly discharge
53 plot(w,q_dash,'r—')
54 a.xlabel.text = 'Time( week )'
          // labels
          x-axis
55 a.ylabel.text = 'Q(m^3/sec)'
          // labels
          y-axis
56 xtitle("Fig E7.4 . Plot of Hydrograph")
57 xset('thickness',2)

          // sets thickness of axes
58 xstring(13,560,'Q_av')
59 xstring(12.02,110,'Q_min')
60 xstring(2.02,1110,'Q_max')
61
62 // Results
63 disp("PART I - EXAMPLE : 7.25 : SOLUTION :-")
64 printf("\nThe hydrograph is shown in the Figure E7.4
      ")
65 printf("\nAverage discharge available for the whole

```

Fig E7.5 . Plot of Flow-duration curve

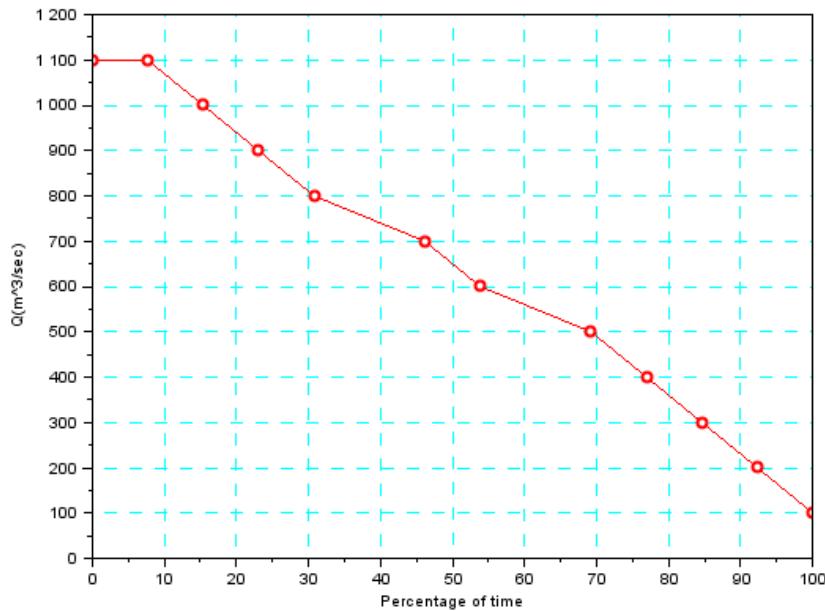


Figure 7.3: Plot of flow duration curve Maximum power Average power developed and Capacity of proposed station

---

period = %.f  $m^3/sec$  , Q\_average)

Scilab code Exa 7.26 Plot of flow duration curve Maximum power Average power devel

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART I : GENERATION

```

```

7 // CHAPTER 7: TARIFFS AND ECONOMIC ASPECTS IN POWER
   GENERATION
8
9 // EXAMPLE : 7.26 :
10 // Page number 85–86
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 Q1 = 1100.0           // Discharge in descending
   order(m^3/sec)
15 Q2 = 1000.0           // Discharge(m^3/sec)
16 Q3 = 900.0             // Discharge(m^3/sec)
17 Q4 = 800.0             // Discharge(m^3/sec)
18 Q5 = 700.0             // Discharge(m^3/sec)
19 Q6 = 600.0             // Discharge(m^3/sec)
20 Q7 = 500.0             // Discharge(m^3/sec)
21 Q8 = 400.0             // Discharge(m^3/sec)
22 Q9 = 300.0             // Discharge(m^3/sec)
23 Q10 = 200.0            // Discharge(m^3/sec)
24 Q11 = 100.0            // Discharge(m^3/sec)
25 no_week = 13.0          // Total weeks of discharge
26 h = 200.0              // Head of installation(m)
27 n_overall = 0.88        // Overall efficiency of
   turbine and generator
28 w = 1000.0             // Density of water(kg/m^3)
29
30 // Calculations
31 n1 = 1.0                // Number of weeks
   for 1100 discharge(m^3/sec)
32 n2 = 2.0                // Number of weeks
   for 1000 and above discharge(m^3/sec)
33 n3 = 3.0                // Number of weeks
   for 900 and above discharge(m^3/sec)
34 n4 = 4.0                // Number of weeks
   for 800 and above discharge(m^3/sec)
35 n5 = 6.0                // Number of weeks
   for 700 and above discharge(m^3/sec)

```

```

36 n6 = 7.0           // Number of weeks
      for 600 and above discharge(m^3/sec)
37 n7 = 9.0           // Number of weeks
      for 500 and above discharge(m^3/sec)
38 n8 = 10.0          // Number of weeks
      for 400 and above discharge(m^3/sec)
39 n9 = 11.0          // Number of weeks
      for 300 and above discharge(m^3/sec)
40 n10 = 12.0         // Number of weeks
      for 200 and above discharge(m^3/sec)
41 n11 = 13.0         // Number of weeks
      for 100 and above discharge(m^3/sec)
42 P1 = n1/no_week*100 // Percentage of
      total period for n1
43 P2 = n2/no_week*100 // Percentage of
      total period for n2
44 P3 = n3/no_week*100 // Percentage of
      total period for n3
45 P4 = n4/no_week*100 // Percentage of
      total period for n4
46 P5 = n5/no_week*100 // Percentage of
      total period for n5
47 P6 = n6/no_week*100 // Percentage of
      total period for n6
48 P7 = n7/no_week*100 // Percentage of
      total period for n7
49 P8 = n8/no_week*100 // Percentage of
      total period for n8
50 P9 = n9/no_week*100 // Percentage of
      total period for n9
51 P10 = n10/no_week*100 // Percentage of
      total period for n10
52 P11 = n11/no_week*100 // Percentage of
      total period for n11
53 P = [0,P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,P11]
54 Q = [Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,Q11]
      // Plotting flow
      duration curve

```

```

55 a = gca() ;
56 a.thickness = 2
    // sets thickness of plot
57 plot(P,Q,'ro-')
58 a.xlabel.text = 'Percentage of time'
                                // labels x-axis
59 a.ylabel.text = 'Q(m^3/sec)'
                                //
    labels y-axis
60 xtitle("Fig E7.5 . Plot of Flow-duration curve")
61 xset('thickness',2)

    // sets thickness of axes
62 xgrid(4)
63 Q_1 = 1.0
    // Discharge
    (m^3/sec)
64 P_1 = 0.736/75*w*Q_1*h*n_overall
    // Power
    developed for Q_1(kW)
65 Q_av = 600.0
    // Average
    discharge(m^3/sec). Obtained from Example 1.7.25
66 P_av = P_1*Q_av/1000.0
    // Average
    power developed (MW)
67 Q_max = Q1
    // Maximum
    discharge(m^3/sec)
68 P_max = P_1*Q_max/1000.0
    // Maximum
    power developed (MW)
69 Q_10 = 1070.0
    // Discharge
    for 10% of time(m^3/sec). Value is obtained from
    graph
70 P_10 = P_1*Q_10/1000.0
    // Installed
    capacity (MW)

71
72 // Results
73 disp("PART I - EXAMPLE : 7.26 : SOLUTION :-")
74 printf("\nFlow-duration curve is shown in the Figure
    E7.5")
75 printf("\nMaximum power developed = %.f MW", P_max)
76 printf("\nAverage power developed = %.f MW", P_av)

```

```
77 printf("\nCapacity of proposed station = %.f MW \n",  
    P_10)  
78 printf("\nNOTE: Changes in the obtained answer from  
    that of textbook is due to more precision here &  
    approximation in textbook solution")
```

---

# Chapter 9

## CONSTANTS OF OVERHEAD TRANSMISSION LINES

Scilab code Exa 9.1 Loop inductance and Reactance of transmission line

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.1 :
10 // Page number 100
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 D = 100.0           // Distance between conductors (
    cm)
```

```

15 d = 1.25 // Diameter of conductor (cm)
16 f = 50.0 // Frequency (Hz)
17
18 // Calculations
19 r_GMR = 0.7788*d/2.0 // GMR of
    conductor (cm)
20 L = 4.0*10**-4*log(D/r_GMR) // Loop
    inductance (H/km)
21 X_L = 2*pi*f*L // Reactance of
    transmission line (ohm)
22
23 // Results
24 disp("PART II - EXAMPLE : 2.1 : SOLUTION :-")
25 printf("\nLoop inductance of transmission line , L =
    %.2e H/km" , L)
26 printf("\nReactance of transmission line , X_L = %.2f
    ohm" , X_L)

```

---

### Scilab code Exa 9.2 Inductance per phase of the system

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.2 :
10 // Page number 101
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data

```

```

14 l = 100.0           // Length of 3-phase
    transmission line(km)
15 D = 120.0           // Distance between conductors(
    cm)
16 d = 0.5             // Diameter of conductor(cm)
17
18 // Calculations
19 r_GMR = 0.7788*d/2.0          // GMR of
    conductor(cm)
20 L = 2.0*10**-4*log(D/r_GMR)      // Inductance
    per phase(H/km)
21 L_1 = L*l                  // Inductance
    per phase for 100km length(H)
22
23 // Results
24 disp("PART II – EXAMPLE : 2.2 : SOLUTION :-")
25 printf("\nInductance per phase of the system , L = %
    .4f H \n", L_1)
26 printf("\nNOTE: ERROR: In textbook to calculate L,
    log10 is used instead of ln i.e natural logarithm
    . So , there is change in answer")

```

---

### Scilab code Exa 9.3 Loop inductance of line per km

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.3 :
10 // Page number 101

```

```

11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 D = 135.0           // Spacing between conductors (cm
    )
15 r = 0.8             // Radius of conductor (cm)
16
17 // Calculations
18 L = (1+4*log(D/r))*10**-7*1000.0          // Loop
    inductance per km(H)
19 L_mH = L*1000.0          // Loop
    inductance per km(mH)
20
21 // Results
22 disp("PART II - EXAMPLE : 2.3 : SOLUTION :-")
23 printf("\nLoop inductance of line per km, L = %.2f
    mH", L_mH)

```

---

#### Scilab code Exa 9.4 Inductance per phase of the system

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.4 :
10 // Page number 101
11 clear ; clc ; close ; // Clear the work space and
    console
12

```

```

13 // Given data
14 l = 80.0           // Length of 3-phase
    transmission line(km)
15 D = 100.0          // Distance between conductors(
    cm)
16 d = 1.0            // Diameter of conductor(cm)
17
18 // Calculations
19 r_GMR = 0.7788*d/2.0           // GMR of
    conductor(cm)
20 L = 2.0*10**-7*log(D/r_GMR)    // Inductance
    per phase(H/m)
21 L_1 = L*l*1000.0                // Inductance
    per phase for 80km(H)
22
23 // Results
24 disp("PART II - EXAMPLE : 2.4 : SOLUTION :-")
25 printf("\nInductance per phase of the system , L = %
    .4f H \n", L_1)
26 printf("\nNOTE: ERROR: Calculation mistake in
    textbook to find Inductance per phase of the
    system")

```

---

### Scilab code Exa 9.5 Total inductance of the line

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L.Soni , P.V.Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.5 :

```

```

10 // Page number 103–104
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 D_a_b = 120.0           // Distance between
   conductors a & b(cm)
15 D_a_bb = 140.0          // Distance between
   conductors a & b'(cm)
16 D_aa_b = 100.0          // Distance between
   conductors a' & b(cm)
17 D_aa_bb = 120.0          // Distance between
   conductors a' & b'(cm)
18 D_a_aa = 20.0           // Distance between
   conductors a & a'(cm)
19 d = 2.0                 // Diameter of conductor(cm
   )
20
21 // Calculations
22 D_m = (D_a_b*D_a_bb*D_aa_b*D_aa_bb)**(1.0/4)
   // Mutual GMD(cm)
23 D_a_a = 0.7788*d/2.0
   // Self GMD of
   conductor a(cm)
24 D_aa_aa = D_a_a
   // Self GMD
   of conductor a'(cm)
25 D_aa_a = D_a_aa
   // Distance
   between conductors a' & a(cm)
26 D_s = (D_a_a*D_a_aa*D_aa_aa*D_aa_a)**(1.0/4)
   // Self GMD(cm)
27 L = 4*10**-4*log(D_m/D_s)
   // Total inductance
   of the line(H/km)
28 L_mH = L*1000.0
   // Total
   inductance of the line(mH/km)

```

```

29
30 // Results
31 disp("PART II - EXAMPLE : 2.5 : SOLUTION :-")
32 printf("\nTotal inductance of the line , L = %.2f mH/
km" , L_mH)

```

---

### Scilab code Exa 9.6 Inductance of the line

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
8 // LINES
9 // EXAMPLE : 2.6 :
10 // Page number 104
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 D_a_b = 175.0           // Distance between
    conductors a & b(cm)
15 D_a_aa = 90.0           // Distance between
    conductors a & a'(cm)
16 d = 2.5                 // Diameter of conductor(cm
    )
17
18 // Calculations
19 GMR = 0.7788*d/2.0      // GMR(cm)
20 D_a_a = GMR             // Self

```

```

        GMD of conductor a(cm)
21 D_aa_aa = D_a_a
                                // Self GMD
        of conductor a'(cm)
22 D_aa_a = 90.0
                                //
        Distance between conductors a' & a(cm)
23 D_s = (D_a_a*D_a_aa*D_aa_aa*D_aa_a)**(1.0/4)
                                // Self GMD of conductor A = Self GMD of
        conductor B(cm)
24 D_a_bb = (D_a_aa**2+D_a_b**2)**(1.0/2)
                                // Distance between conductors a &
        b'(cm)
25 D_m = ((D_a_b*D_a_bb)**2)**(1.0/4)
                                // Mutual GMD(cm)
26 L = 4*10**-4*log(D_m/D_s)
                                // Inductance of the
        line(H/km)
27
28 // Results
29 disp("PART II - EXAMPLE : 2.6 : SOLUTION :-")
30 printf("\nInductance of the line , L = %.1e H/km" , L)

```

---

### Scilab code Exa 9.7 Inductance per km of the double circuit line

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.7 :

```

```

10 // Page number 104
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 D_a_a = 100.0           // Distance between
   conductors a & a(cm)
15 D_a_b = 25.0            // Distance between
   conductors a & b(cm)
16 d = 2.0                 // Diameter of conductor(cm
   )
17
18 // Calculations
19 r = d/2.0
                           //
   Conductor radius(cm)
20 GMR = 0.7788*r
                           // GMR(cm)
21 D_aa_aa = GMR
                           // GMR
   of conductors a & a'(cm)
22 D_aa_a = D_aa_aa
                           // GMR of
   conductors a' & a(cm)
23 D_aa_aa = D_a_a
                           // GMR of
   conductors a' & a'(cm)
24 D_s = (D_a_a*D_a_aa*D_aa_aa*D_aa_a)**(1.0/4)
                           // Self GMD of conductor A = Self GMD of
   conductor B(cm)
25 D_a_bb = (D_a_a**2+D_a_b**2)**(1.0/2)
                           // Distance between conductors a
   & b'(cm)
26 D_aa_b = D_a_bb
                           // Distance
   between conductors a' & b(cm)
27 D_aa_bb = D_a_b
                           // Distance

```

```

        between conductors a' & b'(cm)
28 D_m = (D_a_b*D_a_bb*D_aa_b*D_aa_bb)**(1.0/4)
           // Mutual GMD(cm)
29 L = 2*10**-7*log(D_m/D_s)
           // Inductance/
           conductor/mt(H)
30 L_mH = 2.0*L*1000.0*1000.0
           // Loop inductance per
           km(mH)
31
32 // Results
33 disp("PART II - EXAMPLE : 2.7 : SOLUTION :-")
34 printf("\nInductance per km of the double circuit
line , L = %.1f mH" , L_mH)

```

---

### Scilab code Exa 9.8 Geometric mean radius of the conductor and Ratio of GMR to ove

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.8 :
10 // Page number 104–105
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 n = 7.0          // Number of strands
15 r = 1.0          // Radius of each conductor. Assume
    it 1 for calculation purpose

```

```

16
17 // Calculations
18 D_1_2 = 2.0*r // Distance between conductor 1 & 2
19 D_1_6 = 2.0*r // Distance between conductor 1 & 6
20 D_1_7 = 2.0*r // Distance between conductor 1 & 7
21 D_3_4 = 2.0*r // Distance between conductor 3 & 4
22 D_1_4 = 4.0*r // Distance between conductor 1 & 4
23 D_1_3 = (D_1_4**2-D_3_4**2)**(1.0/2) // Distance between conductor 1 & 3
24 D_1_5 = D_1_3 // Distance between conductor 1 & 5
25 GMR = 0.7788*r // GMR
26 n_o = n-1 // Number of outside strands
27 D_s = (GMR**n*(D_1_2**2*D_1_3**2*D_1_4*D_1_7)**6*(2*r)**n_o)**(1.0/49) // GMR
28 overall_radius = 3*r // Overall conductor radius
29 ratio = D_s/overall_radius // Ratio of GMR to overall conductor radius
30
31 // Results
32 disp("PART II - EXAMPLE : 2.8 : SOLUTION :-")
33 printf("\nGeometric mean radius of the conductor ,\nD_s = %.3 f *r" , D_s)
34 printf("\nRatio of GMR to overall conductor radius =\n%.4 f " , ratio)

```

---

**Scilab code Exa 9.9 Inductance of the line per phase**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.9 :
10 // Page number 108–109
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 d = 1.8                      // Diameter of conductor(cm
    )
15 D_A_B = 4.0                  // Distance between
    conductor A & B(cm)
16 D_B_C = 9.0                  // Distance between
    conductor B & C(cm)
17 D_A_C = 6.0                  // Distance between
    conductor A & C(cm)
18
19 // Calculations
20 D_eq = (D_A_B*D_B_C*D_A_C)**(1.0/3)      //
    Equivalent distance(cm)
21 r_GMR = 0.7788*d/2.0                // GMR(cm)
22 L = 2*10**-4*log(D_eq/r_GMR)        //
    Inductance per phase(H/km)
23 L_mH = L*1000.0                   //
    Inductance per phase(mH/km)
24
25 // Results
26 disp("PART II – EXAMPLE : 2.9 : SOLUTION :-")
27 printf("\nInductance of the line per phase , L = %.3f
    mH/km \n" , L_mH)
28 printf("\nNOTE: ERROR: Calculation mistake in the

```

textbook”)

---

### Scilab code Exa 9.10 Inductance per km of 3 phase transmission line

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.10 :
10 // Page number 109
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 d = 5.0                      // Diameter of conductor(cm
    )
15 d_1 = 400.0                   // Distance between
    conductor 1 & 2(cm)
16 d_2 = 500.0                   // Distance between
    conductor 2 & 3(cm)
17 d_3 = 600.0                   // Distance between
    conductor 1 & 3(cm)
18
19 // Calculations
20 D_eq = (d_1*d_2*d_3)**(1.0/3) // Equivalent distance(cm)
21 r_GMR = 0.7788*d/2.0          // GMR(cm)
22 L = 0.2*log(D_eq/r_GMR)      // Inductance per phase per km(mH)
```

```

23
24 // Results
25 disp("PART II - EXAMPLE : 2.10 : SOLUTION :-")
26 printf("\nInductance per km of 3 phase transmission
line , L = %.3f mH \n", L)
27 printf("\nNOTE: ERROR: Calculation mistake in the
textbook")

```

---

### Scilab code Exa 9.11 Inductance of each conductor per phase per km

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
// LINES
8
9 // EXAMPLE : 2.11 :
10 // Page number 109
11 clear ; clc ; close ; // Clear the work space and
// console
12
13 // Given data
14 d = 3.0 // Diameter of conductor (
// cm)
15 D_12 = 200.0 // Distance between
// conductor 1 & 2(cm)
16 D_23 = 200.0 // Distance between
// conductor 2 & 3(cm)
17 D_31 = 400.0 // Distance between
// conductor 1 & 3(cm)
18
19 // Calculations

```

```

20 D_eq = (D_12*D_23*D_31)**(1.0/3) //  

    Equivalent distance(cm)  

21 r = d/2.0 //  

    Radius of conductor(cm)  

22 L = (0.5+2*log(D_eq/r))*10**-7 //  

    Inductance/phase/m(H)  

23 L_mH = L*1000.0*1000.0 //  

    Inductance per phase per km(mH)  

24  

25 // Results  

26 disp("PART II - EXAMPLE : 2.11 : SOLUTION :-")  

27 printf("\nInductance of each conductor per phase per  

    km, L = %.3f mH \n", L_mH)  

28 printf("\nNOTE: ERROR: Calculation mistake in the  

    textbook")

```

---

**Scilab code Exa 9.12** Inductance of each conductor and Average inductance of each p

```

1 // A Texbook on POWER SYSTEM ENGINEERING  

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  

3 // DHANPAT RAI & Co.  

4 // SECOND EDITION  

5  

6 // PART II : TRANSMISSION AND DISTRIBUTION  

7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION  

    LINES  

8  

9 // EXAMPLE : 2.12 :  

10 // Page number 109-110  

11 clear ; clc ; close ; // Clear the work space and  

    console  

12  

13 // Given data  

14 d = 2.0 // Diameter of conductor(  

    cm)

```

```

15 D_ab = 400.0 // Distance between
    conductor a & b(cm)
16 D_bc = 400.0 // Distance between
    conductor b & c(cm)
17 D_ca = 800.0 // Distance between
    conductor c & a(cm)
18
19 // Calculations
20 I_ab = 1.0*exp(%i*-240.0*%pi/180)

    // I_a/I_b
21 I_cb = 1.0*exp(%i*-120.0*%pi/180)

    // I_c/I_b
22 r_GMR = 0.7788*d/2.0

    // GMR(cm)
23 L_a = 2.0*10**-7*complex(log((D_ab*D_ca)**0.5/r_GMR)
    ,(3**0.5/2*log(D_ab/D_ca))) // Inductance per
    phase of A(H/m)
24 L_amH = L_a*10.0**6

    // Inductance per phase of A(mH/km)
25 L_b = 2.0*10**-7*complex(log((D_bc*D_ab)**0.5/r_GMR)
    ,(3**0.5/2*log(D_bc/D_ab))) // Inductance per
    phase of B(H/m)
26 L_bmH = L_b*10.0**6

    // Inductance per phase of B(mH/km)
27 L_c = 2.0*10**-7*complex(log((D_ca*D_bc)**0.5/r_GMR)
    ,(3**0.5/2*log(D_ca/D_bc))) // Inductance per
    phase of C(H/m)
28 L_cmH = L_c*10.0**6

    // Inductance per phase of C(mH/km)
29 D_eq = (D_ab*D_bc*D_ca)**(1.0/3)

    // Equivalent distance (cm)

```

```

30 L_avg = 0.2*log(D_eq/r_GMR)
           // Average inductance per phase (mH/km)
31
32 // Results
33 disp("PART II - EXAMPLE : 2.12 : SOLUTION :-")
34 printf("\nInductance of conductor a, L_a = (%.4f%.2
         f) mH/km", real(L_amH), imag(L_amH))
35 printf("\nInductance of conductor b, L_b = %.3f mH/
         km", abs(L_bmH))
36 printf("\nInductance of conductor c, L_c = (%.4f%.2
         f) mH/km", real(L_cmH), imag(L_cmH))
37 printf("\nAverage inductance of each phase, L_avg = 
         %.3f mH/km", L_avg)

```

---

### Scilab code Exa 9.13 Inductance per phase

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.13 :
10 // Page number 110
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 D_a_a = 0.9                      // Self GMD of
    conductor a(cm)
15 D_a_aa = 40.0                     // Distance between

```

```

        conductor a & a'(cm)
16 D_a_b = 1000.0           // Distance between
        conductor a & b(cm)
17 D_a_bb = 1040.0          // Distance between
        conductor a & b'(cm)
18 D_aa_b = 960.0           // Distance between
        conductor a' & b(cm)
19 D_c_a = 2000.0           // Distance between
        conductor a & c(cm)
20 D_c_aa = 1960.0          // Distance between
        conductor a' & c(cm)
21 D_cc_a = 2040.0          // Distance between
        conductor a & c'(cm)
22
23 // Calculations
24 D_aa_aa = D_a_a           //
        Self GMD of conductor a'(cm)
25 D_aa_a = D_a_aa           //
        Distance between conductor a' & a(cm)
26 D_s1 = (D_a_a*D_a_aa*D_aa_aa*D_aa_a)**(1.0/4)   //
        Self GMD in position 1(cm)
27 D_s2 = D_s1               //
        Self GMD in position 2(cm)
28 D_s3 = D_s1               //
        Self GMD in position 3(cm)
29 D_s = (D_s1*D_s2*D_s3)**(1.0/3)                   //
        Equivalent self GMD(cm)
30 D_aa_bb = D_a_b           //
        Distance between conductor a' & b'(cm)
31 D_AB = (D_a_b*D_a_bb*D_aa_b*D_aa_bb)**(1.0/4)   //
        Mutual GMD(cm)
32 D_BC = D_AB               //
        Mutual GMD(cm)
33 D_cc_aa = D_c_a           //
        Distance between conductor a' & c'(cm)
34 D_CA = (D_c_a*D_c_aa*D_cc_a*D_cc_aa)**(1.0/4)   //
        Mutual GMD(cm)
35 D_m = (D_AB*D_BC*D_CA)**(1.0/3)                   //

```

```

        Equivalent Mutual GMD(cm)
36 L = 0.2*log(D_m/D_s) //  

        Inductance per phase(mH/km)
37  

38 // Results
39 disp("PART II - EXAMPLE : 2.13 : SOLUTION :-")
40 printf("\nInductance per phase , L = %.3f mH/km" , L)

```

---

### Scilab code Exa 9.14 Inductance per phase of double circuit

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.14 :
10 // Page number 110–111
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 r = 6.0/1000 // Radius of conductor(m
    )
15 D_a_cc = 5.0 // Distance between
    conductor a & c'(m)
16 D_b_bb = 6.0 // Distance between
    conductor b & b'(m)
17 D_c_aa = 5.0 // Distance between
    conductor c & a'(m)
18 D_acc_bbb = 3.0 // Distance between
    conductor ac' & bb'(m)

```

```

19 D_bbb_caa = 3.0          // Distance between
    conductor bb' & ca'(m)
20 D_a_c = 6.0              // Distance between
    conductor a & c(m)
21
22 // Calculations
23 r_GMR = 0.7788*r

    // GMR of conductor(m)
24 D_a_b = (D_acc_bbb**2+((D_b_bb-D_a_cc)/2)**2)
    **(1.0/2)           // Distance between
    conductor a & b(m)
25 D_a_bb = (D_acc_bbb**2+(D_a_cc+(D_b_bb-D_a_cc)/2)
    **2)**(1.0/2)       // Distance between conductor a
    & b'(m)
26 D_a_aa = ((D_acc_bbb+D_bbb_caa)**2+D_c_aa**2)
    **(1.0/2)           // Distance between
    conductor a & a'(m)
27 D_a_a = r_GMR

    // Self GMD of conductor a(m)
28 D_aa_aa = D_a_a

    // Self GMD of conductor a'(m)
29 D_aa_a = D_a_aa

    // Distance between conductor a' & a(m)
30 D_S1 = (D_a_a*D_a_aa*D_aa_aa*D_aa_a)**(1.0/4)
            // Self GMD in position 1(m)
31 D_bb_b = D_b_bb

    // Distance between conductor b' & b(m)
32 D_S2 = (D_a_a*D_b_bb*D_aa_aa*D_bb_b)**(1.0/4)
            // Self GMD in position 2(m)
33 D_S3 = (D_a_a*D_a_aa*D_aa_aa*D_aa_a)**(1.0/4)
            // Self GMD in position 3(m)
34 D_S = (D_S1*D_S2*D_S3)**(1.0/3)
            // Equivalent

```

```

        self GMD(m)
35 D_aa_bb = D_a_b

        // Distance between conductor a' & b'(m)
36 D_aa_b = D_a_bb

        // Distance between conductor a' & b(m)
37 D_AB = (D_a_b*D_a_bb*D_aa_b*D_aa_bb)**(1.0/4)
            // Mutual GMD(m)
38 D_BC = D_AB

        // Mutual GMD(m)
39 D_c_a = D_a_c

        // Distance between conductor c & a(m)
40 D_cc_aa = D_c_a

        // Distance between conductor a' & c'(m)
41 D_cc_a = D_a_cc

        // Distance between conductor c' & a(m)
42 D_CA = (D_c_a*D_c_aa*D_cc_a*D_cc_aa)**(1.0/4)
            // Mutual GMD(m)
43 D_m = (D_AB*D_BC*D_CA)**(1.0/3)
            // Equivalent
            Mutual GMD(m)
44 L = 0.2*log(D_m/D_S)
            //
            Inductance per phase (mH/km)
45
46 // Results
47 disp("PART II - EXAMPLE : 2.14 : SOLUTION :-")
48 printf("\nInductance per phase , L = %.2f mH/km" , L)

```

---

**Scilab code Exa 9.15 Spacing between adjacent conductor to keep same inductance**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.15 :
10 // Page number 111
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 D_eq = 2.88                                // Equilateral
    spacing of line(m)
15
16 // Calculations
17 D = D_eq/2**1.0/3                           // Distance (m)
18 D_13 = 2.0*D                                 // Distance between
    conductor 1 & 3(m)
19 D_12 = D                                     // Distance between
    conductor 1 & 2(m)
20 D_23 = D                                     // Distance between
    conductor 2 & 3(m)
21
22 // Results
23 disp("PART II – EXAMPLE : 2.15 : SOLUTION :-")
24 printf("\nSpacing between conductor 1 & 2 to keep
    inductance same, D_12 = %.1f m", D_12)
25 printf("\nSpacing between conductor 2 & 3 to keep
    inductance same, D_23 = %.1f m", D_23)
26 printf("\nSpacing between conductor 1 & 3 to keep
    inductance same, D_13 = %.1f m", D_13)

```

---

### Scilab code Exa 9.16 Capacitance of line neglecting and taking presence of ground

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.16 :
10 // Page number 112
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 l = 40.0          // Length of line(km)
15 d = 5.0/1000     // Diameter of wire(m)
16 D = 1.5           // Spacing between conductor(m)
17 h = 7.0           // Height of conductors above ground(m
    )
18
19 // Calculations
20 r = d/2

    // Radius of wire(m)
21 e = 1.0/(36*%pi)*10**-9
                                // Constant
    _0
22 // Neglecting presence of ground
23 C_ab_1 = %pi*e/(log(D/r))
                                //
    Capacitance(F/m)
```

```

24 C_ab_12 = C_ab_1*l*1000.0*10**6 // Capacitance( F )
25 // Taking presence of ground
26 C_ab_2 = %pi*e/log(D/(r*(1+(D/(2*h))**2)**(1.0/2)))
// Capacitance(F/m)
27 C_ab_22 = C_ab_2*l*1000.0*10**6
// Capacitance( F )
28
29 // Results
30 disp("PART II - EXAMPLE : 2.16 : SOLUTION :-")
31 printf("\nCapacitance of line neglecting presence of
ground , C_ab = %.3f F ", C_ab_12)
32 printf("\nCapacitance of line taking presence of
ground , C_ab = %.3f F ", C_ab_22)

```

---

### Scilab code Exa 9.17 Capacitance of conductor

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
LINES
8
9 // EXAMPLE : 2.17 :
10 // Page number 114–115
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 d = 2.0/100 // Diameter of conductor(m)
15 D_AB = 4.0 // Spacing between conductor A & B(m)
16 D_BC = 4.0 // Spacing between conductor B & C(m)

```

```

17 D_CA = 8.0      // Spacing between conductor C & A(m)
18
19 // Calculations
20 r = d/2

        // Radius of conductor(m)
21 D = 4.0

        // Assuming coomon distance(m)
22 e = 1.0/(36*%pi)*10**-9
                //
        Constant _0
23 C_A = 2*%pi*e/(log(D/r)-complex(-0.5,0.866)*log(2))
        *1000.0      // Capacitance of conductor A(F/km)
        )
24 C_Au = C_A*10.0**6
                //
        Capacitance of conductor A( F /km)
25 C_B = 2*%pi*e*log(D/r)*1000.0
                //
        Capacitance of conductor B(F/km)
26 C_Bu = C_B*10.0**6
                //
        Capacitance of conductor B( F /km)
27 C_C = 2*%pi*e/(log(D/r)-complex(-0.5,-0.866)*log(2))
        *1000.0      // Capacitance of conductor C(F/km)
28 C_Cu = C_C*10.0**6
                //
        Capacitance of conductor C( F /km)
29
30 // Results
31 disp("PART II - EXAMPLE : 2.17 : SOLUTION :-")
32 printf("\nCapacitance of conductor A, C_A = (%.5f+%
        .6fj) F /km", real(C_Au), imag(C_Au))
33 printf("\nCapacitance of conductor B, C_B = %.6f F
        /km", C_Bu)
34 printf("\nCapacitance of conductor C, C_C = (%.5f%.6
        f) F /km", real(C_Cu), imag(C_Cu))

```

---

### Scilab code Exa 9.18 New value of capacitance

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.18 :
10 // Page number 115
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 d = 2.0/100      // Diameter of conductor(m)
15 D_AB = 4.0       // Spacing between conductor A & B(m)
16 D_BC = 4.0       // Spacing between conductor B & C(m)
17 D_CA = 8.0       // Spacing between conductor C & A(m)
18
19 // Calculations
20 r = d/2           //
    Radius of conductor(m)
21 e = 1.0/(36*%pi)*10**-9
    //
    Constant _0
22 D_eq = (D_AB*D_BC*D_CA)**(1.0/3)
    //
    Equivalent distance(m)
23 C_n = 2*%pi*e*log(D_eq/r)*1000.0
    //
    Capacitance to neutral(F/km)
24 C_nu = C_n*10.0**6
    //
    Capacitance to neutral( F /km)
25
```

```

26 // Results
27 disp("PART II - EXAMPLE : 2.18 : SOLUTION :-")
28 printf("\nNew value of capacitance , C_n = %.5f F /
    km \n", C_nu)
29 printf("\nNOTE: Changes in the obtained answer from
    that of textbook is due to more approximation in
    the textbook")

```

---

### Scilab code Exa 9.19 Capacitance per phase to neutral of a line

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.19 :
10 // Page number 115
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 d = 2.6          // Outside diameter of conductor(cm)
15 D_RY = 8.0       // Spacing between conductor R & Y(m)
16 D_YB = 8.0       // Spacing between conductor Y & B(m)
17 D_RB = 16.0      // Spacing between conductor R & B(m)
18 h = 13.0         // Height of conductor from ground(m)
19
20 // Calculations
21 r = d/2

// Radius of conductor(m)

```

```

22 e = 1.0/(36*pi)*10**-9

        // Constant _0
23 h_12 = (D_RY**2+(2*h)**2)**(1.0/2)                                // Height
        of conductor 1 & 2(m)
24 h_23 = h_12

        // Height of conductor 2 & 3(m)
25 h_31 = (D_RB**2+(2*h)**2)**(1.0/2)                                // Height
        of conductor 3 & 1(m)
26 h_1 = 2*h

        // Height of transposed conductor 1(m)
27 h_2 = 2*h

        // Height of transposed conductor 2(m)
28 h_3 = 2*h

        // Height of transposed conductor 3(m)
29 D_eq = (D_RY*D_YB*D_RB)**(1.0/3)                                     //
        Equivalent distance(m)
30 h_123 = (h_12*h_23*h_31)**(1.0/3)                                    // Height(
        m)
31 h_1_2_3 = (h_1*h_2*h_3)**(1.0/3)                                     // Height
        (m)
32 C_n = 2*pi*e/(log(D_eq*100/r)-log(h_123/h_1_2_3)) *1000.0           // Capacitance of
        conductor A(F/km)
33
34 // Results
35 disp("PART II - EXAMPLE : 2.19 : SOLUTION :-")
36 printf("\nCapacitance per phase to neutral of a line
        , C_n = %.1e F/km" , C_n)

```

---

### Scilab code Exa 9.20 Phase to neutral capacitance

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.20 :
10 // Page number 117–118
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 d = 2.5          // Diameter of conductor(cm)
15 D = 200.0        // Distance of separation(cm)
16 l = 100.0        // Length of line(km)
17
18 // Calculations
19 r = d/2          // Radius of conductor(cm)
20 e = 1.0/(36*%pi)*10**-9           // Constant _0
21 D_m = (D*(3**0.5)*D*(3**0.5)*D*D)**(1.0/4)      // Mutual GMD(cm)
22 D_s = (2*D*r)**(1.0/2)           // Self GMD(cm)
23 C_n = 2*%pi*e*log(D_m/D_s)*1000.0      // Phase-to-neutral capacitance(F/km)
24 C_nu = C_n*l*10.0**6           // Phase-to-neutral capacitance( F )
```

```

25
26 // Results
27 disp("PART II - EXAMPLE : 2.20 : SOLUTION :-")
28 printf("\nPhase-to-neutral capacitance , C_n = %.2f
F ", C_nu)

```

---

### Scilab code Exa 9.21 Capacitance per phase to neutral

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.21 :
10 // Page number 118
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 d = 2.5/100           // Diameter of conductor (m)
15 D = 5.0               // Distance of separation (m)
16 h = 2.0               // Height of separation (m)
17
18 // Calculations
19 r = d/2                //
    Radius of conductor (m)
20 e = 1.0/(36*pi)*10**-9          //
    Constant _0
21 m = (D**2+h**2)**(1.0/2)          //
    (m)
22 n = (D**2+(h*2)**2)**(1.0/2)          //

```

```

(m)
23 D_ab = (D*m)**(1.0/2) // Distance between conductor a & b(m)
24 D_bc = (D*m)**(1.0/2) // Distance between conductor b & c(m)
25 D_ca = (2*D*h)**(1.0/2) // Distance between conductor c & a(m)
26 D_eq = (D_ab*D_bc*D_ca)**(1.0/3) // Equivalent GMD(m)
27 D_s1 = (r*n)**(1.0/2) // Self GMD in position 1(m)
28 D_s2 = (r*h)**(1.0/2) // Self GMD in position 2(m)
29 D_s3 = (r*n)**(1.0/2) // Self GMD in position 3(m)
30 D_s = (D_s1*D_s2*D_s3)**(1.0/3) // Self GMD(m)
31 C_n = 2*%pi*e/log(D_eq/D_s)*1000.0 // Capacitance per phase to neutral(F/km)
32 C_nu = C_n*10.0**6 // Capacitance per phase to neutral( F /km)
33
34 // Results
35 disp("PART II - EXAMPLE : 2.21 : SOLUTION :-")
36 printf("\\nCapacitance per phase to neutral , C_n = % .2 f F /km" , C_nu)

```

---

### Scilab code Exa 9.22 Capacitive reactance to neutral and Charging current per phas

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION

```

```

7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
   LINES
8
9 // EXAMPLE : 2.22 :
10 // Page number 119
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 d = 2.5/100           // Diameter of conductor (m)
15 V = 132.0*10**3      // Line voltage (V)
16 f = 50.0              // Frequency (Hz)
17 h = 4.0                // Height (m)
18 H = 8.0                // Height of separation (m)
19 D_1_33 = 7.0          // Distance between conductors 1
   & 3'(m)
20 D_1_22 = 9.0          // Distance between conductors 1
   & 2'(m)
21 D_1_11 = 8.0          // Distance between conductors 1
   & 1'(m)
22 D_1 = 1.0              // Distance (m)
23
24 // Calculations
25 r = d/2                //
   Radius of conductor (m)
26 e = 1.0/(36*%pi)*10**-9    //
   Constant _0
27 D_12 = (h**2+D_1**2)**(1.0/2)    //
   Distance between conductors 1 & 2(m)
28 D_122 = (h**2+D_1_11**2)**(1.0/2)    //
   Distance between conductors 1 & 2'(m)
29 D_111 = (D_1_11**2+D_1_33**2)**(1.0/2)    //
   Distance between conductors 1 & 1'(m)
30 D_1_2 = (D_12*D_122)**(1.0/2)    //
   Mutual GMD(m)
31 D_2_3 = (D_12*D_122)**(1.0/2)    //
   Mutual GMD(m)
32 D_3_1 = (D_1_33*D_1_11)**(1.0/2)    //

```

```

        Mutual GMD(m)
33 D_eq = (D_1_2*D_2_3*D_3_1)**(1.0/3) // 
        Equivalent GMD(m)
34 D_s1 = (r*D_111)**(1.0/2) // 
        Self GMD in position 1(m)
35 D_s2 = (r*D_1_22)**(1.0/2) // 
        Self GMD in position 2(m)
36 D_s3 = (r*D_111)**(1.0/2) // 
        Self GMD in position 3(m)
37 D_s = (D_s1*D_s2*D_s3)**(1.0/3) // 
        Self GMD(m)
38 C_n = 2*%pi*e/log(D_eq/D_s) // 
        Capacitance per phase to neutral(F/m)
39 X_cn = 1/(2.0*%pi*f*C_n) // 
        Capacitive reactance to neutral(ohms/m)
40 V_ph = V/(3**0.5) // 
        Phase voltage(V)
41 I_charg = V_ph/X_cn*1000.0 // 
        Charging current per phase(A/km)
42
43 // Results
44 disp("PART II - EXAMPLE : 2.22 : SOLUTION :-")
45 printf("\nCapacitive reactance to neutral , X_cn = %.
.2e ohms/m" , X_cn)
46 printf("\nCharging current per phase , I_charg = %.3f
A/km" , I_charg)

```

---

### Scilab code Exa 9.23 Inductive reactance Capacitance and Capacitive reactance of t

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION

```

```

7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
   LINES
8
9 // EXAMPLE : 2.23 :
10 // Page number 119
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 d = 0.8/100           // Diameter of conductor (m)
15 f = 50.0              // Frequency(Hz)
16 D_a_b = 5.0           // Distance between conductors a
   & b(m)
17 D_b_c = 5.0           // Distance between conductors b
   & c(m)
18 D_c_a = 8.0           // Distance between conductors c
   & a(m)
19 l = 25.0               // Length of line(km)
20
21 // Calculations
22 r = d/2                //
   Radius of conductor (m)
23 e = 8.854*10**-12        //
   Constant _0
24 D_e = (D_a_b*D_b_c*D_c_a)**(1.0/3)        //
   Equivalent GMD(m)
25 L = 2*((1.0/4)+log(D_e/r))*10**-4        //
   Inductance(H/km)
26 X_L = 2*%pi*f*L            //
   Inductive reactance per km(ohms)
27 C = %pi*e/log(D_e/r)        //
   Capacitance(F/m)
28 C_l = C*1000.0*l          //
   Capacitance for entire length(F)
29 C_lu = C_l*10.0**6         //
   Capacitance for entire length( F )
30 X_c = 1/(2.0*%pi*f*C_l)      //
   Capacitive reactance to neutral(ohm)

```

```

31 X_ck = X_c/1000.0 //  

    Capacitive reactance to neutral(kilo-ohm)  

32  

33 // Results  

34 disp("PART II - EXAMPLE : 2.23 : SOLUTION :-")  

35 printf("\nInductive reactance of the line per  

    kilometer per phase , X_L = %.3f ohm" , X_L)  

36 printf("\nCapacitance of the line , C = %.3f F " ,  

    C_lu)  

37 printf("\nCapacitive reactance of the transmission  

    line , X_c = %.1f kilo-ohm\n" , X_ck)  

38 printf("\nNOTE: ERROR: Change in obtained answer  

    from that of textbook due to wrong substitution  

    in finding Capacitance")

```

---

### Scilab code Exa 9.24 Capacitance of the line and Charging current

```

1 // A Texbook on POWER SYSTEM ENGINEERING  

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  

3 // DHANPAT RAI & Co.  

4 // SECOND EDITION  

5  

6 // PART II : TRANSMISSION AND DISTRIBUTION  

7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION  

    LINES  

8  

9 // EXAMPLE : 2.24 :  

10 // Page number 119-120  

11 clear ; clc ; close ; // Clear the work space and  

    console  

12  

13 // Given data  

14 V = 250.0 // Line voltage(V)  

15 f = 50.0 // Frequency(Hz)  

16 D = 1.5 // Distance of separation(m)

```

```

17 d = 1.5/100          // Diameter of conductor (m)
18 l = 50.0              // Length of line (km)
19
20 // Calculations
21 // Case(i)
22 r = d/2                //
23 e = 8.854*10**-12      //
24 C = %pi*e/log(D/r)    //
25 C_l = C*1000.0*l       //
26 C_lu = C_l*10.0**6     //
27 // Case(ii)
28 I_charg = 2.0*%pi*f*C_l*V*1000.0   //
29 // Results
30 disp("PART II - EXAMPLE : 2.24 : SOLUTION :-")
31 printf("\nCase(i) : Capacitance of the line , C = %.3
32 f F ", C_lu)
33 printf("\nCase(ii) : Charging current , I_charg = %.2 f
34 mA", I_charg)

```

---

### Scilab code Exa 9.25 Capacitance of the line

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION

```

## LINES

```
8
9 // EXAMPLE : 2.25 :
10 // Page number 120
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 d_1 = 6.0           // Distance between conductor
   1 & 2(m)
15 d_2 = 6.0           // Distance between conductor
   2 & 3(m)
16 d_3 = 12.0          // Distance between conductor
   3 & 1(m)
17 dia = 1.24/100      // Diameter of conductor(m)
18 l = 100.0           // Length of line(km)
19
20 // Calculations
21 r = dia/2           //
   Radius of conductor(m)
22 e = 8.854*10**-12    //
   Constant _0
23 d = (d_1*d_2*d_3)**(1.0/3) //
   Distance(m)
24 C = 2*%pi*e*log(d/r) //
   Capacitance(F/m)
25 C_l = C*1000.0*l     //
   Capacitance for entire length(F)
26 C_lu = C_l*10.0**6    //
   Capacitance for entire length( F )
27
28 // Results
29 disp("PART II - EXAMPLE : 2.25 : SOLUTION :-")
30 printf("\nCapacitance of the line , C = %.3f F ",  

   C_lu)
```

---

### Scilab code Exa 9.26 Capacitance of each line conductor

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 2: CONSTANTS OF OVERHEAD TRANSMISSION
    LINES
8
9 // EXAMPLE : 2.26 :
10 // Page number 120
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 d = 2.0           // Spacing between conductors
    (m)
15 dia = 1.25/100      // Diameter of conductor(m)
16
17 // Calculations
18 r = dia/2          // Radius of
    conductor(m)
19 e = 8.854*10**-12      // Constant  $\epsilon_0$ 
20 C = 2*%pi*e*log(d/r)  // Capacitance(F
    /m)
21 C_u = C*1000*10.0**6      // Capacitance
    for entire length( F /km)
22
23 // Results
24 disp("PART II - EXAMPLE : 2.26 : SOLUTION :-")
25 printf("\nCapacitance of each line conductor , C = %
    .4 f F /km" , C_u)
```

---

# Chapter 10

## STEADY STATE CHARACTERISTICS AND PERFORMANCE OF TRANSMISSION LINES

Scilab code Exa 10.1 Voltage regulation Sending end power factor and Transmission

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
    PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.1 :
10 // Page number 127–128
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
```

```

14 P = 2.0*10**6           // Power delivered (W)
15 V_r = 33.0*10**3        // Receiving end voltage (V)
16 PF_r = 0.8               // Receiving end lagging
    power factor
17 R = 10.0                 // Total resistance of the
    line (ohm)
18 X = 18.0                 // Total inductive
    resistance of the line (ohm)
19
20 // Calculations
21 // Case(i)
22 I = P/(V_r*PF_r)         // Line current
    (A)
23 sin_phi_r = (1-PF_r**2)**0.5 // Sin _R
24 V_s = V_r+I*R*PF_r+I*X*sin_phi_r // Sending end
    voltage (V)
25 reg = (V_s-V_r)/V_r*100 // Voltage
    regulation (%)
26 // Case(ii)
27 PF_s = (V_r*PF_r+I*R)/V_s // Sending end
    lagging power factor
28 // Case(iii)
29 loss = I**2*R             // Losses (W)
30 P_s = P+loss              // Sending end
    power (W)
31 n = P/P_s*100             // Transmission
    efficiency (%)
32
33 // Results
34 disp("PART II - EXAMPLE : 3.1 : SOLUTION :-")
35 printf("\nCase(i) : Percentage voltage regulation =
    %.3f percent", reg)
36 printf("\nCase(ii) : Sending end power factor = %.2f f
    (lag)", PF_s)
37 printf("\nCase(iii) : Transmission efficiency ,      = %
    .2f percent \n", n)
38 printf("\nNOTE: ERROR: pf is 0.8 and not 0.9 as
    mentioned in the textbook problem statement")

```

---

### Scilab code Exa 10.2 Line current Receiving end voltage and Efficiency of transmission lines

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
     PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.2 :
10 // Page number 128–129
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 l = 10.0                      // Length(km)
15 V_s = 11.0*10**3               // Sending end voltage(V)
16 P = 1000.0*10**3               // Load delivered at
    receiving end(W)
17 PF_r = 0.8                     // Receiving end lagging
    power factor
18 r = 0.5                        // Resistance of each
    conductor(ohm/km)
19 x = 0.56                       // Reactance of each
    conductor(ohm/km)
20
21 // Calculations
22 // Case(a)
23 R = r*l                         // Resistance
    per phase(ohm)
24 X = x*l                         // Reactance per
    phase(ohm)
```

```

25 E_s = V_s/3**0.5 // Phase voltage
      (V)
26 I = P/(3**0.5*V_s*PF_r) // Line current (
      A)
27 // Case(b)
28 sin_phi_r = (1-PF_r**2)**0.5 // Sin _R
29 E_r = E_s-I*R*PF_r-I*X*sin_phi_r // Receiving end
      voltage (V)
30 E_r_ll = 3**0.5*E_r/1000 // Receiving end
      line to line voltage (kV)
31 // Case(c)
32 loss = 3*I**2*R // Loss in the
      transmission line (W)
33 P_s = P+loss // Sending end
      power (W)
34 n = P/P_s*100 // Transmission
      efficiency (%)
35 // Alternate method
36 Z = R**2+X**2
37 P_A = 1.0/3*P // Load
      delivered (W/phase)
38 Q = 1.0*P*sin_phi_r/(3*PF_r) // Reactive load
      delivered (VAR/phase)
39 A = (V_s**2/3.0)-2*(P_A*R+Q*X) // Constant
40 B = (1/9.0)*P**2*Z/PF_r**2 // Constant
41 const = (A**2-4*B)**0.5 // sqrt(A^2-4B)
42 E_r_A = ((A+const)/2)**0.5/1000.0 // Receiving end
      voltage (kV/phase)
43 E_r_A_ll = 3**0.5*E_r_A // Receiving end
      line-line voltage (kV)
44 I_A = P/(3**0.5*E_r_A_ll*1000*PF_r) // Line current (
      A)
45 loss_A = 3*I_A**2*R // Loss in the
      transmission line (W)
46 P_s_A = P+loss_A // Sending end
      power (W)
47 n_A = P/P_s_A*100 // Transmission
      efficiency (%)

```

```

48
49 // Results
50 disp("PART II - EXAMPLE : 3.2 : SOLUTION :-")
51 printf("\nCase(a): Line current , |I| = %.1f A", I)
52 printf("\nCase(b): Receiving end voltage , E_r = %.f
      V (line-to-neutral) = %.2f kV (line-to-line)" ,
      E_r,E_r_11)
53 printf("\nCase(c): Efficiency of transmission = %.2f
      percent \n", n)
54 printf("\nAlternative solution by mixed condition:")
55 printf("\nCase(a): Line current , |I| = %.1f A", I_A)
56 printf("\nCase(b): Receiving end voltage , E_r = %.3f
      kV/phase = %.2f kV (line-line)" , E_r_A,E_r_A_11)
57 printf("\nCase(c): Efficiency of transmission = %.2f
      percent" , n_A)

```

---

### Scilab code Exa 10.3 Sending end voltage

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
     PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.3 :
10 // Page number 129
11 clear ; clc ; close ; // Clear the work space and
     console
12
13 // Given data
14 I = 200.0           // Line current (A)
15 PF_r = 0.8          // Receiving end lagging

```

```

    power factor
16 R = 0.6 // Total resistance of the
             line(ohm)
17 X = 1.0 // Total inductive
             resistance of the line(ohm)
18 n = 0.93 // Efficiency(%)
19
20 // Calculations
21 V_r = 3*I**2*R/((3*I*PF_r/n)-3*I*PF_r) // Receiving end phase voltage(V)
22 sin_phi_r = (1-PF_r**2)**0.5 // Sin _R
23 V_s = V_r+I*R*PF_r+I*X*sin_phi_r // Sending
             end voltage(V)
24 V_s_ll = 3**0.5*V_s // Sending
             end line voltage(V)
25
26 // Results
27 disp("PART II - EXAMPLE : 3.3 : SOLUTION :-")
28 printf("\nSending end voltage , V_s(line-line) = %.2
f V" , V_s_ll)

```

---

#### Scilab code Exa 10.4 Distance over which load is delivered

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
             PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.4 :
10 // Page number 129
11 clear ; clc ; close ; // Clear the work space and

```

```

        console
12
13 // Given data
14 P = 15.0*10**6           // Load delivered at
15 PF_r = 0.85              // Receiving end lagging
16 r = 0.905                // Resistance of each
17 conductor(ohm/km)        // Receiving end voltage(V)
18 V_r = 132.0*10**3         )
19
20 // Calculations
21 loss = loss_per*P         // Losses in line(W)
22 I = P/(3**0.5*V_r*PF_r)  // Line current(A)
23 l = loss/(3*I**2*r)      // Length of line(km)
24
25 // Results
26 disp("PART II - EXAMPLE : 3.4 : SOLUTION :-")
27 printf("\nDistance over which load is delivered , l =
%.2f km", l)

```

---

### Scilab code Exa 10.5 Sending end voltage Voltage regulation Value of capacitors and inductors

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
    PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.5 :

```

```

10 // Page number 130
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 f = 50.0 // Frequency (Hz)
15 l = 20.0 // Length (km)
16 P = 5.0*10**6 // Load delivered at
   receiving end (W)
17 PF_r = 0.8 // Receiving end lagging
   power factor
18 r = 0.02 // Resistance of each
   conductor (ohm/km)
19 L = 0.65*10**-3 // Inductance of each
   conductor (H/km)
20 E_r = 10.0*10**3 // Receiving end voltage (V)
21
22 // Calculations
23 R = r*l // Resistance per phase (ohm)
24 X = 2*%pi*f*L*l // Reactance
   per phase (ohm)
25 // Case(a)
26 I = P/(E_r*PF_r) // Line
   current (A)
27 sin_phi_r = (1-PF_r**2)**0.5 // Sin _R
28 E_s = E_r+I*R*PF_r+I*X*sin_phi_r // 
   Sending end voltage (V)
29 E_s_kV = E_s/1000.0 // 
   Sending end voltage (kV)
30 reg = (E_s-E_r)/E_r*100 // 
   Voltage regulation (%)
31 // Case(b)
32 reg_new = reg/2 // New
   regulation (%)
33 E_s_new = (reg_new/100)*E_r+E_r // New
   value of sending end voltage (V)

```

```

34 tan_phi_r1 = ((E_s_new-E_r)*(E_r/P)-R)/X      //  

            tan _r1  

35 phi_r1 = atan(tan_phi_r1)                      // _r1  

            (radians)  

36 phi_r1d = phi_r1*180/%pi                       // _r1  

            (degree)  

37 PF_r1 = cos(phi_r1)                            //  

            Lagging power factor of receiving end  

38 sin_phi_r1 = (1-PF_r1**2)**0.5                //  

            Sin _r1  

39 I_R_new = P/(E_r*PF_r1)                        // New  

            line current(A)  

40 I_R = I_R_new*complex(PF_r1,-sin_phi_r1)        //  

41 I_c = I_R-I*complex(PF_r,-sin_phi_r)           //  

            Capacitive current(A)  

42 I_C = imag(I_c)                                //  

            Imaginary part of Capacitive current(A)  

43 c = I_C/(2*%pi*f*E_r)*10.0**6                 //  

            Capacitance( F )  

44 // Case(c)  

45 loss_1 = I**2*R                                 // Loss(  

            W)  

46 n_1 = P/(P+loss_1)*100                         //  

            Transmission efficiency (%)  

47 loss_2 = I_R_new**2*R                          // Loss(  

            W)  

48 n_2 = P/(P+loss_2)*100                         //  

            Transmission efficiency (%)  

49  

50 // Results  

51 disp("PART II - EXAMPLE : 3.5 : SOLUTION :-")  

52 printf("\nCase(a): Sending end voltage , E_s = %.2f  

            kV" , E_s_kV)  

53 printf("\n                  Voltage regulation of the line =  

            %.1f percent" , reg)  

54 printf("\nCase(b): Value of capacitors to be placed  

            in parallel with load , c = %.2f F " , c)  

55 printf("\nCase(c): Transmission efficiency in part (a

```

```

) , _1 = %.2f percent" , n_1)
56 printf("\n           Transmission efficiency in part(b
) , _2 = %.1f percent" , n_2)

```

---

### Scilab code Exa 10.6 Voltage regulation Sending end voltage Line loss and Sending

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
     PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.6 :
10 // Page number 130–131
11 clear ; clc ; close ; // Clear the work space and
     console
12
13 // Given data
14 f = 50.0
15 l = 10.0
16 Z_l = 0.5*exp(%i*60.0*%pi/180)
17 P = 316.8*10**3
18 PF_r = 0.8

```

// Frequency (Hz)

// Line length (km)

// Load impedance (ohm/km)

// Load side power (W)

// Load side power factor

```

19 E_r = 3.3*10**3
                                // Load bus
        voltage (V)
20
21 // Calculations
22 Z_line = Z_l*l
                                // Load
        impedance (ohm)
23 I_r = P/(E_r*PF_r)*exp(%i*-acos(PF_r))
                                // Line current (A)
24 sin_phi_r = (1-PF_r**2)**0.5
                                // Sin _R
25 E_s = E_r+I_r*Z_line
                                // Sending end
        voltage (V)
26 reg = (abs(E_s)-abs(E_r))/abs(E_r)*100
                                // Voltage regulation (%)
27 R = real(Z_line)
                                // Resistance
        of the load line (ohm)
28 loss = abs(I_r)**2*R
                                // Loss in the
        transmission line (W)
29 loss_kW = loss/1000.0
                                // Loss in the
        transmission line (kW)
30 P_s = P+loss
                                //
        Sending end power (W)
31 angle_Er_Es = phasemag(E_s)
                                // Angle between V_r and
        V_s( )
32 angle_Er_Ir = acosd(PF_r)
                                // Angle between V_r
        and I_r( )
33 angle_Es_Is = angle_Er_Es+angle_Er_Ir
                                // Angle between V_s and I_s( )
34 PF_s = cosd(angle_Es_Is)

```

```

                                // Sending end power
factor
35
36 // Results
37 disp("PART II - EXAMPLE : 3.6 : SOLUTION :-")
38 printf("\nVoltage regulation = %.2f percent", reg)
39 printf("\nSending end voltage , E_s = %.f % .1 f V"
, abs(E_s),phasemag(E_s))
40 printf("\nLine loss = %.f kW", loss_kW)
41 printf("\nSending end power factor = %.2f ", PF_s)

```

---

### Scilab code Exa 10.7 Nominal pi equivalent circuit parameters and Receiving end vo

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.7 :
10 // Page number 132–133
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 V_s = 66.0 // Voltage(kV)
15 f = 50.0 // Frequency(Hz)
16 l = 150.0 // Line length(km)
17 r = 0.25 // Resistance of each
conductor(ohm/km)
18 x = 0.5 // Inductive reactance
of each conductor(ohm/km)

```

```

19 y = 0.04*10**-4 // Capacitive
    admittance (s/km)
20
21 // Calculations
22 // Case(a)
23 R = r*1 // Total resistance (ohm)
24 X = x*1 // Inductive reactance (ohm)
25 Y = y*1 // Capacitive resistance (s)
26 Y_2 = Y/2 // 1/2 of Capacitive resistance (s)
27 // Case(b)
28 Z = complex(R,X) // Total impedance (ohm)
29 A = 1+(Y*exp(%i*90.0*%pi/180)*Z/2) // Line constant
30 V_R_noload = V_s/abs(A) // Receiving end voltage at no-load (kV)
31
32 // Results
33 disp("PART II - EXAMPLE : 3.7 : SOLUTION :-")
34 printf("\nCase(a): Total resistance , R = %.1f ohm",
        R)
35 printf("\n          Inductive reactance , X = %.1f ohm
        ", X)
36 printf("\n          Capacitive resistance , Y = %.1e s
        ", Y)
37 printf("\n          Capacitive resistance , Y/2 = %.1e
        s", Y_2)
38 printf("\nCase(b): Receiving end voltage at no-load ,
        V_R = %.2f kV", V_R_noload)

```

---

**Scilab code Exa 10.8 Voltage Current and Power factor at sending end**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
    PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.8 :
10 // Page number 133–134
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 f = 50.0                      // Frequency (Hz)
15 V_r = 132.0*10**3             // Line voltage at
    receiving end (V)
16 L = 100.0                      // Line length (km)
17 r = 0.17                        // Resistance (ohm/km/
    phase)
18 l = 1.1*10**-3                // Inductance (H/km/
    phase)
19 c = 0.0082*10**-6              // Capacitance (F/km/
    phase)
20 P_L = 70.0*10**6                // Load at receiving
    end (W)
21 PF_r = 0.8                      // Lagging load power
    factor
22
23 // Calculations
24 E_r = V_r/3**0.5               //
    Receiving end phase voltage (V)
25 I_r = P_L/(3**0.5*V_r*PF_r)*exp(%i*-acos(PF_r))
    // Receiving end current (A)
26 R = r*L                         //

```

```

        Total resistance (ohm/phase)
27 X = 2*pi*f*l*L
                                //
        Inductive reactance (ohm/phase)
28 Z = complex(R,X)
                                // Total
        impedance(ohm/phase)
29 Y = 2*pi*f*c*exp(%i*90.0*pi/180)/L
                                // Shunt admittance of line (mho
        /phase)
30 E = E_r+I_r*(Z/2)
                                // Voltage
        across shunt admittance(V/phase)
31 I_s = I_r+E*Y
                                //
        Sending end current(A)
32 E_s = E+I_s*(Z/2)
                                // Sending
        end voltage(V/phase)
33 E_s_ll = 3**0.5*abs(E_s)/1000
                                // Sending end line to
        line voltage(kV)
34 angle_Er_Es = phasemag(E_s)
                                // Angle between E_r
        and V_s( )
35 angle_Er_Is = phasemag(I_s)
                                // Angle between E_r
        and I_s( )
36 angle_Es_Is = angle_Er_Es-angle_Er_Is
                                // Angle between E_s and I_s( )
37 PF_s = cosd(angle_Es_Is)
                                // Sending end
        power factor
38
39 // Results
40 disp("PART II - EXAMPLE : 3.8 : SOLUTION :-")
41 printf("\nVoltage at sending end, E_s = %.2 f %.2
        f    V/phase = %.f kV (line-to-line)", abs(E_s),

```

```

    phasemag(E_s),E_s_11)
42 printf("\nCurrent at sending end , I_s = %.1f % .1
        f A" , abs(I_s),phasemag(I_s))
43 printf("\nSending end power factor = %.3f (lagging)"
        , PF_s)

```

---

### Scilab code Exa 10.9 Sending end voltage Current and Transmission efficiency

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.9 :
10 // Page number 134
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 f = 50.0 // Frequency (Hz)
15 E_r = 66.0*10**3 // Line voltage at
receiving end (V)
16 l = 120.0 // Line length (km)
17 r = 0.1 // Resistance (ohm/km/
phase)
18 x = 0.3 // Inductive reactance
(ohm/km/phase)
19 y = 0.04*10**-4 // Capacitive
susceptance (S/km/phase)
20 P_L = 10.0*10**6 // Load at receiving
end (W)

```

```

21 PF_r = 0.8 // Lagging load power
               factor
22
23 // Calculations
24 R = r*l

               // Total resistance (ohm/phase)
25 X = x*l

               // Inductive reactance (ohm/phase)
26 Y = y*l

               // Susceptance (mho)
27 Z = complex(R,X)

               // Total impedance (ohm/phase)
28 V_r = E_r/3**0.5

               // Receiving end phase voltage (V)
29 I_r = P_L/(3**0.5*E_r*PF_r)*exp(%i*-acos(PF_r))
               // Load current (A)
30 V_1 = V_r+I_r*(Z/2)
               //
               Voltage across capacitor (V)
31 I_c = %i*Y*V_1

               // Charging current (A)
32 I_s = I_r+I_c

               // Sending end current (A)
33 V_s = V_1+I_s*(Z/2)
               //
               Sending end voltage (V/phase)
34 V_s_ll = 3**0.5*abs(V_s)/1000.0
               // Sending end
               line to line voltage (kV)
35 angle_Vr_Vs = phasemag(V_s)
               // Angle

```

```

        between V_r and V_s( )
36 angle_Vr_Is = phasemag(I_s)                                // Angle
        between V_r and I_s( )
37 angle_Vs_Is = angle_Vr_Vs-angle_Vr_Is                         // Angle between V_s
        and I_s( )
38 PF_s = cosd(angle_Vs_Is)                                     //
        Sending end power factor
39 P_s = 3*abs(V_s*I_s)*PF_s                                    //
        Sending end power(W)
40 n = P_L/P_s*100

        // Transmission efficiency (%)

41
42 // Results
43 disp("PART II – EXAMPLE : 3.9 : SOLUTION :-")
44 printf("\nSending end voltage , | V_s | = %.f V/phase = "
       "%3.f V (line-to-line)", abs(V_s),V_s_11)
45 printf("\nSending end current , | I_s | = %.2f A", abs(
       I_s))
46 printf("\nTransmission efficiency = %.2f percent \n"
       , n)
47 printf("\nNOTE: ERROR: Calculation mistake in
       finding sending end power factor")
48 printf("\n      Changes in the obtained answer from
       that of textbook is due to more precision")

```

---

**Scilab code Exa 10.10 Line to line voltage and Power factor at sending end**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.

```

```

4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
    PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.10 :
10 // Page number 135
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 f = 50.0                                // Frequency(Hz)
15 l = 125.0                                 // Line length(km)
16 P_r = 40.0*10**6                         // Load at receiving
    end(VA)
17 V_r = 110.0*10**3                         // Line voltage at
    receiving end(V)
18 PF_r = 0.8                                // Lagging load power
    factor
19 R = 11.0                                   // Resistance(ohm/
    phase)
20 X = 38.0                                    // Inductive reactance
    (ohm/phase)
21 Y = 3.0*10**-4                            // Capacitive
    susceptance(S)
22
23 // Calculations
24 // Case(i)
25 E_r = V_r/3**0.5                           //
    Receiving end phase voltage(V)
26 Z = complex(R,X)                           // Total
    impedance(ohm/phase)
27 I_c1 = E_r*(Y/2)*exp(%i*90.0*%pi/180)   // Current through shunt
    admittance at receiving end(A)

```

```

28 I_r = P_r/(3**0.5*V_r)*exp(%i*-acos(PF_r))
          // Load current(A)
29 I = I_r+I_c1
          //
          Current through series impedance(A)
30 E_s = I*Z+E_r
          //
          Voltage across shunt admittance at sending end(V)
31 E_s_ll = 3**0.5*E_s/1000.0
          //
          Line to line
          voltage at sending end(kV)
32 I_c2 = E_s*(Y/2)*exp(%i*90.0*pi/180)
          //
          Current through shunt
          admittance at sending end(A)
33 // Case(ii)
34 I_s = I_c2+I_r
          //
          Sending end current(A)
35 angle_Er_Es = phasemag(E_s)
          //
          Angle between E_r
          and E_s( )
36 angle_Er_Is = phasemag(I_s)
          //
          Angle between E_r
          and I_s( )
37 angle_Es_Is = angle_Er_Es-angle_Er_Is
          //
          Angle between E_s and I_s(
          )
38 PF_s = cosd(angle_Es_Is)
          //
          Sending end
          power factor
39
40 // Results
41 disp("PART II - EXAMPLE : 3.10 : SOLUTION :-")
42 printf("\nCase(i) : Line to line voltage at sending
          end , E_s = %.f kV" , abs(E_s_ll))
43 printf("\nCase(ii) : Sending end power factor = %.3f
          \n" , PF_s)
44 printf("\nNOTE: Answers in the textbook are

```

incomplete")

---

### Scilab code Exa 10.11 Voltage Current Power factor at sending end Regulation and T

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.11 :
10 // Page number 135–137
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 f = 50.0 // Frequency (Hz)
15 R = 28.0 // Resistance (ohm/
phasemag)
16 X = 63.0 // Inductive reactance
(ohm/phasemag)
17 Y = 4.0*10**-4 // Capacitive
susceptance (mho)
18 P_r = 75.0*10**6 // Load at receiving
end (VA)
19 PF_r = 0.8 // Lagging load power
factor
20 V_r = 132.0*10**3 // Line voltage at
receiving end (V)
21
22 // Calculations
23 // Case(i) Nominal T method
```

```

24 Z = complex(R,X)
      // Total impedance (ohm/phasemag)
25 E_r = V_r/3**0.5
      // Receiving end phasemag voltage (V)
26 I_r = P_r/(3**0.5*V_r)*exp(%i*-acos(PF_r))
      // Line current at receiving end (A)
27 E = E_r+I_r*(Z/2)
28 I_c = %i*Y*E
      // Capacitive current (A)
29 I_s = I_r+I_c
      // Sending end current (A)
30 v_drop = I_s*(Z/2)
      // Voltage drop (V)
31 E_s = E+I_s*(Z/2)
      // Sending end voltage (V)
32 E_s_kV = E_s/1000.0
      // Sending end voltage (kV)
33 E_s_ll= 3**0.5*abs(E_s)
      // Sending end line voltage (V)
34 E_s_llkV = E_s_ll/1000.0
      // Sending end line voltage (kV)
35 angle_Er_Es = phasemag(E_s)
      // Angle between E_r and E_s( )
36 angle_Er_Is = phasemag(I_s)
      // Angle between E_r and I_s( )
37 angle_Es_Is = angle_Er_Es-angle_Er_Is

```

```

// Angle between E_s
and I_s( )
38 PF_s = cosd(angle_Es_Is)
// Sending end power factor
39 P_s = 3**0.5*E_s_ll*abs(I_s)*PF_s
// Power at
// sending end(W)
40 reg = (abs(E_s_ll)-V_r)/V_r*100
// Regulation(%)
41 n = (P_r*PF_r)/P_s*100
// Transmission efficiency(%)
42 // Case(ii) Nominal method
43 I_c2 = E_r*(%i*Y/2)
// Current through shunt admittance at receiving
// end(A)
44 I = I_r+I_c2
// Line current(A)
45 E_s_p = E_r+I*Z
// Sending end voltage(V)
46 E_s_pkV = E_s_p/1000.0
// Sending end voltage(kV)
47 E_s_pll = 3**0.5*abs(E_s_p)
// Sending end line voltage(V)
48 E_s_pkV = E_s_pll/1000.0
// Sending end line voltage(kV)
49 I_c1 = E_s_p*(%i*Y/2)
// Current through shunt admittance at sending end(A)

```

```

50 I_s_p = I+I_c1
      // Sending end current (A)
51 angle_Er_Esp = phasemag(E_s) // Angle
      between E_r and E_s( )
52 angle_Er_Isp = phasemag(I_s) // Angle
      between E_r and I_s( )
53 angle_Es_Isp = angle_Er_Esp-angle_Er_Isp // Angle between E_s
      and I_s( )
54 PF_s_p = cosd(angle_Es_Isp) // Sending
      end power factor
55 P_s_p = 3**0.5*E_s_pll*abs(I_s_p)*PF_s_p
      // Power at sending end
      (W)
56 reg_p = (abs(E_s_pll)-V_r)/V_r*100 // Regulation (%)
57 n_p = (P_r*PF_r)/P_s_p*100 // Transmission efficiency (%)
58
59 // Results
60 disp("PART II - EXAMPLE : 3.11 : SOLUTION :-")
61 printf("\n(i) Nominal T method")
62 printf("\nCase(a): Voltage at sending end , E_s = %.2
      f %.2 f kV = %.1 f kV (line-to-line)" , abs(
      E_s_kV) , phasemag(E_s_kV) , E_s_llkV)
63 printf("\nCase(b): Sending end current , I_s = %.1
      f %.2 f A" , abs(I_s) , phasemag(I_s))
64 printf("\nCase(c): Power factor at sending end = %.4
      f (lagging)" , PF_s)
65 printf("\nCase(d): Regulation = %.2 f percent" , reg)
66 printf("\nCase(e): Efficiency of transmission = %.2 f
      percent \n" , n)
67 printf("\n(ii) Nominal method")

```

```

68 printf("\nCase(a): Voltage at sending end, E_s = %.2
       f %.2 f kV = %.1f kV (line-to-line)", abs(
       E_s_pkV), phasemag(E_s_pkV), E_s_pkV)
69 printf("\nCase(b): Sending end current, I_s = %.1
       f %.2 f A", abs(I_s_p), phasemag(I_s_p))
70 printf("\nCase(c): Power factor at sending end = %.4
       f (lagging)", PF_s_p)
71 printf("\nCase(d): Regulation = %.2 f percent", reg_p
       )
72 printf("\nCase(e): Efficiency of transmission = %.2 f
       percent \n", n_p)
73 printf("\nNOTE: Changes in the obtained answer from
       that of textbook is due to more precision here
       and more approximation in textbook")

```

---

### Scilab code Exa 10.12 Receiving end Voltage Load and Nature of compensation required

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
    PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.12 :
10 // Page number 143
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 E_s = 275.0           // Sending end voltage (
    kV)
15 f = 50.0              // Frequency (Hz)

```

```

16 l = 400.0 // Line length (km)
17 x = 0.05 // Inductive reactance (
    ohm/km)
18 y = 3.0*10**-6 // Line charging
    susceptance (S/km)
19 r = 0.0 // Lossless line
20
21 // Calculations
22 // Case(a)
23 R = r*l // Total resistance (ohm/
    phase)
24 X = x*l // Inductive reactance (
    ohm/phase)
25 Y = y*l // Susceptance (mho)
26 Z = complex(R,X) // Total impedance (ohm/
    phase)
27 A = 1+(Y*Z/2)*%i // Line constant
28 E_r = E_s/abs(A) // Receiving end voltage
    at no load (kV)
29 // case(b)
30 Z_0 = (X/Y)**0.5 // Load at receiving end
    (ohm)
31 // Case(c)
32 Z_0_new = 1.2*Z_0 // New load at receiving
    station (ohm)
33
34 // Results
35 disp("PART II - EXAMPLE : 3.12 : SOLUTION :-")
36 printf("\nCase(a): Receiving end voltage on open
    circuit = %.1f kV", E_r)
37 printf("\nCase(b): Load at receiving end for flat
    voltage profile on line , Z_0 = %.1f ", Z_0)
38 printf("\nCase(c): Distributed inductive reactance
    of the line is to be increased as , Loading for
    new voltage profile = %.2f ", Z_0_new)

```

---

### Scilab code Exa 10.13 Sending end voltage and Current

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
    PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.13 :
10 // Page number 143–144
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V_r = 220.0*10**3 // Receiving end voltage
    (V)
15 Z = complex(20,100) // Impedance (ohm/phase)
16 Y = %i*0.0010 // Admittance (mho)
17 I_r = 300.0 // Receiving end current
    (A)
18 PF_r = 0.9 // Lagging power factor
19
20 // Calculations
21 V_2 = V_r/3**0.5 // Receiving end phase voltage (V)
22 I_2 = I_r*exp(%i*-acos(PF_r)) // Receiving end current (A)
23 I_C2 = (Y/2)*V_2 // Capacitive current at receiving end (A)
24 I = I_2+I_C2
25 V_1 = V_2+I*Z //
```

```

    Voltage across shunt admittance at sending end(V)
26 V_1kV = V_1/1000.0                                //
    Voltage across shunt admittance at sending end(kV
)
27 I_C1 = (Y/2)*V_1                                  //
    Capacitive current at sending end(A)               //
28 I_1 = I_C1+I_2                                     //
    Sending end current(A)                            //

29
30 // Results
31 disp("PART II - EXAMPLE : 3.13 : SOLUTION :-")
32 printf("\nSending end voltage , V_1 = %.2 f % .2 f
      kV" , abs(V_1kV),phasemag(V_1kV))
33 printf("\nSending end current , I_1 = %.3 f % .4 f   A
      " , abs(I_1),phasemag(I_1))

```

---

**Scilab code Exa 10.14 Incident voltage and Reflected voltage at receiving end and**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
      PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.14 :
10 // Page number 144
11 clear ; clc ; close ; // Clear the work space and
      console
12 funcprot(0)
13
14 // Given data
15 f = 50.0                                         // Frequency (Hz)

```

```

16 r = 0.1 // Resistance (ohm/km)
17 l = 1.4*10**-3 // Inductance (H/km)
18 c = 8.0*10**-9 // Capacitance (F/km)
19 g = 4.0*10**-8 // conductance (mho/km)
20 V_r = 400.0 // Receiving end
    voltage (kV)
21 x = 200.0 // Length of line (km)
22
23 // Calculations
24 V_2 = V_r/3**0.5 // Receiving end phase voltage (kV)
25 z = r+%i*2*%pi*f*l // Total impedance (ohm/km)
26 y = g+%i*2*%pi*f*c // Total susceptance (mho/km)
27 Z_c = (z/y)**0.5 // Surge impedance (ohm)
28 gamma = (z*y)**0.5 // 
29 // Case(i)
30 V_0_plus = V_2/2 // Incident voltage to neutral at receiving end (kV)
31 // Case(ii)
32 V_0_minus = V_2/2 // Reflected voltage to neutral at receiving end (kV)
33 // Case(iii)
34 gamma_l = gamma*x // l
35 V_1_plus = (V_2/2)*exp(gamma_l) // Incident voltage to neutral at 200 km from receiving end (kV)
36 V_1_minus = (V_2/2)*exp(-gamma_l) // Reflected voltage to neutral at 200 km from receiving end (kV)
37 // Case(iv)
38 V_1 = V_1_plus+V_1_minus // Resultant voltage to neutral (kV)
39 V_L = abs(V_1) // Resultant voltage to neutral (kV)
40 V_L_ll = 3**0.5*V_L // Line to

```

```

        line voltage at 200 km from receiving end(kV)
41
42 // Results
43 disp("PART II - EXAMPLE : 3.14 : SOLUTION :-")
44 printf("\nCase(i) : Incident voltage to neutral at
        receiving end, V_0_plus = %.1 f % . f    kV", abs(
        V_0_plus), phasemag(V_0_plus))
45 printf("\nCase(ii) : Reflected voltage to neutral at
        receiving end, V_0_minus = %.1 f % . f    kV", abs(
        V_0_minus), phasemag(V_0_minus))
46 printf("\nCase(iii): Incident voltage to neutral at
        200 km from receiving end, V_1_plus = (%.3f+%.2fj
        ) kV", real(V_1_plus), imag(V_1_plus))
47 printf("\nCase(iv) : Resultant voltage to neutral at
        200 km from receiving end, V_L = %.2f kV", V_L)
48 printf("\n          Line to line voltage at 200 km
        from receiving end = %.2f kV", V_L_ll)

```

---

### Scilab code Exa 10.15 A B C D constants

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
    PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.15 :
10 // Page number 145
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data

```

```

14 f = 50.0 // Frequency (Hz)
15 L = 200.0 // Line length (km)
16 l = 1.20*10**-3 // Inductance (H/km)
17 c = 8.0*10**-9 // Capacitance (F/km)
18 r = 0.15 // Resistance (ohm/km)
19 g = 0.0 // Conductance (mho/km)
20
21 // Calculations
22 z = r+%i*2*%pi*f*l // Total
    impedance(ohm/km)
23 Z = z*L // Total
    impedance(ohm)
24 y = g+%i*2*%pi*f*c // Total
    susceptance(mho/km)
25 Y = y*L // Total
    susceptance(mho/km)
26 gamma_1 = (Z*Y)**0.5 // l
27 alpha_1 = real(gamma_1) // l
28 beta_1 = imag(gamma_1) // l
29 Z_c = (Z/Y)**0.5 // Surge
    impedance(ohm)
30 A = cosh(gamma_1) // Constant
31 B = Z_c*sinh(gamma_1) // Constant(
    ohm)
32 C = (1/Z_c)*sinh(gamma_1) // Constant(
    S)
33 D = A // Constant
34
35 // Results
36 disp("PART II - EXAMPLE : 3.15 : SOLUTION :-")
37 printf("\nA = D = %.3 f % .2 f    ", abs(A),phasemag(A))
38 printf("\nB = %.2 f % .3 f      ", abs(B),phasemag(B))
39 printf("\nC = %.2 e % .3 f    S", abs(C),phasemag(C))

```

---

### Scilab code Exa 10.16 Sending end voltage Current Power factor and Efficiency

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
    PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.16 :
10 // Page number 145–146
11 clear ; clc ; close ; // Clear the work space and
    console
12 funcprot(0)
13
14 // Given data
15 V_r = 132.0*10**3 // Receiving end voltage
    (V)
16 f = 50.0 // Frequency (Hz)
17 L = 200.0 // Line length (km)
18 l = 1.3*10**-3 // Inductance (H/km)
19 c = 9.0*10**-9 // Capacitance (F/km)
20 r = 0.2 // Resistance (ohm/km)
21 g = 0.0 // Conductance (mho/km)
22 P_r = 50.0*10**6 // Power received (VA)
23 PF_r = 0.8 // Lagging power factor
    at receiving end
24
25 // Calculations
26 z = r+%i*2*%pi*f*l // Total impedance (ohm/km)
27 y = g+%i*2*%pi*f*c // Total susceptance (mho/km)
28 Z_c = (z/y)**0.5 // Surge impedance (ohm)
29 gamma = (z*y)**0.5 //
```

```

30 gamma_1 = gamma*L //  

    1  

31 cosh_g1 = cosh(gamma_1) //  

    cosh 1  

32 sinh_g1 = sinh(gamma_1) //  

    sinh 1  

33 V_2 = V_r/(3**0.5) //  

    Receiving end phase voltage(V)  

34 I_2 = P_r/(3*V_2)*exp(%i*-acos(PF_r)) //  

    Line current(A)  

35 V_1 = V_2*cosh_g1+I_2*Z_c*sinh_g1 //  

    Sending end voltage(V)  

36 V_1kV = V_1/1000.0 //  

    Sending end voltage(kV)  

37 I_1 = (V_2/Z_c)*sinh_g1+I_2*cosh_g1 //  

    Sending end current(A)  

38 angle_V2_V1 = phasemag(V_1) //  

    Angle between V_2 and V_1( )  

39 angle_V2_I1 = phasemag(I_1) //  

    Angle between V_2 and I_1( )  

40 angle_V1_I1 = angle_V2_V1-angle_V2_I1 //  

    Angle between V_1 and I_1( )  

41 PF_s = cosd(angle_V1_I1) //  

    Sending end power factor  

42 P_1 = 3*abs(V_1*I_1)*PF_s //  

    Sending end power(W)  

43 P_2 = P_r*PF_r //  

    Receiving end power(W)  

44 n = P_2/P_1*100 //  

    Efficiency  

45  

46 // Results  

47 disp("PART II - EXAMPLE : 3.16 : SOLUTION :-")  

48 printf("\nSending end voltage , V_1 = %.3 f %.4 f  

    kV per phase" , abs(V_1kV) , phasemag(V_1kV))  

49 printf("\nSending end current , I_1 = %.3 f %.2 f A  

    " , abs(I_1) , phasemag(I_1))

```

```
50 printf("\nPower factor = %.3f ", PF_s)
51 printf("\nEfficiency ,      = %.2f percent", n)
```

---

### Scilab code Exa 10.17 Values of auxiliary constants A B C D

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
     PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.17 :
10 // Page number 147–148
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 f = 50.0                      // Frequency (Hz)
15 L = 160.0                      // Line length (km)
16 r = 0.15                       // Resistance (ohm/km/
    phasemag)
17 l = 1.2*10**-3                // Inductance (H/km/
    phasemag)
18 c = 0.008*10**-6              // Capacitance (F/km/
    phasemag)
19 g = 0.0                         // Conductance (mho/km/
    phasemag)
20
21 // Calculations
22 // Case(i) Using convergent series (Complex angles)
     method
23 z = r+i*2*pi*f*l             //
```

```

        Impedance (ohm/km)
24 Z = z*L                                     // Total
      series impedance(ohm)
25 y = g+%i*2*%pi*f*c                         // Shunt
      admittance(S/km)
26 Y = y*L                                     // Total
      shunt admittance(S)
27 A = 1+(Y*Z/2)+((Y*Z)**2/24)                //
      Constant
28 B = Z*(1+(Y*Z/6)+((Y*Z)**2/120))          //
      Constant(ohm)
29 C = Y*(1+(Y*Z/6)+((Y*Z)**2/120))          //
      Constant(mho)
30 D = A                                       //
      Constant
31 // Case(ii) Using convergent series(Real angles)
      method
32 gamma_1 = (Z*Y)**0.5                         // l
33 alpha_1 = real(gamma_1)                        // l
34 beta_1 = imag(gamma_1)                         // l
35 Z_c = (Z/Y)**0.5                             // Surge
      impedance(ohm)
36 A_2 = cosh(gamma_1)                           //
      Constant
37 B_2 = Z_c*sinh(gamma_1)                       //
      Constant(ohm)
38 C_2 = (1/Z_c)*sinh(gamma_1)                  //
      Constant(mho)
39 D_2 = A_2                                     //
      Constant
40
41 // Results
42 disp("PART II - EXAMPLE : 3.17 : SOLUTION :-")
43 printf("\nCase(i) : Using convergent series(Complex
      Angles) method")
44 printf("\nA = D = %.3 f % .1 f    " , abs(A),phasemag(A
      ))
45 printf("\nB = %. f % .1 f    ohm" , abs(B),phasemag(B))

```

```

46 printf("\nC = %.4 f % .1 f    mho \n" , abs(C) ,phasemag
        (C))
47 printf("\nCase( ii ) : Using convergent series (Real
        Angles) method")
48 printf("\nA = D = %.3 f % .1 f    " , abs(A_2) ,phasemag
        (A_2))
49 printf("\nB = %.1 f % .1 f    ohm" , abs(B_2) ,phasemag(
        B_2))
50 printf("\nC = %.4 f % .1 f    S \n" , abs(C_2) ,phasemag
        (C_2))
51 printf("\nNOTE: Slight change in obtained answer
        from that of textbook is due to more precision")

```

---

### Scilab code Exa 10.18 Sending end voltage and Current using convergent series meth

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
    PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.18 :
10 // Page number 148
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V_r = 220.0*10**3                                // Line voltage
    at receiving end(V)
15 Z = complex(40,200)                                // Impedance
    per phasemag(ohm)
16 Y = %i*0.0015                                     // Admittance(

```

```

        mho)
17 I_r = 200.0                                // Receiving
      end current(A)
18 PF_r = 0.95                                 // Lagging
      power factor
19
20 // Calculations
21 // Case(a)
22 A = 1+(Y*Z/2)+((Y*Z)**2/24)               // Constant
23 B = Z*(1+(Y*Z/6)+((Y*Z)**2/120)+((Y*Z)**3/5040)) // Constant(ohm)
24 C = Y*(1+(Y*Z/6)+((Y*Z)**2/120)+((Y*Z)**3/5040)) // Constant(mho)
25 D = A

      // Constant
26 E_r = V_r/3**0.5                            //
      Receiving end phasemag voltage(V)
27 I_r1 = I_r*exp(%i*-acos(PF_r))             // Line current(A)
28 E_s = A*E_r+B*I_r1                          // Sending
      end voltage(V)
29 E_s_ll = 3**0.5*E_s/1000.0                  // Sending end
      line voltage(kV)
30 // Case(b)
31 I_s = C*E_r+D*I_r1                         // Sending
      end current(A)
32
33 // Results
34 disp("PART II - EXAMPLE : 3.18 : SOLUTION :-")
35 printf("\nCase(a): Sending end voltage , E_s = %.1
      f %.2 f kV (line-to-line)" , abs(E_s_ll),
      phasemag(E_s_ll))

```

```

36 printf("\nCase(b): Sending end current , I_s = %.1
          f % .2 f A\n", abs(I_s), phasemag(I_s))
37 printf("\nNOTE: ERROR: Z = (40+j200) , not Z=(60+
          j200) as given in problem statement")
38 printf("\n      Changes in obtained answer from that
          of textbook is due to more precision")

```

---

### Scilab code Exa 10.19 Sending end voltage and Current using nominal pi and nominal

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
     PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.19 :
10 // Page number 148–149
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V_r = 220.0*10**3                                // Line voltage
           at receiving end(V)
15 Z = complex(40,200)                                // Impedance
           per phasemag(ohm)
16 Y = %i*0.0015                                     // Admittance(S
           )
17 I_R = 200.0                                       // Receiving
           end current(A)
18 PF_r = 0.95                                         // Lagging
           power factor
19

```

```

20 // Calculations
21 // Case(i) Nominal      method
22 // Case(a)
23 E_r = V_r/3**0.5                                //
24 I_r = I_R*exp(%i*-acos(PF_r))                  // Line current (A)
25 Y_2 = Y/2.0                                     //
26 I_c2 = Y_2*E_r                                    //
27 I = I_r+I_c2                                     //
28 IZ_drop = I*Z                                      //
29 E_s = E_r+IZ_drop                                //
30 E_s_kV = E_s/1000.0                               //
31 // Case(b)
32 I_c1 = E_s*Y_2                                    //
33 I_s = I+I_c1                                     //
34 // Case(ii) Nominal T method
35 // Case(a)
36 I_r_Z2 = I_r*Z/2                                 //
37 E = E_r+I_r_Z2                                   //
38 I_c = Y*E                                         //
39 I_s_2 = I_c+I_r                                    //
40 I_s_Z2 = I_s_2*(Z/2)                             //
41 E_s_2 = I_s_Z2+E                                //

```

```

        Sending end voltage(V)
42 E_s_2kV = E_s_2/1000.0 //  

        Sending end voltage(kV)

43  

44 // Results
45 disp("PART II - EXAMPLE : 3.19 : SOLUTION :-")
46 printf("\nCase(i): Nominal method")
47 printf("\n      Case(a): Sending end voltage , E_s  

        = %.1 f %.2 f kV", abs(E_s_kV),phasemag(E_s_kV))
48 printf("\n      Case(b): Sending end current , I_s  

        = %.1 f %.2 f A", abs(I_s),phasemag(I_s))
49 printf("\nCase(ii): Nominal T method")
50 printf("\n      Case(a): Sending end voltage , E_s  

        = %.1 f %.2 f kV", abs(E_s_2kV),phasemag(
E_s_2kV))
51 printf("\n      Case(b): Sending end current , I_s  

        = %.1 f %.2 f A \n", abs(I_s_2),phasemag(I_s_2))
52 printf("\nThe results are tabulated below")
53 printf("\n-----\n")
54 printf("\nMETHOD           E_s (kV)  

I_s (A)")
55 printf("\n-----\n")
56 printf("\nRigorous          3 *132.6   16   .46  

        209.8   39   .42   ")
57 printf("\nNominal          3 *%.1 f %.2 f  

        %.1 f %.2 f ", abs(E_s_kV),phasemag(
E_s_kV),abs(I_s),phasemag(I_s))
58 printf("\nNominal T         3 *%.1 f %.2 f  

        %.1 f %.2 f ", abs(E_s_2kV),phasemag(E_s_2kV),
abs(I_s_2),phasemag(I_s_2))
59 printf("\n-----\n")

```

” )

---

### Scilab code Exa 10.20 Sending end voltage Voltage regulation Transmission efficiency

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.20 :
10 // Page number 149–153
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 f = 50.0 // Frequency (Hz)
15 L = 280.0 // Line length (km)
16 Z = complex(35,140) // Series impedance (ohm)
17 Y = %i*930.0*10**-6 // Shunt admittance (S)
18 P_r = 40.0*10**6 // Power delivered (W)
19 V_r = 220.0*10**3 // Voltage at receiving
end(V)
20 PF_r = 0.9 // Lagging power factor
21
22 // Calculations
23 R = real(Z)

// Resistance of the line (ohm)
24 // Case(a)
25 I_r_a = P_r/(3**0.5*V_r*PF_r)*exp(%i*-acos(PF_r))
// Receiving end current (A)
```

```

26 I_s_a = I_r_a
          // Sending end current (A)
27 V_r_a = V_r/3**0.5
          //
          phasemag voltage at receiving end (V)
28 V_s_a = V_r_a+I_r_a*Z
          //
          Sending end voltage (V)
29 V_s_a_ll = 3**0.5*V_s_a
          //
          Sending end line voltage (V)
30 V_s_a_llkv = V_s_a_ll/1000.0
          //
          Sending end line voltage (kV)
31 reg_a = (abs(V_s_a_ll)-V_r)/V_r*100
          //
          Voltage regulation (%)
32 loss_a = 3*abs(I_r_a)**2*R
          //
          Line loss (W)
33 input_a = P_r+loss_a
          //
          Input to line (W)
34 n_a = P_r/input_a*100
          //
          Efficiency of transmission (%)
35 A_a = 1.0
          //
          Constant
36 B_a = Z
          //
          Constant (ohm)
37 C_a = 0
          //
          Constant (mho)
38 D_a = A_a

```

```

        // Constant
39 // Case(b)
40 V_b = V_r_a+I_r_a*Z/2
                           // Voltage drop
   across shunt admittance(V)
41 I_c_b = Y*V_b
           //
   Current through shunt admittance(A)
42 I_s_b = I_r_a+I_c_b
           // Sending end
   current (A)
43 V_s_b = V_b+I_s_b*Z/2
           // Sending end
   voltage (V)
44 V_s_b_ll = 3**0.5*V_s_b
           // Sending end
   line voltage (V)
45 V_s_b_llkv = V_s_b_ll/1000.0
           // Sending end line
   voltage (kV)
46 angle_V_Is_b = phasemag(I_s_b)
           // Angle between V_r and
   I_s_b( )
47 angle_V_Vs_b = phasemag(V_s_b)
           // Angle between V_r and
   V_s_b( )
48 angle_Is_Vs_b = angle_V_Is_b-angle_V_Vs_b
           // Angle between V_s_b and I_s_b( )
49 PF_s_b = cosd(angle_Is_Vs_b)
           // Sending end power
   factor
50 P_s_b = 3**0.5*abs(V_s_b_ll*I_s_b)*PF_s_b
           // Sending end power(W)
51 n_b = P_r/P_s_b*100
           // Efficiency
   of transmission(%)
52 reg_b = (abs(V_s_b_ll)-V_r)/V_r*100
           // Voltage regulation(%)

```

```

53 A_b = 1+(1.0/2)*Y*Z                                // Constant
54 B_b = Z*(1+(1.0/4)*Y*Z)                            // Constant (ohm)
55 C_b = Y
56 D_b = A_b                                         //
57 // Constant (mho)
58 V_s_ba = A_b*V_r_a+B_b*I_r_a                      // Sending end voltage(
59 V_s_ba_ll = 3**0.5*V_s_ba                           // Sending end line
60 V_s_ba_llkv = V_s_ba_ll/1000.0                      // Sending end line
61 I_s_ba = C_b*V_r_a+D_b*I_r_a                      // Sending end current(
62 angle_V_Is_ba = phasemag(I_s_ba)                  // Angle between V_r and
63 angle_V_Vs_ba = phasemag(V_s_ba)                  // Angle between V_r and
64 angle_Is_Vs_ba = angle_V_Is_ba-angle_V_Vs_ba      // Angle between V_s_b and I_s_b( )
65 PF_s_ba = cosd(angle_Is_Vs_ba)                    // Sending end power
66 P_s_ba = 3**0.5*abs(V_s_ba_ll*I_s_ba)*PF_s_ba    // Sending end power(W)
67 n_ba = P_r/P_s_ba*100                               // Efficiency of

```

```

        transmission (%)

68 reg_ba = (abs(V_s_ba_ll)-V_r)/V_r*100
                // Voltage regulation (%)

69 // Case(c)
70 I_c2_c = Y/2.0*V_r_a
                // Current through shunt admittance at receiving
                end(A)

71 I_c = I_r_a+I_c2_c
                // Current through impedance(A)

72 V_s_c = V_r_a+I_c*Z
                // Sending end voltage(V)

73 V_s_c_ll = 3**0.5*V_s_c
                // Sending end line voltage(V)

74 V_s_c_llkv = V_s_c_ll/1000.0
                // Sending end line voltage(kV)

75 I_c1_c = V_s_c*Y/2.0
                // Current through shunt admittance at sending end(
                A)

76 I_s_c = I_c+I_c1_c
                // Sending end current(A)

77 angle_V_Is_c = phasemag(I_s_c)
                // Angle between V_r and I_s_c( )

78 angle_V_Vs_c = phasemag(V_s_c)
                // Angle between V_r and V_s_c( )

79 angle_Is_Vs_c = angle_V_Is_c-angle_V_Vs_c
                // Angle between V_s_c and I_s_c( )

80 PF_s_c = cosd(angle_Is_Vs_c)
                // Sending end power factor

81 P_s_c = 3**0.5*abs(V_s_c_ll*I_s_c)*PF_s_c
                // Sending end power(W)

82 n_c = P_r/P_s_c*100
                // Efficiency of transmission(%)

83 reg_c = (abs(V_s_c_ll)-V_r)/V_r*100
                // Voltage regulation(%)

84 A_c = 1+(1.0/2)*Y*Z
                // Constant

85 B_c = Z
                // Constant(ohm)

```

```

86 C_c = Y*(1+(1.0/4)*Y*Z) // Constant(mho)
87 D_c = A_c // Constant
88 // Alternative solution for case(c)
89 V_s_ca = A_c*V_r_a+B_c*I_r_a // Sending end voltage(V)
90 V_s_ca_ll = 3**0.5*V_s_ca // Sending end line voltage(V)
91 V_s_ca_llkv = V_s_ca_ll/1000.0 // Sending end line voltage(kV)
92 I_s_ca = C_c*V_r_a+D_c*I_r_a // Sending end current(A)
93 angle_V_Is_ca = phasemag(I_s_ca) // Angle between V_r and I_s_c( )
94 angle_V_Vs_ca = phasemag(V_s_ca) // Angle between V_r and V_s_c( )
95 angle_Is_Vs_ca = angle_V_Is_ca-angle_V_Vs_ca // Angle between V_s_b and I_s_c( )
96 PF_s_ca = cosd(angle_Is_Vs_ca) // Sending end power factor
97 P_s_ca = 3**0.5*abs(V_s_ca_ll*I_s_ca)*PF_s_ca // Sending end power(W)
98 n_ca = P_r/P_s_ca*100 // Efficiency of transmission(%)
99 reg_ca = (abs(V_s_ca_ll)-V_r)/V_r*100 // Voltage regulation(%)
100 // Case(d).(i)
101 gamma_1 = (Y*Z)**0.5 // 1
102 Z_c = (Z/Y)**0.5 // Surge impedance(ohm)
103 V_s_d1 = V_r_a*cosh(gamma_1)+I_r_a*Z_c*sinh(gamma_1) // Sending end voltage(V)
104 V_s_d1_ll = 3**0.5*V_s_d1 // 

```

```

        Sending end line voltage(V)
105 V_s_d1_llkv = V_s_d1_ll/1000.0
                                // Sending
        end line voltage(kV)
106 I_s_d1 = V_r_a/Z_c*sinh(gamma_1)+I_r_a*cosh(gamma_1)
                                // Sending end current(A)
107 angle_V_Is_d1 = phasemag(I_s_d1)
                                // Angle
        between V_r and I_s_d( )
108 angle_V_Vs_d1 = phasemag(V_s_d1)
                                // Angle
        between V_r and V_s_d( )
109 angle_Is_Vs_d1 = angle_V_Is_d1-angle_V_Vs_d1
                                // Angle between V_s_d and
        I_s_d( )
110 PF_s_d1 = cosd(angle_Is_Vs_d1)
                                // Sending
        end power factor
111 P_s_d1 = 3**0.5*abs(V_s_d1_ll*I_s_d1)*PF_s_d1
                                // Sending end power(W)
112 n_d1 = P_r/P_s_d1*100
                                //
        Efficiency of transmission(%)
113 reg_d1 = (abs(V_s_d1_ll)-V_r)/V_r*100
                                // Voltage
        regulation(%)
114 A_d1 = cosh(gamma_1)
                                //
        Constant
115 B_d1 = Z_c*sinh(gamma_1)
                                //
        Constant(ohm)
116 C_d1 = (1/Z_c)*sinh(gamma_1)
                                //
        Constant(mho)
117 D_d1 = A_d1
                                //
        Constant

```

```

118 // Case(d).(ii)
119 A_d2 = (1+(Y*Z/2)+((Y*Z)**2/24.0))
           // Constant
120 B_d2 = Z*(1+(Y*Z/6)+((Y*Z)**2/120))
           // Constant(ohm)
121 C_d2 = Y*(1+(Y*Z/6)+((Y*Z)**2/120))
           // Constant(mho)
122 D_d2 = A_d2
           //
           Constant
123 V_s_d2 = A_d2*V_r_a+B_d2*I_r_a
           // Sending end voltage(
           V)
124 V_s_d2_ll = 3**0.5*V_s_d2
           // Sending end
           line voltage(V)
125 V_s_d2_llkv = V_s_d2_ll/1000.0
           // Sending end line
           voltage(kV)
126 I_s_d2 = C_d2*V_r_a+D_d2*I_r_a
           // Sending end current(
           A)
127 angle_V_Is_d2 = phasemag(I_s_d2)
           // Angle between V_r and
           I_s_d( )
128 angle_V_Vs_d2 = phasemag(V_s_d2)
           // Angle between V_r and
           V_s_d( )
129 angle_Is_Vs_d2 = angle_V_Is_d2-angle_V_Vs_d2
           // Angle between V_s_d and I_s_d( )
130 PF_s_d2 = cosd(angle_Is_Vs_d2)
           // Sending end power
           factor
131 P_s_d2 = 3**0.5*abs(V_s_d2_ll*I_s_d2)*PF_s_d2
           // Sending end power(W)
132 n_d2 = P_r/P_s_d2*100
           // Efficiency
           of transmission(%)

```

```

133 reg_d2 = (abs(V_s_d2_11)-V_r)/V_r*100
                // Voltage regulation (%)
134
135 // Results
136 disp("PART II - EXAMPLE : 3.20 : SOLUTION :-")
137 printf("\nCase(a) : Short line approximation")
138 printf("\nSending end voltage , V_s = %.1 f % .1 f
        kV (line-to-line)" , abs(V_s_a_llkv),phasemag(
        V_s_a_llkv))
139 printf("\nVoltage regulation = %.1 f percent" , reg_a)
140 printf("\nTransmission efficiency ,      = %.1 f percent
        " , n_a)
141 printf("\nA = D = %. f " , A_a)
142 printf("\nB = %.1 f % .1 f ohm" , abs(B_a),phasemag(
        B_a))
143 printf("\nC = %. f \n" , C_a)
144 printf("\nCase(b) : Nominal T method approximation")
145 printf("\nSending end voltage , V_s = %.1 f % .1 f
        kV (line-to-line)" , abs(V_s_b_llkv),phasemag(
        V_s_b_llkv))
146 printf("\nVoltage regulation = %.2 f percent" , reg_b)
147 printf("\nTransmission efficiency ,      = %.1 f percent
        " , n_b)
148 printf("\nA = D = %.3 f % .2 f " , abs(A_b),phasemag(
        A_b))
149 printf("\nB = %.1 f % .1 f ohm" , abs(B_b),phasemag(
        B_b))
150 printf("\nC = %.2 e % . f S " , abs(C_b),phasemag(
        C_b))
151 printf("\n\tALTERNATIVE SOLUTION:")
152 printf("\n\tSending end voltage , V_s = %.1 f % .1 f
        kV (line-to-line)" , abs(V_s_ba_llkv),phasemag(
        V_s_ba_llkv))
153 printf("\n\tVoltage regulation = %.2 f percent" ,
        reg_ba)
154 printf("\n\tTransmission efficiency ,      = %.1 f
        percent" , n_ba)
155 printf("\n\tA = D = %.3 f % .2 f " , abs(A_b),

```

```

    phasemag(A_b))
156 printf("\n\tB = %.1 f % .1 f ohm", abs(B_b),
    phasemag(B_b))
157 printf("\n\tC = %.2 e % .f S \n", abs(C_b),
    phasemag(C_b))
158 printf("\nCase(c): Nominal method approximation")
159 printf("\nSending end voltage, V_s = %. f % .1 f kV
    (line-to-line)", abs(V_s_c_llkv), phasemag(
    V_s_c_llkv))
160 printf("\nVoltage regulation = %.2 f percent", reg_c)
161 printf("\nTransmission efficiency , = %.1 f percent
    ", n_c)
162 printf("\nA = D = %.3 f % .2 f ", abs(A_c), phasemag(
    A_c))
163 printf("\nB = %.1 f % .1 f ohm", abs(B_c), phasemag(
    B_c))
164 printf("\nC = %.2 e % .1 f mho", abs(C_c), phasemag(
    C_c))
165 printf("\n\tALTERNATIVE SOLUTION:")
166 printf("\n\tSending end voltage, V_s = %.1 f % .1 f
    kV (line-to-line)", abs(V_s_ca_llkv), phasemag(
    V_s_ca_llkv))
167 printf("\n\tVoltage regulation = %.2 f percent",
    reg_ca)
168 printf("\n\tTransmission efficiency , = %.1 f
    percent", n_ca)
169 printf("\n\tA = D = %.3 f % .2 f ", abs(A_c),
    phasemag(A_c))
170 printf("\n\tB = %.1 f % .1 f ohm", abs(B_c),
    phasemag(B_c))
171 printf("\n\tC = %.2 e % .f S \n", abs(C_c),
    phasemag(C_c))
172 printf("\nCase(d): Long Line Rigorous Solution")
173 printf("\n Case(i): Using Convergent Series (Real
    Angles) Method")
174 printf("\n Sending end voltage, V_s = %. f % .1 f
    kV (line-to-line)", abs(V_s_d1_llkv), phasemag(
    V_s_d1_llkv))

```

```

175 printf("\n Voltage regulation = %.2f percent" ,
      reg_d1)
176 printf("\n Transmission efficiency ,      = %.1f
      percent" , n_d1)
177 printf("\n A = D = %.3f % .2f " , abs(A_d1) ,
      phasemag(A_d1))
178 printf("\n B = %. f % .1f ohm" , abs(B_d1) , phasemag
      (B_d1))
179 printf("\n C = %.2 e % .1f mho \n" , abs(C_d1) ,
      phasemag(C_d1))
180 printf("\n Case(ii): Using Convergent Series (
      Complex Angles) Method")
181 printf("\n Sending end voltage , V_s = %. f % .1f
      kV (line-to-line)" , abs(V_s_d2_llkv) , phasemag(
      V_s_d2_llkv))
182 printf("\n Voltage regulation = %.2f percent" ,
      reg_d2)
183 printf("\n Transmission efficiency ,      = %.1f
      percent" , n_d2)
184 printf("\n A = D = %.3 f % .2f " , abs(A_d2) ,
      phasemag(A_d2))
185 printf("\n B = %.1 f % .1f ohm" , abs(B_d2) ,
      phasemag(B_d2))
186 printf("\n C = %.2 e % .1f mho \n" , abs(C_d2) ,
      phasemag(C_d2))
187 printf("\n NOTE: Changes in obtained answer from that
      of textbook is due to more precision")

```

---

### Scilab code Exa 10.21 Sending end voltage Current Power factor and Efficiency of t

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5

```

```

6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
    PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.21 :
10 // Page number 153
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V_r = 132.0*10**3 // Line voltage at receiving end(V)
15 P_L = 45.0*10**6 // Load delivered (VA)
16 PF_r = 0.8 // Lagging power factor
17 A = 0.99*exp(%i*0.3*pi/180) // Constant
18 B = 70.0*exp(%i*69.0*pi/180) // Constant (ohms)
19 C = A // Constant
20 D = 4.0*10**-4*exp(%i*90.0*pi/180) // Constant
21
22 // Calculations
23 E_r = V_r/3**0.5 // Receiving end phasemag voltage(V)
24 I_r = P_L/(3**0.5*V_r)*exp(%i*-acos(PF_r)) // Line current (A)
25 E_s = A*E_r+B*I_r // Sending end voltage(V)
26 E_s_llkV = 3**0.5*E_s/1000.0 // Sending end line voltage (kV)
27 I_s = C*I_r+D*E_r

```

```

//  

    Sending end current(A)  

28 angle_Er_Es = phasemag(E_s)                                // Angle between  

    E_r and E_s( )  

29 angle_Er_Is = phasemag(I_s)                                // Angle between  

    E_r and I_s( )  

30 angle_Es_Is = angle_Er_Es-angle_Er_Is                      // Angle between E_s and I_s(  

    )  

31 PF_s = cosd(angle_Es_Is)                                    // Sending end  

    power factor  

32 P_s = 3*abs(E_s*I_s)*PF_s                                  // Sending end  

    power(W)  

33 P_skW = P_s/1000.0                                         // Sending  

    end power(kW)  

34 P_r = P_L*PF_r                                              //  

    Receiving end power(W)  

35 n = P_r/P_s*100                                            //  

    Transmission efficiency(%)
36
37 // Results
38 disp("PART II - EXAMPLE : 3.21 : SOLUTION :-")
39 printf("\nCase(i) : Sending end voltage , E_s = %.1
    f % .f kV (line-to-line)" , abs(E_s_11kV) ,
    phasemag(E_s_11kV))
40 printf("\nCase(ii) : Sending end current , I_s = %.1
    f % .1 f A" , abs(I_s) , phasemag(I_s))
41 printf("\nCase(iii) : Sending end power , P_s = %.f kW
    " , P_skW)
42 printf("\nCase(iv) : Efficiency of transmission = %
    .2 f percent \n" , n)

```

43 **printf**("\\nNOTE: Changes in obtained answer from that  
textbook is due to more precision")

---

**Scilab code Exa 10.23 Overall constants A B C D**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
    PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.23 :
10 // Page number 156
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 A_1 = 0.98*exp(%i*2.0*%pi/180)           // Constant
    of 1st line
15 B_1 = 28.0*exp(%i*69.0*%pi/180)         // Constant
    of 1st line(ohms)
16 C_1 = 0.0002*exp(%i*88.0*%pi/180)       // Constant
    of 1st line(mho)
17 D_1 = A_1                                     // Constant
    of 1st line
18 A_2 = 0.95*exp(%i*3.0*%pi/180)          // Constant
    of 2nd line
19 B_2 = 40.0*exp(%i*85.0*%pi/180)         // Constant
    of 2nd line(ohms)
20 C_2 = 0.0004*exp(%i*90.0*%pi/180)       // Constant
    of 2nd line(mho)
21 D_2 = A_2                                     // Constant
```

```

        of 2nd line

22
23 // Calculations
24 A = A_1*A_2+B_1*C_2          // Constant
25 B = A_1*B_2+B_1*D_2          // Constant(ohm)
26 C = C_1*A_2+D_1*C_2          // Constant(mho)
27 D = C_1*B_2+D_1*D_2          // Constant
28
29 // Results
30 disp("PART II - EXAMPLE : 3.23 : SOLUTION :-")
31 printf("\nA = %.3 f % .1 f    ", abs(A),phasemag(A))
32 printf("\nB = %.1 f % .f      ohm", abs(B),phasemag(B))
33 printf("\nC = %.6 f % .1 f    mho", abs(C),phasemag(C))
34 printf("\nD = %.3 f % .1 f    ", abs(D),phasemag(D))

```

---

#### Scilab code Exa 10.24 Values of constants A0 B0 C0 D0

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
     PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.24 :
10 // Page number 156-157
11 clear ; clc ; close ; // Clear the work space and
     console
12
13 // Given data
14 A = 0.94*exp(%i*1.5*pi/180)          // Constant
15 B = 150.0*exp(%i*67.2*pi/180)          // Constant(ohm)

```

```

)
16 D = A                                     // Constant
17 Y_t = 0.00025*exp(%i*-75.0*%pi/180)    // Shunt
      admittance(mho)
18 Z_t = 100.0*exp(%i*70.0*%pi/180)        // Series
      impedance(ohm)
19
20 // Calculations
21 C = (A*D-1)/B                         // Constant (mho)
22 A_0 = A*(1+Y_t*Z_t)+B*Y_t             // Constant
23 B_0 = A*Z_t+B                          // Constant (ohm)
24 C_0 = C*(1+Y_t*Z_t)+D*Y_t             // Constant (mho)
25 D_0 = C*Z_t+D                          // Constant
26
27 // Results
28 disp("PART II - EXAMPLE : 3.24 : SOLUTION :-")
29 printf("\nA_0 = %.3 f % .f      ", abs(A_0),phasemag(
      A_0))
30 printf("\nB_0 = %. f % .1 f      ohm", abs(B_0),phasemag(
      B_0))
31 printf("\nC_0 = %.6 f % .1 f      mho", abs(C_0),
      phasemag(C_0))
32 printf("\nD_0 = %.3 f % .1 f      \n", abs(D_0),phasemag(
      D_0))
33 printf("\nNOTE: Changes in obtained answer from that
      of textbook is due to more precision")

```

---

### Scilab code Exa 10.25 Maximum power transmitted Receiving end power factor and Total

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION

```

```

7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.25 :
10 // Page number 163
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 z = complex(0.2,0.6)           // Per phase impedance(ohm)
15 V_r = 6351.0                  // Receiving end voltage
per phase(V)
16 reg = 7.5/100.0                // Voltage regulation
17
18 // Calculations
19 V_s = (1+reg)*V_r
// Sending end voltage per phase(V)
20 R = real(z)
// Resistance of the line(ohm)
21 X = imag(z)
// Reactance of the line(ohm)
22 Z = (R**2+X**2)**0.5
// Impedance per phase(ohm)
23 P_m = (V_r**2/Z)*((Z*V_s/V_r)-R)
// Maximum power transmitted through line(W/phase
)
24 P_m_MW = P_m/10**6
// Maximum power transmitted through line(MW/
phase)
25 P_m_MWtotal = 3*P_m_MW
// Total maximum power(MW)
26 Q = -(V_r**2*X)/Z**2
// Reactive power per phase(Var)
27 Q_MW = Q/10**6
// Reactive power per phase(MVAR)
28 phi_r = atand(abs(Q_MW/P_m_MW))
// r ( )
29 PF_r = cosd(phi_r)

```

```

        // Receiving end lagging PF
30 I = P_m/(V_r*PF_r)
        // Current delivered (A)
31 I_KA = I/1000.0
        // Current delivered (KA)
32 loss = 3*I**2*R
        // Total line loss (W)
33 loss_MW = loss/10**6
        // Total line loss (MW)
34
35 // Results
36 disp("PART II - EXAMPLE : 3.25 : SOLUTION :-")
37 printf("\nMaximum power transmitted through the line
        , P_m = %.1f MW", P_m_MWtotal)
38 printf("\nReceiving end power factor = %.2f (lagging
        )", PF_r)
39 printf("\nTotal line loss = %.2f MW", loss_MW)

```

---

**Scilab code Exa 10.26 Maximum power that can be transferred to the load**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 3: STEADY STATE CHARACTERISTICS AND
    PERFORMANCE OF TRANSMISSION LINES
8
9 // EXAMPLE : 3.26 :
10 // Page number 163-164
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data

```

```

14 L = 100.0 // Length of line(km)
15 PF_r = 1.0 // Receiving end Power factor
16 Z_c = 400.0 // Characteristic impedance(
    ohm)
17 beta = 1.2*10**-3 // Propagation constant(rad/
    km)
18 V_s = 230.0 // Sending end voltage(kV)
19
20 // Calculations
21 beta_L = beta*L // (rad)
22 beta_L_d = beta_L*180/%pi // ( )
23 A = cosd(beta_L) // Constant
24 B = %i*Z_c*sin(beta_L) // Constant
25 alpha_angle = phasemag(A) // ( )
26 beta_angle = phasemag(B) // ( )
27 V_r = V_s // Receiving end
    voltage due to lossless line(kV)
28 P_max = (V_s*V_r/abs(B))-(abs(A)*V_r**2/abs(B))*cosd
    (beta_angle-alpha_angle) // Maximum power
    transferred(MW)
29
30 // Results
31 disp("PART II - EXAMPLE : 3.26 : SOLUTION :-")
32 printf("\nMaximum power that can be transferred to
    the load at receiving end , P_max = %.f MW \n",
    P_max)
33 printf("\nNOTE: Changes in obtained answer from that
    of textbook is due to more precision")

```

---

# Chapter 11

## OVERHEAD LINE INSULATORS

Scilab code Exa 11.1 Ratio of capacitance Line voltage and String efficiency

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 4: OVERHEAD LINE INSULATORS
8
9 // EXAMPLE : 4.1 :
10 // Page number 183
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 V_1 = 9.0          // Potential across top unit(kV)
15 V_2 = 11.0          // Potential across middle unit(kV)
16 n = 3.0           // Number of disc insulators
17
18 // Calculations
```

```

19 // Case(a)
20 K = (V_2-V_1)/V_1           // Ratio of capacitance b/
    w pin & earth to self capacitance
21 // Case(b)
22 V_3 = V_2+(V_1+V_2)*K      // Potential across bottom
    unit (kV)
23 V = V_1+V_2+V_3            // Voltage between line
    and earth (kV)
24 V_1 = 3**0.5*V             // Line voltage (kV)
25 // Case(c)
26 eff = V/(n*V_3)*100        // String efficiency (%)
27
28 // Results
29 disp("PART II - EXAMPLE : 4.1 : SOLUTION :-")
30 printf("\nCase(a): Ratio of capacitance b/w pin &
    earth to self-capacitance of each unit , K = %.2f
    ", K)
31 printf("\nCase(b): Line voltage = %.2f kV", V_1)
32 printf("\nCase(c): String efficiency = %.f percent",
    eff)

```

---

### Scilab code Exa 11.2 Mutual capacitance of each unit in terms of C

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 4: OVERHEAD LINE INSULATORS
8
9 // EXAMPLE : 4.2 :
10 // Page number 183-184
11 clear ; clc ; close ; // Clear the work space and
    console

```

```

12
13 // Given data
14 m = 10.0           // Mutual capacitance of top
15      insulator in terms of C
16 // Calculations
17 X = 1+m           // Mutual capacitance in
18      terms of C
19 Y = (1.0+2)+m     // Mutual capacitance in
20      terms of C
21 Z = (1.0+2+3)+m   // Mutual capacitance in
22      terms of C
23 U = (1.0+2+3+4)+m // Mutual capacitance in
24      terms of C
25 V = (1.0+2+3+4+5)+m // Mutual capacitance in
26      terms of C
27
28 // Results
29 disp("PART II - EXAMPLE : 4.2 : SOLUTION :-")
30 printf("\nMutual capacitance of each unit:")
31 printf("\n X = %.f*C", X)
32 printf("\n Y = %.f*C", Y)
33 printf("\n Z = %.f*C", Z)
34 printf("\n U = %.f*C", U)
35 printf("\n V = %.f*C", V)

```

---

### Scilab code Exa 11.3 Voltage distribution over a string of three suspension insula

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 4: OVERHEAD LINE INSULATORS

```

```

8
9 // EXAMPLE : 4.3 :
10 // Page number 184
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 n = 3.0           // Number of insulators
15
16 // Calculations
17 V_1 = 155.0/475.0          // Potential across top
    unit
18 V_2 = 154.0/155.0*V_1      // Potential across
    middle unit
19 V_3 = 166.0/155.0*V_1      // Potential across
    bottom unit
20 eff = 100/(n*V_3)          // String efficiency (%)
21
22 // Results
23 disp("PART II – EXAMPLE : 4.3 : SOLUTION :-")
24 printf("\nVoltage across top unit , V_1 = %.3f *V",
    V_1)
25 printf("\nVoltage across middle unit , V_2 = %.3f *V",
    V_2)
26 printf("\nVoltage across bottom unit , V_3 = %.2f *V",
    V_3)
27 printf("\nString efficiency = %.2f percent" , eff)

```

---

#### Scilab code Exa 11.4 Line to neutral voltage and String efficiency

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5

```

```

6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 4: OVERHEAD LINE INSULATORS
8
9 // EXAMPLE : 4.4 :
10 // Page number 184–185
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V_3 = 17.5           // Voltage across line unit(kV)
15 c = 1.0/8            // Shunt capacitance = 1/8 of
    insulator capacitance
16 n = 3.0              // Number of insulators
17
18 // Calculations
19 K = c                  // String constant
20 V_1 = V_3/(1+3*K+K**2) // Voltage across top
    unit(kV)
21 V_2 = (1+K)*V_1        // Voltage across middle
    unit(kV)
22 V = V_1+V_2+V_3        // Voltage between line
    & earth(kV)
23 eff = V*100/(n*V_3)    // String efficiency(%)
24
25 // Results
26 disp("PART II – EXAMPLE : 4.4 : SOLUTION :-")
27 printf("\nLine to neutral voltage , V = %.2f kV" , V)
28 printf("\nString efficiency = %.2f percent" , eff)

```

---

### Scilab code Exa 11.5 Value of line to pin capacitance

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION

```

```

5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 4: OVERHEAD LINE INSULATORS
8
9 // EXAMPLE : 4.5 :
10 // Page number 185
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 n = 8.0           // Number of insulators
15
16 // Calculations
17 A = 1.0/(n-1)      // Line to pin capacitance
18 B = 2.0/(n-2)      // Line to pin capacitance
19 C = 3.0/(n-3)      // Line to pin capacitance
20 D = 4.0/(n-4)      // Line to pin capacitance
21 E = 5.0/(n-5)      // Line to pin capacitance
22 F = 6.0/(n-6)      // Line to pin capacitance
23 G = 7.0/(n-7)      // Line to pin capacitance
24
25 // Results
26 disp("PART II - EXAMPLE : 4.5 : SOLUTION :-")
27 printf("\nLine-to-pin capacitance are:")
28 printf("\n A = %.3f*C" , A)
29 printf("\n B = %.3f*C" , B)
30 printf("\n C = %.3f*C" , C)
31 printf("\n D = %.3f*C" , D)
32 printf("\n E = %.3f*C" , E)
33 printf("\n F = %.3f*C" , F)
34 printf("\n G = %.3f*C" , G)

```

---

**Scilab code Exa 11.6 Voltage distribution as a percentage of voltage of conductor**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
```

```

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 4: OVERHEAD LINE INSULATORS
8
9 // EXAMPLE : 4.6 :
10 // Page number 186
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 m = 6.0           // Mutual capacitance
15 n = 5.0           // Number of insulators
16
17 // Calculations
18 E_4 = (1+(1/m))
    // Voltage across 4th insulator as percent of E_5
    (%)
19 E_3 = (1+(3/m)+(1/m**2))
    // Voltage across 3rd insulator as percent of E_5
    (%)
20 E_2 = (1+(6/m)+(5/m**2)+(1/m**3))
    // Voltage across 2nd insulator as percent of E_5
    (%)
21 E_1 = (1+(10/m)+(15/m**2)+(7/m**3)+(1/m**4))
    // Voltage across 1st insulator as percent of E_5
    (%)
22 E_5 = 100/(E_4+E_3+E_2+E_1+1)
    // Voltage across 5th insulator as percent of E_5
    (%)
23 E4 = E_4*E_5
    // Voltage across 4th insulator as percent of E_5
    (%)
24 E3 = E_3*E_5
    // Voltage across 3rd insulator as percent of E_5
    (%)

```

```

25 E2 = E_2*E_5
    // Voltage across 2nd insulator as percent of E_5
    (%)
26 E1 = E_1*E_5
    // Voltage across 1st insulator as percent of E_5
    (%)
27 eff = 100/(n*E1/100)
    // String efficiency (%)

28
29 // Results
30 disp("PART II - EXAMPLE : 4.6 : SOLUTION :-")
31 printf("\nVoltage distribution as a percentage of
        voltage of conductor to earth are:")
32 printf("\n E_1 = %.2f percent", E1)
33 printf("\n E_2 = %.2f percent", E2)
34 printf("\n E_3 = %.1f percent", E3)
35 printf("\n E_4 = %.1f percent", E4)
36 printf("\n E_5 = %.2f percent", E_5)
37 printf("\nString efficiency = %.f percent \n", eff)
38 printf("\nNOTE: Changes in obtained answer from that
        of textbook is due to more precision")

```

---

### Scilab code Exa 11.7 Voltage across each insulator as a percentage of line voltage

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 4: OVERHEAD LINE INSULATORS
8
9 // EXAMPLE : 4.7 :
10 // Page number 186-187
11 clear ; clc ; close ; // Clear the work space and

```

```

        console
12
13 // Given data
14 n = 3.0                      // Number of insulators
15 C_1 = 0.2                     // Capacitance in terms of C
16 C_2 = 0.1                     // Capacitance in terms of C
17
18 // Calculations
19 // Without guard ring
20 e_2_a = 13.0/13.3            // Potential
   across middle unit as top unit
21 e_1_a = 8.3/6.5*e_2_a       // Potential
   across bottom unit
22 E_a = 1+(1/(8.3/6.5))+(1/e_1_a) // Voltage in
   terms of e_1
23 eff_a = E_a/n*100           // String
   efficiency (%)
24 e1_a = 1/E_a                // Voltage across
   bottom unit as a % of line voltage
25 e2_a = 1/(8.3/6.5)*e1_a    // Voltage across
   middle unit as a % of line voltage
26 e3_a = 1/e_1_a*e1_a        // Voltage across
   top unit as a % of line voltage
27 // With guard ring
28 e_2_b = 15.4/15.5          // Potential
   across middle unit as top unit
29 e_1_b = 8.3/7.7*e_2_b      // Potential
   across bottom unit
30 E_b = 1+(1/(8.3/7.7))+(1/e_1_b) // Voltage in
   terms of e_1
31 eff_b = E_b/n*100          // String
   efficiency (%)
32 e1_b = 1/E_b               // Voltage across
   bottom unit as a % of line voltage
33 e2_b = 1/(8.3/7.7)*e1_b    // Voltage across
   middle unit as a % of line voltage
34 e3_b = 1/e_1_b*e1_b        // Voltage across
   top unit as a % of line voltage

```

```

35
36 // Results
37 disp("PART II - EXAMPLE : 4.7 : SOLUTION :-")
38 printf("\nWithout guard ring:")
39 printf("\n Voltage across bottom unit , e_1 = %.2f *E"
       , e1_a)
40 printf("\n Voltage across bottom unit , e_2 = %.2f *E"
       , e2_a)
41 printf("\n Voltage across bottom unit , e_3 = %.2f *E"
       , e3_a)
42 printf("\n String efficiency = %.1f percent \n" ,
       eff_a)
43 printf("\nWith guard ring:")
44 printf("\n Voltage across bottom unit , e_1 = %.2f *E"
       , e1_b)
45 printf("\n Voltage across bottom unit , e_2 = %.2f *E"
       , e2_b)
46 printf("\n Voltage across bottom unit , e_3 = %.3f *E"
       , e3_b)
47 printf("\n String efficiency = %.2f percent" , eff_b)

```

---

### Scilab code Exa 11.8 Voltage across each insulator as a percentage of line voltage

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 4: OVERHEAD LINE INSULATORS
8
9 // EXAMPLE : 4.8 :
10 // Page number 187-188
11 clear ; clc ; close ; // Clear the work space and
                           console

```

```

12
13 // Given data
14 n = 3.0 // Number of insulators
15
16 // Calculations
17 V_1 = 0.988 // Voltage across top
    unit as middle unit
18 V_3 = 1.362 // Voltage across bottom
    unit as middle unit
19 V_2 = 1/(V_1+1+V_3) // Voltage across middle
    unit as % of line voltage to earth
20 V1 = V_1*V_2*100 // Voltage across top
    unit as % of line voltage to earth
21 V2 = V_2*100 // Voltage across middle
    unit as % of line voltage to earth
22 V3 = V_3*V_2*100 // Voltage across bottom
    unit as % of line voltage to earth
23 eff = 100/(n*V3/100) // String efficiency (%)
24
25 // Results
26 disp("PART II - EXAMPLE : 4.8 : SOLUTION :-")
27 printf("\nCase(a): Voltage across top unit as a
    percentage of line voltage to earth , V_1 = %.2f
    percent", V1)
28 printf("\n        Voltage across middle unit as a
    percentage of line voltage to earth , V_2 = %.2f
    percent", V2)
29 printf("\n        Voltage across bottom unit as a
    percentage of line voltage to earth , V_3 = %.2f
    percent", V3)
30 printf("\nCase(b): String efficiency = %.2f percent"
    , eff)

```

---

**Scilab code Exa 11.9 Voltage on the line end unit and Value of capacitance required**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 4: OVERHEAD LINE INSULATORS
8
9 // EXAMPLE : 4.9 :
10 // Page number 188
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 n = 3.0           // Number of insulators
15 V = 20.0          // Voltage across each
   conductor (kV)
16 c = 1.0/5         // Capacitance ratio
17
18 // Calculations
19 V_2 = 6.0/5.0      // Voltage across middle
   unit as top unit
20 V_1 = V/(1+2*V_2) // Voltage across top unit (
   kV)
21 V_3 = V_2*V_1      // Voltage across bottom
   unit (kV)
22 C_x = c*(1+(1/V_2)) // Capacitance required
23
24 // Results
25 disp("PART II - EXAMPLE : 4.9 : SOLUTION :-")
26 printf("\nCase(a): Voltage on the line-end unit , V_3
   = %.2f kV", V_3)
27 printf("\nCase(b): Value of capacitance required , Cx
   = %.3f *C", C_x)

```

---

# Chapter 12

## MECHANICAL DESIGN OF OVERHEAD LINES

Scilab code Exa 12.1 Weight of conductor

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 5: MECHANICAL DESIGN OF OVERHEAD LINES
8
9 // EXAMPLE : 5.1 :
10 // Page number 198
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 u = 5758.0          // Ultimate strength (kg)
15 S = 2.0             // Sag(m)
16 s = 2.0             // Factor of safety
17 L = 250.0           // Span length(m)
18
```

```

19 // Calculations
20 T = u/s                                     // Allowable
     max tension (kg)
21 w = S*8.0*T/L**2                           // weight (kg/
     m)
22 l = L/2                                     // Half span
     length (m)
23 half_span = l+(w**2*l**3/(6*T**2))        // Half span
     length (m)
24 total_length = 2*half_span                  // Total
     length (m)
25 weight = w*total_length                     // Weight of
     conductor (kg)
26
27 // Results
28 disp("PART II - EXAMPLE : 5.1 : SOLUTION :-")
29 printf("\nWeight of conductor = %.2f kg", weight)

```

---

**Scilab code Exa 12.2 Point of maximum sag at the lower support**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 5: MECHANICAL DESIGN OF OVERHEAD LINES
8
9 // EXAMPLE : 5.2 :
10 // Page number 198
11 clear ; clc ; close ; // Clear the work space and
     console
12
13 // Given data
14 L = 250.0                                // Span length (m)

```

```

15 h = 10.0          // Difference in height (m)
16 r = 1.0           // Radius of conductor (cm)
17 w = 2.5           // Weight of conductor (kg/m)
18 wind = 1.2         // Wind load (kg/m)
19 s = 3.0           // Factor of safety
20 tensile = 4300.0   // Maximum tensile strength (kg
                     /sq.cm)

21
22 // Calculations
23 W = (w**2+wind**2)**0.5      // Total pressure on
                                conductor (kg/m)
24 f = tensile/s               // Permissible stress
                                in conductor (kg/sq.cm)
25 a = %pi*r**2                // Area of the
                                conductor (sq.cm)
26 T = f*a                     // Allowable max
                                tension (kg)
27 x = (L/2)-(T*h/(L*W))       // Point of maximum
                                sag at the lower support (m)
28
29 // Results
30 disp("PART II - EXAMPLE : 5.2 : SOLUTION :-")
31 printf("\nPoint of maximum sag at the lower support ,
        x = %.2f metres", x)

```

---

### Scilab code Exa 12.3 Vertical sag

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 5: MECHANICAL DESIGN OF OVERHEAD LINES
8

```

```

9 // EXAMPLE : 5.3 :
10 // Page number 198–199
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 a = 2.5                  // Cross-sectional area (sq.cm)
15 L = 250.0                 // Span(m)
16 w_c = 1.8                 // Weight of conductor (kg/m)
17 u = 8000.0                // Ultimate strength (kg/cm^2)
18 wind = 40.0                // Wind load (kg/cm^2)
19 s = 3.0                   // Factor of safety
20
21 // Calculations
22 d = (4.0*a/%pi)**0.5      // Diameter (cm)
23 T = u*a/s                  // Allowable max
    tension (kg)
24 w_w = wind*d/100.0         // Horizontal wind
    force (kg)
25 w_r = (w_c**2+w_w**2)**0.5 // Resultant force (kg
    /m)
26 S = w_r*L**2/(8*T)        // Slant sag (m)
27 vertical_sag = S*(w_c/w_r) // Vertical sag (m)
28
29 // Results
30 disp("PART II – EXAMPLE : 5.3 : SOLUTION :-")
31 printf("\nVertical sag = %.3f metres", vertical_sag)

```

---

#### Scilab code Exa 12.4 Height above ground at which the conductors should be supported

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5

```

```

6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 5: MECHANICAL DESIGN OF OVERHEAD LINES
8
9 // EXAMPLE : 5.4 :
10 // Page number 199
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 a = 110.0           // Cross-sectional area (sq .
    mm)
15 w_c = 844.0/1000   // Weight of conductor (kg/m)
16 U = 7950.0          // Ultimate strength (kg)
17 L = 300.0            // Span (m)
18 s = 2.0              // Factor of safety
19 wind = 75.0          // Wind pressure (kg/m^2)
20 h = 7.0               // Ground clearance (m)
21 d = 2.79             // Diameter of copper (mm)
22 n = 7.0               // Number of strands
23
24 // Calculations
25 dia = n*d           // Diameter of
    conductor (mm)
26 w_w = wind*dia/1000.0 // Horizontal wind
    force (kg)
27 w = (w_c**2+w_w**2)**0.5 // Resultant force (
    kg)
28 T = U/2.0             // Allowable
    tension (m)
29 l = L/2.0              // Half-span (m)
30 D = w*l**2/(2*T)       // Distance (m)
31 height = h+D           // Height above
    ground at which the conductors should be
    supported (m)
32
33 // Results
34 disp("PART II - EXAMPLE : 5.4 : SOLUTION :-")
35 printf("\nHeight above ground at which the

```

conductors should be supported = %.2 f metres" ,  
height)

---

### Scilab code Exa 12.5 Permissible span between two supports

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 5: MECHANICAL DESIGN OF OVERHEAD LINES
8
9 // EXAMPLE : 5.5 :
10 // Page number 199
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 w_w = 1.781           // Wind pressure on conductor (
    kg/m)
15 w_i = 1.08            // Weight of ice on conductor (
    kg/m)
16 D = 6.0               // Maximum permissible sag (m)
17 s = 2.0               // Factor of safety
18 w_c = 0.844           // Weight of conductor (kg/m)
19 u = 7950.0             // Ultimate strength (kg)
20
21 // Calculations
22 w = ((w_c+w_i)**2+w_w**2)**0.5          // Total force
    on conductor (kg/m)
23 T = u/s                         // Allowable
    maximum tension (kg)
24 l = ((D*2*T)/w)**0.5           // Half span (m)
25 L = 2.0*l                        // Permissible
```

```

        span between two supports (m)
26
27 // Results
28 disp("PART II - EXAMPLE : 5.5 : SOLUTION :-")
29 printf("\nPermissible span between two supports = %.
          f metres \n", L)
30 printf("\nNOTE: ERROR: Horizontal wind load, w_w =
          1.781 kg/m, not 1.78 kg/m as mentioned in problem
          statement")

```

---

### Scilab code Exa 12.6 Maximum sag of line due to weight of conductor Additional wei

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 5: MECHANICAL DESIGN OF OVERHEAD LINES
8
9 // EXAMPLE : 5.6 :
10 // Page number 199–200
11 clear ; clc ; close ; // Clear the work space and
                           console
12
13 // Given data
14 a = 0.484           // Area of conductor (sq.cm)
15 d = 0.889           // Overall diameter (cm)
16 w_c = 428/1000.0   // Weight(kg/m)
17 u = 1973.0          // Breaking strength (kg)
18 s = 2.0             // Factor of safety
19 L = 200.0            // Span(m)
20 t = 1.0              // Ice thickness (cm)
21 wind = 39.0          // Wind pressure (kg/m^2)
22

```

```

23 // Calculations
24 // Case(i)
25 l = L/2.0
      Half span(m)                                //
26 T = u/s
      Allowable maximum tension(kg)               //
27 D_1 = w_c*l**2/(2*T)
      Maximum sag due to weight of conductor(m)  //
28 // Case(ii)
29 w_i = 913.5*%pi*t*(d+t)*10**-4
      Weight of ice on conductor(kg/m)           //
30 w = w_c+w_i
      Total weight of conductor & ice(kg/m)     //
31 D_2 = w*l**2/(2*T)
      Maximum sag due to additional weight of ice(m)//
32 // Case(iii)
33 D = d+2.0*t
      Diameter due to ice(cm)                   //
34 w_w = wind*D*10**-2
      Wind pressure on conductor(kg/m)          //
35 w_3 = ((w_c+w_i)**2+w_w**2)**0.5
      Total force on conductor(kg/m)            //
36 D_3 = w_3*l**2/(2*T)
      Maximum sag due to (i), (ii) & wind(m)   //
37 theta = atand(w_w/(w_c+w_i))
      ( )
38 vertical_sag = D_3*cosd(theta)
      Vertical sag(m)                          //
39
40 // Results
41 disp("PART II - EXAMPLE : 5.6 : SOLUTION :-")
42 printf("\nCase(i) : Maximum sag of line due to
      weight of conductor , D = %.2f metres" , D_1)
43 printf("\nCase(ii) : Maximum sag of line due to
      additional weight of ice , D = %.2f metres" , D_2)
44 printf("\nCase(iii) : Maximum sag of line due to (i)
      ,(ii) plus wind , D = %.2f metres" , D_3)
45 printf("\n      Vertical sag = %.2f metres" ,

```

```
vertical_sag)
```

---

### Scilab code Exa 12.7 Point of minimum sag

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 5: MECHANICAL DESIGN OF OVERHEAD LINES
8
9 // EXAMPLE : 5.7 :
10 // Page number 200
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 W = 428/1000.0          // Weight(kg/m)
15 u = 1973.0              // Breaking strength(kg)
16 s = 2.0                 // Factor of safety
17 l = 200.0               // Span(m)
18 h = 3.0                 // Difference in tower height(m)
19
20 // Calculations
21 T = u/s                  // Allowable
    maximum tension(kg)
22 x_2 = (1/2.0)+(T*h/(W*l))      // Point of
    minimum sag from tower at higher level(m)
23 x_1 = l-x_2                // Point of
    minimum sag from tower at lower level(m)
24
25 // Results
26 disp("PART II - EXAMPLE : 5.7 : SOLUTION :-")
27 printf("\nPoint of minimum sag , x_1 = %.1f metres" ,
```

```

        x_1)
28 printf("\nPoint of minimum sag , x_2 = %.1f metres" ,
        x_2)

```

---

**Scilab code Exa 12.8 Clearance between conductor and water at a point midway between two towers**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 5: MECHANICAL DESIGN OF OVERHEAD LINES
8
9 // EXAMPLE : 5.8 :
10 // Page number 200–201
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 h_1 = 50.0           // Height of tower P1(m)
15 h_2 = 80.0           // Height of tower P2(m)
16 L = 300.0           // Horizontal distance b/w
    towers(m)
17 T = 2000.0          // Tension in conductor(kg)
18 w = 0.844            // Weight of conductor(kg/m)
19
20 // Calculations
21 h = h_2-h_1          // Difference
    in height of tower(m)
22 x_2 = (L/2.0)+(T*h/(w*L))          // Point of
    minimum sag from tower P2(m)
23 x_1 = (L/2.0)-(T*h/(w*L))          // Point of
    minimum sag from tower at lower level(m)
24 P = (L/2.0)-x_1                  // Distance of

```

```

    point P(m)
25 D = w*P**2/(2*T)                                // Height of P
    above O(m)
26 D_2 = w*x_2**2/(2*T)                            // Height of
    P2 above O(m)
27 mid_point_P2 = D_2-D                           // Mid-point
    below P2(m)
28 clearance = h_2-mid_point_P2                   // Clearance b
    /w conductor & water(m)
29 D_1 = w*x_1**2/(2*T)                            // Height of
    P1 above O(m)
30 mid_point_P1 = D-D_1                           // Mid-point
    above P1(m)
31 clearance_alt = h_1+mid_point_P1             // Clearance b
    /w conductor & water(m)
32
33 // Results
34 disp("PART II - EXAMPLE : 5.8 : SOLUTION :-")
35 printf("\nClearance between conductor & water at a
        point midway b/w towers = %.2f m above water\n",
        clearance)
36 printf("\nALTERNATIVE METHOD:")
37 printf("\nClearance between conductor & water at a
        point midway b/w towers = %.2f m above water",
        clearance_alt)

```

---

### Scilab code Exa 12.9 Sag at erection and Tension of the line

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 5: MECHANICAL DESIGN OF OVERHEAD LINES

```

```

8
9 // EXAMPLE : 5.9 :
10 // Page number 201
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 L = 300.0          // Span(m)
15 T_still = 45.0      // Temperature in still air( C
    )
16 a = 226.0          // Area(mm^2)
17 d = 19.53/10        // Overall diameter(cm)
18 w_2 = 0.844         // Weight of conductor(kg/m)
19 u = 7950.0          // Ultimate strength(kg)
20 alpha = 18.44*10**-6 // Co-efficient of linear
    expression(/ C )
21 E = 9.32*10**3       // Modulus of elasticity(kg/mm
    ^2)
22 t = 0.95            // Ice thickness(cm)
23 wind = 39.0           // Wind pressure(kg/m^2)
24 T_worst = -5.0        // Temperature in worst
    condition( C )
25
26 // Calculations
27 w_i = 915.0*pi*t*(d+t)*10**-4           // Weight of
    ice on conductor(kg/m)
28 w_w = wind*(d+2*t)*10**-2                 // Wind load
    of conductor(kg/m)
29 w_1 = ((w_2+w_i)**2+w_w**2)**0.5          // Total
    force on conductor(kg/m)
30 t = T_still-T_worst                      //
    Temperature( C )
31 l = L/2.0                                // Half span
    (m)
32 T = u/2.0                                 // Allowable
    tension(kg)
33 A = 1.0                                    // Co-
    efficient of x^3

```

```

34 B = a*E*(alpha*t+((w_1*l/T)**2/6))-T // Co-
    efficient of x^2
35 C = 0 // Co-
    efficient of x
36 D = -(w_2**2*l**2*a*E/6) // Co-
    efficient of constant
37 T_2_sol = roots([A,B,C,D]) // Roots of
    tension of a line
38 T_2_s = T_2_sol(3) // Feasible
    solution of tension of
39 T_2 = 1710.0 // Tension
    in conductor(kg). Obtained directly from textbook
40 sag = w_2*l**2/(2*T_2) // Sag at
    erection(m)
41
42 // Results
43 disp("PART II - EXAMPLE : 5.9 : SOLUTION :-")
44 printf("\nSag at erection = %.2f metres", sag)
45 printf("\nTension of the line , T_2 = %.f kg (An app.
    solution as per calculation) = %.f kg (More
    correctly as standard value)", T_2_s,T_2)

```

---

### Scilab code Exa 12.10 Sag in inclined direction and Vertical direction

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 5: MECHANICAL DESIGN OF OVERHEAD LINES
8
9 // EXAMPLE : 5.10 :
10 // Page number 201–202
11 clear ; clc ; close ; // Clear the work space and

```

```

        console
12
13 // Given data
14 L = 250.0          // Span (m)
15 d = 1.42           // Diameter (cm)
16 w = 1.09           // Dead weight (kg/m)
17 wind = 37.8         // Wind pressure (kg/m^2)
18 r = 1.25           // Ice thickness (cm)
19 f_m = 1050.0       // Maximum working stress (kg/sq.
                      cm)

20
21 // Calculations
22 w_i = 913.5 * %pi * r * (d+r) * 10**-4           // Weight of
              ice on conductor (kg/m)
23 w_w = wind * (d+2*r) * 10**-2                     // Wind load
              of conductor (kg/m)
24 w_r = ((w+w_i)**2 + w_w**2)**0.5                 // Resultant
              pressure (kg/m)
25 a = %pi * d**2 / 4.0                                // Area (cm
              ^2)
26 T_0 = f_m * a                                       // Tension (
              kg)
27 S = w_r * L**2 / (8 * T_0)                         // Total sag
              (m)
28 vertical_sag = S * (w+w_i) / w_r                  // Vertical
              component of sag (m)
29
30 // Results
31 disp("PART II - EXAMPLE : 5.10 : SOLUTION :-")
32 printf("\nCase(i) : Sag in inclined direction = %.f
          m", S)
33 printf("\nCase(ii) : Sag in vertical direction = %.2f
          m", vertical_sag)

```

---

**Scilab code Exa 12.11 Sag in still air Wind pressure Ice coating and Vertical sag**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 5: MECHANICAL DESIGN OF OVERHEAD LINES
8
9 // EXAMPLE : 5.11 :
10 // Page number 202–203
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 a = 120.0           // Area(mm^2)
15 ds = 2.11          // Diameter of each strand
   (mm)
16 W = 1118.0/1000    // Weight of conductor(kg/
   m)
17 L = 200.0           // Span(m)
18 stress = 42.2       // Ultimate tensile stress
   (kg/mm^2)
19 wind = 60.0          // Wind pressure(kg/m^2)
20 t = 10.0            // Ice thickness(mm)
21
22 // Calculations
23 n = 3.0             // Number of layers
24 d = (2*n+1)*ds      // Overall diameter of conductor(mm)
25 u = stress*a         // Ultimate strength(kg)
26 T = u/4.0             // Working stregth (kg)
27 // Case(a)
28 S_a = W*L**2/(8*T)    // Sag in still air(m)
29 // Case(b)

```

```

30 area = d*100*10.0*10**-6 //  

   Projected area to wind pressure(m^2)  

31 w_w = wind*area //  

   Wind load/m(kg)  

32 w_r = (W**2+w_w**2)**0.5 //  

   Resultant weight/m(kg)  

33 S_b = w_r*L**2/(8*T) //  

   Total sag with wind pressure(m)  

34 w_i = 0.915*pi/4*((d+2*t)**2-(d**2))/1000.0 //  

   Weight of ice on conductor(kg/m)  

35 area_i = (d+2*t)*1000.0*10**-6 //  

   Projected area to wind pressure(m^2)  

36 w_n = wind*area_i //  

   Wind load/m(kg)  

37 w_r_c = ((W+w_i)**2+w_n**2)**0.5 //  

   Resultant weight/m(kg)  

38 S_c = w_r_c*L**2/(8*T) //  

   Total sag with wind pressure and ice coating(m)  

39 S_v = S_c*(W+w_i)/w_r_c //  

   Vertical component of sag(m)  

40  

41 // Results  

42 disp("PART II - EXAMPLE : 5.11 : SOLUTION :-")  

43 printf("\nCase(a) : Sag in still air , S = %.2f m ,  

   S_a)  

44 printf("\nCase(b) : Sag with wind pressure , S = %.2f  

   m" , S_b)  

45 printf("\n           Sag with wind pressure and ice  

   coating , S = %.2f m" , S_c)  

46 printf("\n           Vertical sag , S_v = %.2f m \n" ,  

   S_v)  

47 printf("\nNOTE: ERROR: calculation mistake in the  

   textbook")

```

---

# Chapter 13

## INTERFERENCE OF POWER LINES WITH NEIGHBOURING COMMUNICATION CIRCUITS

Scilab code Exa 13.1 Mutual inductance between the circuits and Voltage induced in

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 6: INTERFERENCE OF POWER LINES WITH
    NEIGHBOURING COMMUNICATION CIRCUITS
8
9 // EXAMPLE : 6.1 :
10 // Page number 206
11 clear ; clc ; close ; // Clear the work space and
    console
```

```

12
13 // Given data
14 f = 50.0 // Frequency (Hz)
15 d = 4.0 // Spacing b/w conductors (m)
16 D = 2.0 // Distance of telephone line
17 s = 60.0/100 // Spacing b/w telephone line (m)
18 r = 2.0 // Radius of power line (mm)
19 I = 150.0 // Current in power line (A)
20
21 // Calculations
22 D_ac = (D**2+((d-s)/2)**2)**0.5 // Distance b/w a & c (m)
23 D_ad = (D**2+((d-s)/2)+s)**2)**0.5 // Distance b/w a & d (m)
24 M = 4.0*10**-7*log(D_ad/D_ac)*1000 // Mutual inductance b/w circuits (H/km)
25 V_CD = 2.0*%pi*f*M*I // Voltage induced in the telephone line (V/km)
26
27 // Results
28 disp("PART II - EXAMPLE : 6.1 : SOLUTION :-")
29 printf("\nMutual inductance between the circuits , M
      = %.e H/km" , M)
30 printf("\nVoltage induced in the telephone line ,
      V_CD = %.2f V/km" , V_CD)

```

---

### Scilab code Exa 13.2 Induced voltage at fundamental frequency and Potential of tel

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5

```

```

6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 6: INTERFERENCE OF POWER LINES WITH
    NEIGHBOURING COMMUNICATION CIRCUITS
8
9 // EXAMPLE : 6.2 :
10 // Page number 206–207
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 f = 50.0          // Frequency (Hz)
15 l = 160.0         // Length of line (km)
16 V = 132.0*10**3  // Line voltage (V)
17 P = 25.0*10**6   // Load delivered (W)
18 PF = 0.8          // Lagging power factor
19 r = 5.0/1000     // Radius of power line
    conductor (m)
20 d = 4.0           // Spacing b/w conductors (m)
21 OS = 6.0          // Distance (m)
22 OT = 6.5          // Distance (m)
23 CT = 18.0         // Distance (m)
24
25 // Calculations
26 A0 = 3**0.5*d/2.0
    // Distance A to O(m). From figure E6.2
27 AS = OS+A0
    // Distance A to S(m)
28 AT = A0+OT
    // Distance A to T(m)
29 OB = d/2.0
    // Distance O to B(m)
30 BS = (OB**2+OS**2)**0.5
    // Distance
    B to S (m)

```

```

31 BT = (OB**2+OT**2)**0.5 // Distance
      B to T(m)
32 M_A = 0.2*log(AT/AS) //
      Mutual inductance at A(mH/km)
33 M_B = 0.2*log(BT/BS) //
      Mutual inductance at B(mH/km)
34 M = M_B-M_A //
      Mutual inductance at C(mH/km)
35 I = P/(3**0.5*V*PF) //
      Current (A)
36 E_m = 2.0*%pi*f*M*I*10**-3*l //
      Induced voltage (V)
37 V_A = V/3**0.5 //
      Phase voltage (V)
38 h = A0+CT //
      Height (m)
39 V_SA = V_A*log10(((2*h)-AS)/AS)/log10(((2*h)-r)/r) //
      Potential (V)
40 H = CT //
      Height (m)
41 V_B = V_A //
      Phase voltage (V)
42 V_SB = V_B*log10(((2*H)-BS)/BS)/log10(((2*H)-r)/r) //
      Potential (V)
43 V_S = V_SB-V_SA //
      Total potential of S w.r.t earth (V)
44

```

```
45 // Results
46 disp("PART II - EXAMPLE : 6.2 : SOLUTION :-")
47 printf("\nInduced voltage at fundamental frequency ,
        E_m = %.1f V", E_m)
48 printf("\nPotential of telephone conductor S above
        earth , V_S = %.f V \n", V_S)
49 printf("\nNOTE: ERROR: Changes in obtained answer is
        due to precision and calculation mistakes in
        textbook")
```

---

# Chapter 14

## UNDERGROUND CABLES

Scilab code Exa 14.1 Insulation resistance per km

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 7: UNDERGROUND CABLES
8
9 // EXAMPLE : 7.1 :
10 // Page number 211
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 d = 2.5          // Core diameter(cm)
15 t = 1.25         // Insulation thickness(cm)
16 rho = 4.5*10**14 // Resistivity of insulation(
    ohm-cm)
17 l = 10.0**5      // Length(cm)
18
19 // Calculations
```

```

20 D = d+2*t                                // Overall diameter
      (cm)
21 R_i = rho/(2*pi*l)*log(D/d)           // Insulation
      resistance (ohm)
22
23 // Results
24 disp("PART II - EXAMPLE : 7.1 : SOLUTION :-")
25 printf("\nInsulation resistance per km, R_i = %.2e
      ohm\n", R_i)
26 printf("\nNOTE: ERROR: Mistake in final answer in
      textbook")

```

---

### Scilab code Exa 14.2 Insulation thickness

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 7: UNDERGROUND CABLES
8
9 // EXAMPLE : 7.2 :
10 // Page number 211
11 clear ; clc ; close ; // Clear the work space and
      console
12
13 // Given data
14 R = 495.0*10**6          // Insulation resistance (ohm/km
      )
15 d = 3.0                  // Core diameter (cm)
16 rho = 4.5*10**14         // Resistivity of insulation (
      ohm-cm)
17
18 // Calculations

```

```

19 l = 1000.0 // Length
    of cable(m)
20 r_2 = d/2.0 // Core
    radius(cm)
21 Rho = rho/100.0 // Resistivity of insulation(ohm-m)
22 r1_r2 = exp((2*pi*l*R)/Rho) // r1/r2
23 r_1 = 2*r_2 // Cable
    radius(cm)
24 thick = r_1-r_2 // Insulation thickness(cm)
25
26 // Results
27 disp("PART II - EXAMPLE : 7.2 : SOLUTION :-")
28 printf("\nInsulation thickness = %.1f cm", thick)

```

---

### Scilab code Exa 14.3 Capacitance and Charging current of single core cable

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 7: UNDERGROUND CABLES
8
9 // EXAMPLE : 7.3 :
10 // Page number 212
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 66.0*10**3 // Line Voltage(V)
15 l = 1.0 // Length of cable(km)
16 d = 15.0 // Core diameter(cm)

```

```

17 D = 60.0          // Sheath diameter(cm)
18 e_r = 3.6         // Relative permittivity
19 f = 50.0          // Frequency(Hz)
20
21 // Calculations
22 C = e_r/(18.0*log(D/d))*1           // Capacitance(
    F )
23 I_ch = V/3**0.5*2*pi*f*C*10**-6   // Charging
    current(A)
24
25 // Results
26 disp("PART II - EXAMPLE : 7.3 : SOLUTION :-")
27 printf("\nCapacitance of single-core cable , C = %.3f
    F ", C)
28 printf("\nCharging current of single-core cable = %
    .2f A", I_ch)

```

---

#### Scilab code Exa 14.4 Most economical diameter of a single core cable and Overall d

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 7: UNDERGROUND CABLES
8
9 // EXAMPLE : 7.4 :
10 // Page number 212
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V_l = 132.0           // Line Voltage(kV)
15 g_max = 60.0          // Maximum Line Voltage(kV)

```

```

16
17 // Calculations
18 V = V_1/3**0.5*2**0.5      // Phase Voltage (kV)
19 d = 2*V/g_max              // Core diameter (cm)
20 D = 2.718*d                // Overall diameter (cm)
21
22 // Results
23 disp("PART II - EXAMPLE : 7.4 : SOLUTION :-")
24 printf("\nMost economical diameter of a single-core
cable , d = %.1f cm" , d)
25 printf("\nOverall diameter of the insulation , D = %
.3f cm\n" , D)
26 printf("\nNOTE: Slight change in obtained answer due
to precision")

```

---

### Scilab code Exa 14.6 Conductor radius and Electric field strength that must be with

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 7: UNDERGROUND CABLES
8
9 // EXAMPLE : 7.6 :
10 // Page number 212-213
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 V = 11.0*10**3           // Line Voltage (V)
15 dia_out = 8.0             // Outside diameter (cm)
16
17 // Calculations

```

```

18 D = dia_out/2.0 // Overall
    diameter (cm)
19 d = (D)/2.718 // Conductor
    diameter (cm)
20 r = d/2 // Conductor
    radius (cm)
21 g_m = 2*V/(d*log(D/d)*10) // Maximum
    value of electric field strength (kV/m)
22
23 // Results
24 disp("PART II - EXAMPLE : 7.6 : SOLUTION :-")
25 printf("\nConductor radius , r = %.3f cm" , r)
26 printf("\nElectric field strength that must be
    withstood , g_m = %.f kV/m" , g_m)

```

---

### Scilab code Exa 14.7 Location of intersheath and Ratio of maximum electric s

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 7: UNDERGROUND CABLES
8
9 // EXAMPLE : 7.7 :
10 // Page number 214
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 R_3 = 1.00 // Cable radius (cm)
15 R_1 = 2.5 // Cable radius (cm)
16
17 // Calculations

```

```

18 R_2 = (R_1*R_3)**0.5           // Location of intersheath
   (cm)
19 alpha = R_1/R_2                //
20 ratio = 2.0/(1+alpha)          // Ratio of maximum
   electric field strength with & without
   intersheath
21
22 // Results
23 disp("PART II - EXAMPLE : 7.7 : SOLUTION :-")
24 printf("\nLocation of intersheath , R_2 = %.2f cm" ,
   R_2)
25 printf("\nRatio of maximum electric field strength
   with & without intersheath = %.3f " , ratio)

```

---

#### Scilab code Exa 14.8 Maximum and Minimum stress in the insulation

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 7: UNDERGROUND CABLES
8
9 // EXAMPLE : 7.8 :
10 // Page number 215
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 V = 33.0                  // Line Voltage(kV)
15 D_2 = 2.0                  // Conductor diameter(cm)
16 D_1 = 3.0                  // Sheath diameter(cm)
17
18 // Calculations

```

```

19 R_2 = D_2/2 // Conductor
    radius (cm)
20 R_1 = D_1/2 // Sheath radius
    (cm)
21 g_max = V/(R_2*log(R_1/R_2)) // RMS value of
    maximum stress in the insulation (kV/cm)
22 g_min = V/(R_1*log(R_1/R_2)) // RMS value of
    minimum stress in the insulation (kV/cm)
23
24 // Results
25 disp("PART II - EXAMPLE : 7.8 : SOLUTION :-")
26 printf("\nMaximum stress in the insulation , g_max = "
    %.2f kV/cm (rms)" , g_max)
27 printf("\nMinimum stress in the insulation , g_min = "
    %.2f kV/cm (rms)" , g_min)

```

---

### Scilab code Exa 14.9 Maximum stress with and without intersheath Best position and

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 7: UNDERGROUND CABLES
8
9 // EXAMPLE : 7.9 :
10 // Page number 215
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 d = 2.5 // Conductor diameter(cm)
15 D = 6.0 // Sheath diameter(cm)
16 V_l = 66.0 // Line Voltage(kV)

```

```

17
18 // Calculations
19 alpha = (D/d)**(1.0/3) // Best
20 d_1 = d*alpha // Best
    position of first intersheath (cm)
21 d_2 = d_1*alpha // Best
    position of second intersheath (cm)
22 V = V_1/3**0.5*2**0.5 // Peak voltage
    on core (kV)
23 V_2 = V/(1+(1/alpha)+(1/alpha**2)) // Peak voltage
    on second intersheath (kV)
24 V_1 = (1+(1/alpha))*V_2 // Voltage on
    first intersheath (kV)
25 stress_max = 2*V/(d*log(D/d)) // Maximum
    stress without intersheath (kV/cm)
26 stress_min = stress_max*d/D // Minimum
    stress without intersheath (kV/cm)
27 g_max = V*3/(1+alpha+alpha**2) // Maximum
    stress with intersheath (kV/cm)

28
29 // Results
30 disp("PART II - EXAMPLE : 7.9 : SOLUTION :-")
31 printf("\nMaximum stress without intersheath = %.2f
    kV/cm", stress_max)
32 printf("\nBest position of first intersheath , d_1 =
    %.2f cm", d_1)
33 printf("\nBest position of second intersheath , d_2 =
    %.3f cm", d_2)
34 printf("\nMaximum stress with intersheath = %.2f kV/
    cm", g_max)
35 printf("\nVoltage on the first intersheath , V_1 = %
    .2f kV", V_1)
36 printf("\nVoltage on the second intersheath , V_2 = %
    .2f kV \n", V_2)
37 printf("\nNOTE: Changes in the obtained answer is
    due to more precision here")

```

---

### Scilab code Exa 14.10 Maximum stress in the two dielectrics

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 7: UNDERGROUND CABLES
8
9 // EXAMPLE : 7.10 :
10 // Page number 215–216
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 e_1 = 3.6          // Inner relative permittivity
15 e_2 = 2.5          // Outer relative permittivity
16 d = 1.0           // Conductor diameter(cm)
17 d_1 = 3.0          // Sheath diameter(cm)
18 D = 5.0           // Overall diameter(cm)
19 V_1 = 66.0         // Line Voltage(kV)
20
21 // Calculations
22 V = V_1/3**0.5*2**0.5
                           // Peak voltage on
                           core(kV)
23 g1_max = 2*V/(d*(log(d_1/d)+e_1/e_2*log(D/d_1)))
                           // Maximum stress in first dielectric(kV/km)
24 g_max = 2*V/(d_1*(e_2/e_1*log(d_1/d)+log(D/d_1)))
                           // Maximum stress in second dielectric(kV/km)
25
26 // Results
27 disp("PART II – EXAMPLE : 7.10 : SOLUTION :-")
```

```

28 printf("\nMaximum stress in first dielectric ,
29 g_1_max = %.2f kV/cm", g1_max)
30 printf("\nMaximum stress in second dielectric , g_max
31 = %.2f kV/cm", g_max)

```

---

### Scilab code Exa 14.11 Diameter and Voltage of intersheath Conductor and Outside di

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 7: UNDERGROUND CABLES
8
9 // EXAMPLE : 7.11 :
10 // Page number 216–217
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 85.0           // Line Voltage(kV)
15 g_max = 55.0       // Maximum stress (kV/cm)
16
17 // Calculations
18 V_1 = 0.632*V      // Intersheath potential(kV)
19 d = 0.736*V/g_max // Core diameter(cm)
20 d_1 = 2*V/g_max   // Intersheath diameter(cm)
21 D = 3.76*V/g_max // Overall diameter(cm)
22 d_un = 2*V/g_max // Core diameter of ungraded
    cable(cm)
23 D_un = 2.718*d_1 // Overall diameter of
    ungraded cable(cm)
24
25 // Results

```

```

26 disp("PART II – EXAMPLE : 7.11 : SOLUTION :-")
27 printf("\nDiameter of intersheath , d_1 = %.2f cm" ,
d_1)
28 printf("\nVoltage of intersheath , V_1 = %.2f kV, to
neutral" , V_1)
29 printf("\nConductor diameter of graded cable , d = %
.2f cm" , d)
30 printf("\nOutside diameter of graded cable , D = %.2f
cm" , D)
31 printf("\nConductor diameter of ungraded cable , d = %
.2f cm" , d_un)
32 printf("\nOutside diameter of ungraded cable , D = %
.2f cm" , D_un)

```

---

**Scilab code Exa 14.12 Equivalent star connected capacity and kVA required**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 7: UNDERGROUND CABLES
8
9 // EXAMPLE : 7.12 :
10 // Page number 219
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 c = 0.3           // Capacitance b/w any 2 conductor &
sheath earthed( F /km)
15 l = 10.0          // Length(km)
16 V = 33.0          // Line Voltage(kV)
17 f = 50.0          // Frequency(Hz)

```

```

18
19 // Calculations
20 C_eq = l*c // Capacitance
   b/w any 2 conductor & sheath earthed( F )
21 C_p = 2.0*C_eq // Capacitance
   per phase( F )
22 kVA = V**2*2*pi*f*C_p/1000.0 // Three-phase
   kVA required (kVA)
23
24 // Results
25 disp("PART II - EXAMPLE : 7.12 : SOLUTION :-")
26 printf("\nEquivalent star connected capacity , C_eq =
   %.f F ", C_eq)
27 printf("\nkVA required = %.1f kVA", kVA)

```

---

### Scilab code Exa 14.13 Charging current drawn by a cable with three cores

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 7: UNDERGROUND CABLES
8
9 // EXAMPLE : 7.13 :
10 // Page number 219
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 V = 11.0*10**3 // Line Voltage(V)
15 f = 50.0 // Frequency(Hz)
16 C_c = 3.7 // Measured capacitance( F )
17

```

```

18 // Calculations
19 C_0 = 2*C_c // Capacitance( F )
20 I_ch = 2*pi*f*C_0*V/3**0.5*10**-6 // Charging current per phase(A)
21
22 // Results
23 disp("PART II - EXAMPLE : 7.13 : SOLUTION :-")
24 printf("\nCharging current drawn by a cable = %.2f A
", I_ch)

```

---

#### Scilab code Exa 14.14 Capacitance between any two conductors Two bounded conductor

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 7: UNDERGROUND CABLES
8
9 // EXAMPLE : 7.14 :
10 // Page number 219–220
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 c_s = 0.90 // Capacitance b/w all conductors(
    F )
15 C_0 = 0.4 // Capacitance b/w two conductor(
    F )
16 V = 11.0*10**3 // Line Voltage(V)
17 f = 50.0 // Frequency(Hz)
18
19 // Calculations

```

```

20 C_s = c_s/3.0 // Capacitance measured( F )
21 C_c = (C_0-C_s)/2.0 // Capacitance( F )
22 C_a = 3.0/2*(C_c+(1/3.0)*C_s) // Capacitance b/w any two conductors( F )
23 C_b = 2.0*C_c+(2.0/3)*C_s // Capacitance b/w any two bounded conductors and
                                the third conductor( F )
24 C_o = 3.0*C_c+C_s // Capacitance to neutral( F )
25 I_c = 2.0*%pi*f*C_o*V/3**0.5*10**-6 // Charging current(A)
26
27 // Results
28 disp("PART II – EXAMPLE : 7.14 : SOLUTION :-")
29 printf("\nCase(a): Capacitance between any two
           conductors = %.3f F ", C_a)
30 printf("\nCase(b): Capacitance between any two
           bounded conductors and the third conductor = %.1f
           F ", C_b)
31 printf("\nCase(c): Capacitance to neutral , C_0 = %.2
           f F ", C_o)
32 printf("\n          Charging current taken by cable ,
           I_c = %.3f A \n", I_c)
33 printf("\nNOTE: ERROR: Calculation mistakes in
           textbook answer")

```

---

### Scilab code Exa 14.15 Charging current drawn by cable

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5

```

```

6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 7: UNDERGROUND CABLES
8
9 // EXAMPLE : 7.15 :
10 // Page number 220–221
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 13.2*10***3      // Line Voltage(V)
15 f = 50.0              // Frequency(Hz)
16 C_BC = 4.2            // Capacitance b/w two cores( F )
17
18 // Calculations
19 C_n = 2.0*C_BC        //
    Capacitance to neutral( F )
20 V_ph = V/3***0.5       //
    Operating phase voltage(V)
21 I_c = 2.0*%pi*f*C_n*V/3***0.5*10***-6   //
    Charging current(A)
22
23 // Results
24 disp("PART II – EXAMPLE : 7.15 : SOLUTION :-")
25 printf("\nCharging current drawn by cable , I_c = %.2
    f A" , I_c)

```

---

### Scilab code Exa 14.16 Capacitance of the cable Charging current Total charging kVA

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 7: UNDERGROUND CABLES

```

```

8
9 // EXAMPLE : 7.16 :
10 // Page number 222–223
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 33.0*10**3      // Line Voltage(V)
15 f = 50.0              // Frequency(Hz)
16 l = 4.0                // Length(km)
17 d = 2.5                // Diameter of conductor(cm)
18 t = 0.5                // Radial thickness of insulation(
    cm)
19 e_r = 3.0              // Relative permittivity of the
    dielectric
20 PF = 0.02            // Power factor of unloaded cable
21
22 // Calculations
23 // Case(a)
24 r = d/2.0            // Radius of conductor(cm)
25 R = r+t              // External radius(cm)
26 e_0 = 8.85*10**-12        // Permittivity
27 C = 2.0*%pi*e_0*e_r*log(R/r)*l*1000        // Capacitance of cable/phase(F)
28 // Case(b)
29 V_ph = V/3**0.5          // Phase voltage(V)
30 I_c = V_ph*2.0*%pi*f*C        // Charging current/phase(A)
31 // Case(c)
32 kVAR = 3.0*V_ph*I_c          // Total charging kVAR
33 // Case(d)
34 phi = acosd(PF)            // ( )

```

```

35 delta = 90.0-phi                                // 
( )
36 P_c = V_ph*I_c*sind(delta)/1000                // 
    Dielectric loss/phase (kW)
37 // Case(e)
38 E_max = V_ph/(r*log(R/r)*1000)                // 
    RMS value of Maximum stress in cable (kV/cm)
39
40 // Results
41 disp("PART II - EXAMPLE : 7.16 : SOLUTION :-")
42 printf("\nCase(a): Capacitance of the cable , C = %.3
e F/phase" , C)
43 printf("\nCase(b): Charging current = %.2f A/phase" ,
I_c)
44 printf("\nCase(c): Total charging kVAR = %.4e kVAR" ,
kVAR)
45 printf("\nCase(d): Dielectric loss/phase , P_c = %.2f
kW" , P_c)
46 printf("\nCase(e): Maximum stress in the cable ,
E_max = %.1f kV/cm (rms)" , E_max)

```

---

# Chapter 15

## CORONA

Scilab code Exa 15.1 Minimum spacing between conductors

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 8: CORONA
8
9 // EXAMPLE : 8.1 :
10 // Page number 227
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 d = 30.0/10          // Diameter of conductor (cm)
15 delta = 0.95         // Air density factor
16 m = 0.95             // Irregularity factor
17 E = 230.0             // Line voltage (kV)
18 g_0 = 30.0/2**0.5    // Breakdown strength of air (kV
    /cm)
19
```

```

20 // Calculations
21 E_0 = E/3**0.5 // Disruptive critical voltage(kV)
22 r = d/2.0 // Radius of conductor(cm)
23 D = exp(E_0/(m*delta*g_0*r))*r/100 // Minimum spacing between conductors(m)
24
25 // Results
26 disp("PART II - EXAMPLE : 8.1 : SOLUTION :-")
27 printf("\nMinimum spacing between conductors , D = % .3f m \n", abs(D))
28 printf("\nNOTE: Changes in obtained answer from that of textbook due to precision")

```

---

### Scilab code Exa 15.2 Critical disruptive voltage and Corona loss

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 8: CORONA
8
9 // EXAMPLE : 8.2 :
10 // Page number 227–228
11 clear ; clc ; close ; // Clear the work space and console
12
13 // Given data
14 V = 220.0 // Operating line voltage(kV)
15 f = 50.0 // Frequency(Hz)
16 d = 1.5 // Diameter of conductor(cm)
17 D = 300.0 // Distance b/w conductor(cm)

```

```

18 delta = 1.05          // Air density factor
19 g_0 = 21.1             // Breakdown strength of air (kV
                           /cm)
20 m = 1.0                // Irregularity factor
21
22 // Calculations
23 E = V/3**0.5
                           //
                           Phase voltage (kV)
24 r = d/2.0
                           //
                           Radius of conductor (cm)
25 E_0 = m*g_0*delta*r*log(D/r)
                           // Disruptive critical
                           voltage to neutral (kV/phase)
26 E_0_ll = 3**0.5*E_0
                           // Line-to-
                           line Disruptive critical voltage (kV)
27 P = 244.0*10**-5*(f+25)/delta*(r/D)**0.5*(E-E_0)**2
                           // Corona loss (kW/km/phase)
28 P_total = P*3.0
                           // Corona
                           loss (kW/km)
29
30 // Results
31 disp("PART II - EXAMPLE : 8.2 : SOLUTION :-")
32 printf("\nCritical disruptive voltage , E_0 = %.2f kV
                           /phase = %.2f kV (line-to-line)" , E_0,E_0_ll)
33 printf("\nCorona loss , P = %.2f kW/km \n" , P_total)
34 printf("\nNOTE: ERROR: Calculation mistake in the
                           final answer in textbook")

```

---

### Scilab code Exa 15.3 Corona loss in fair weather and Foul weather

```
1 // A Texbook on POWER SYSTEM ENGINEERING
```

```

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 8: CORONA
8
9 // EXAMPLE : 8.3 :
10 // Page number 228
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 132.0          // Operating line voltage(kV)
15 f = 50.0           // Frequency(Hz)
16 d = 1.17            // Diameter of conductor(cm)
17 D = 300.0           // Distance b/w conductor(cm)
18 m = 0.96            // Irregularity factor
19 b = 72.0             // Barometric pressure(cm)
20 t = 20.0             // Temperature( C )
21
22 // Calculations
23 delta = 3.92*b/(273.0+t)                                // Air
    density factor
24 r = d/2.0

    // Radius of conductor(cm)
25 E_0 = 21.1*m*delta*r*log(D/r)                            // Critical
    disruptive voltage for fair weather condition(kV/
    phase)
26 E_0_foul = 0.8*E_0                                         //
    Critical disruptive voltage for foul weather(kV/
    phase)
27 E = V/3**0.5

```

```

    // Phase voltage(kV)
28 P_fair = 244.0*10**-5*(f+25)/delta*(r/D)**0.5*(E-E_0
    )**2           // Corona loss for fair weather
    condition(kW/km/phase)
29 P_foul = 244.0*10**-5*(f+25)/delta*(r/D)**0.5*(E-
    E_0_foul)**2      // Corona loss for foul weather
    condition(kW/km/phase)
30
31 // Results
32 disp("PART II - EXAMPLE : 8.3 : SOLUTION :-")
33 printf("\nCorona loss in fair weather , P = %.3f kW/
    km/phase" , P_fair)
34 printf("\nCorona loss in foul weather , P = %.3f kW/
    km/phase" , P_foul)

```

---

#### Scilab code Exa 15.4 Corona characteristics

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 8: CORONA
8
9 // EXAMPLE : 8.4 :
10 // Page number 228–229
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 110.0          // Operating line voltage(kV)
15 f = 50.0           // Frequency(Hz)
16 l = 175.0          // Line length(km)
17 d = 1.0            // Diameter of conductor(cm)

```

```

18 D = 300.0          // Distance b/w conductor(cm)
19 t = 26.0           // Temperature( C )
20 b = 74.0           // Barometric pressure(cm)
21 m = 0.85           // Irregularity factor
22 m_v_local = 0.72   // Roughness factor for local
                      corona
23 m_v_gen = 0.82    // Roughness factor for general
                      corona
24
25 // Calculations
26 delta = 3.92*b/(273.0+t)

                      // Air density factor
27 r = d/2.0

                      // Radius of conductor(cm)
28 E_0 = 21.1*m*delta*r*log(D/r)                         //
                      Critical disruptive voltage(kV) rms
29 E_v_local = 21.1*m_v_local*delta*r*(1+(0.3/(delta*r) **0.5))*log(D/r)      // Critical disruptive
                      voltage for local corona(kV) rms
30 E_v_gen = 21.1*m_v_gen*delta*r*(1+(0.3/(delta*r) **0.5))*log(D/r)           // Critical disruptive
                      voltage for general corona(kV) rms
31 E = V/3**0.5

                      // Phase voltage(kV)
32 // Case(i)
33 P_c_i = 244.0*10**-5*(f+25)/delta*(r/D)**0.5*(E-E_0) **2                  // Peek's formula for fair
                      weather condition(kW/km/phase)
34 P_c_i_total = P_c_i*l*3

                      // Total power loss(kW)
35 // Case(ii)
36 P_c_ii = 244.0*10**-5*(f+25)/delta*(r/D)**0.5*(E -0.8*E_0)**2             // Peek's formula for

```

```

        stormy condition (kW/km/phase)
37 P_c_ii_total = P_c_ii*l*3

        // Total power loss (kW)
38 // Case(iii)
39 F_iii = 0.0713

        // From text depending on E/E_0
40 P_c_iii = 21.0*10**-6*f*E**2*F_iii/(log10(D/r))**2
        // Peterson's formula for
        fair condition (kW/km/phase)
41 P_c_iii_total = P_c_iii*l*3
        //

        Total power loss (kW)
42 // Case(iv)
43 F_iv = 0.3945

        // From text depending on E/E_0
44 P_c_iv = 21.0*10**-6*f*E**2*F_iv/(log10(D/r))**2
        // Peterson's formula
        for stormy condition (kW/km/phase)
45 P_c_iv_total = P_c_iv*l*3

        // Total power loss (kW)
46
47 // Results
48 disp("PART II - EXAMPLE : 8.4 : SOLUTION :-")
49 printf("\nCase(i) : Power loss due to corona using
        Peek formula for fair weather condition , P_c = %
        .3 f kW/km/phase" , P_c_i)
50 printf("\n            Total corona loss in fair
        weather condition using Peek formula = %.1f kW" ,
        P_c_i_total)
51 printf("\nCase(ii) : Power loss due to corona using
        Peek formula for stormy weather condition , P_c =
        %.2f kW/km/phase" , P_c_ii)
52 printf("\n            Total corona loss in stormy
        condition using Peek formula = %.f kW" ,

```

```

        P_c_ii_total)
53 printf("\nCase( iii): Power loss due to corona using
      Peterson formula for fair weather condition , P_c
      = %.4f kW/km/phase" , P_c_iii)
54 printf("\n          Total corona loss in fair
      condition using Peterson formula = %.2f kW" ,
      P_c_iii_total)
55 printf("\nCase( iii): Power loss due to corona using
      Peterson formula for fair weather condition , P_c
      = %.4f kW/km/phase" , P_c_iv)
56 printf("\n          Total corona loss in stormy
      condition using Peterson formula = %.1f kW \n" ,
      P_c_iv_total)
57 printf("\nNOTE: ERROR: Calculation mistake in the
      final answer in textbook")

```

---

### Scilab code Exa 15.5 Spacing between the conductors

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 8: CORONA
8
9 // EXAMPLE : 8.5 :
10 // Page number 229
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 V = 132.0           // Operating line voltage(kV)
15 dia = 1.956         // Diameter of conductor(cm)
16 v_c = 210.0         // Disruptive voltage(kV)

```

```

17 g_0 = 30.0/2**0.5      // Breakdown strength of air(kV
    /cm)
18
19 // Calculations
20 r = dia/2.0                      // Radius
    of conductor(cm)
21 V_c = v_c/3**0.5                  //
    Disruptive voltage/phase(kV)
22 m_0 = 1.0                        //
    Irregularity factor
23 delta = 1.0                      //
    density factor
24 d = exp(V_c/(m_0*delta*g_0*r))*r // Spacing
    between conductors(cm)
25
26 // Results
27 disp("PART II - EXAMPLE : 8.5 : SOLUTION :-")
28 printf("\nSpacing between the conductors , d = %.f cm
    \n", abs(d))
29 printf("\nNOTE: Changes in the obtained answer from
    that of textbook is due to precision")

```

---

### Scilab code Exa 15.6 Disruptive critical voltage and Corona loss

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 8: CORONA
8
9 // EXAMPLE : 8.6 :
10 // Page number 229
11 clear ; clc ; close ; // Clear the work space and

```

```

        console
12
13 // Given data
14 P_c1 = 53.0          // Total corona loss (kW)
15 V_1 = 106.0          // Operating line voltage (kV)
16 P_c2 = 98.0          // Total corona loss (kW)
17 V_2 = 110.9          // Operating line voltage (kV)
18 V_3 = 113.0          // Operating line voltage (kV)
19
20 // Calculations
21 E_1 = V_1/3**0.5      // Phase
   voltage (kV)
22 E_2 = V_2/3**0.5      // Phase
   voltage (kV)
23 P_ratio = (P_c2/P_c1)**0.5
24 E_0 = (P_ratio*E_1-E_2)/(P_ratio-1)    //
   Disruptive critical voltage (kV)
25 E_3 = V_3/3**0.5      // Phase
   voltage (kV)
26 W = ((E_3-E_0)/(E_1-E_0))**2*P_c1      // Corona
   loss at 113 kV(kW)
27
28 // Results
29 disp("PART II - EXAMPLE : 8.6 : SOLUTION :-")
30 printf("\nDisruptive critical voltage , E_0 = %.f kV"
   , E_0)
31 printf("\nCorona loss at 113 kV, W = %.f kW\n" , W)
32 printf("\nNOTE: Changes in obtained answer from
   textbook is due to more precision here")

```

---

**Scilab code Exa 15.7 Corona will be present in the air space or not**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.

```

```

4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 8: CORONA
8
9 // EXAMPLE : 8.7 :
10 // Page number 229–230
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 d = 3.0           // Diameter of conductor(cm)
15 e_r = 4.0         // Relative permittivity
16 d_1 = 3.5         // Internal diameter of
    porcelain bushing(cm)
17 d_2 = 9.0          // External diameter of
    porcelain bushing(cm)
18 V = 25.0          // Voltage b/w conductor and
    clamp(kV)
19
20 // Calculations
21 r = d/2.0

    // Radius of conductor(cm)
22 r_1 = d_1/2.0

    // Internal radius of porcelain bushing(cm)
23 r_2 = d_2/2.0

    // External radius of porcelain bushing(cm)
24 g_2max = r/(e_r*r_1)
    //
    // Maximum gradient of inner side of porcelain
25 g_1max = V/(r*log(r_1/r)+g_2max*r_1*log(r_2/r_1))
    //
    // Maximum gradient on surface of
    // conductor(kV/cm)
26
27 // Results

```

```

28 disp("PART II – EXAMPLE : 8.7 : SOLUTION :-")
29 printf("\nMaximum gradient on surface of conductor ,
      g_1max = %.2f kV/cm", g_1max)
30 printf("\nSince, gradient exceeds 21.1 kV/cm, corona
      will be present")

```

---

### Scilab code Exa 15.8 Line voltage for commencing of corona

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 8: CORONA
8
9 // EXAMPLE : 8.8 :
10 // Page number 230
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 d = 2.0           // Diameter of conductor(cm)
15 D = 150.0         // Spacing b/w conductor(cm)
16 delta = 1.0       // Air density factor
17
18 // Calculations
19 r = d/2.0          // Radius of
   conductor(cm)
20 V_d = 21.1*delta*r*log(D/r)        // Disruptive
   critical voltage(kV/phase)
21 V_d_ll = 3**0.5*V_d                // Line voltage
   for commencing of corona(kV)
22
23 // Results

```

```
24 disp("PART II – EXAMPLE : 8.8 : SOLUTION :-")  
25 printf("\nLine voltage for commencing of corona = %  
       .2 f kV \n", V_d_11)  
26 printf("\nNOTE: Solution is incomplete in textbook")
```

---

# Chapter 16

## LOAD FLOW STUDY USING COMPUTER TECHNIQUES

Scilab code Exa 16.1 Bus admittance matrix Ybus

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 9: LOAD FLOW STUDY USING COMPUTER
    TECHNIQUES
8
9 // EXAMPLE : 9.1 :
10 // Page number 235–236
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 Z_L1 = complex(14.3,97)          // Series impedance of
    line L1(ohm)
15 Z_PL1 = complex(0,-3274)         // Shunt impedance of
    line L1(ohm)
```

```

16 Z_L2 = complex(7.13,48.6)           // Series impedance of
   line L2(ohm)
17 Z_PL2 = complex(0,-6547)            // Shunt impedance of
   line L2(ohm)
18 Z_L3 = complex(9.38,64)             // Series impedance of
   line L3(ohm)
19 Z_PL3 = complex(0,-4976)            // Shunt impedance of
   line L3(ohm)
20
21 // Calculations
22 Y_S12 = 1.0/Z_L1                  // Series
   admittance(mho)
23 Y_P12 = 1.0/Z_PL1                 // Shunt
   admittance(mho)
24 Y_S23 = 1.0/Z_L3                  // Series
   admittance(mho)
25 Y_P23 = 1.0/Z_PL3                 // Shunt
   admittance(mho)
26 Y_S13 = 1.0/Z_L2                  // Series
   admittance(mho)
27 Y_P13 = 1.0/Z_PL2                 // Shunt
   admittance(mho)
28 Y_11 = Y_P12+Y_P13+Y_S12+Y_S13   // Admittance(mho)
29 Y_12 = -Y_S12                     // Admittance(mho)
30 Y_13 = -Y_S13                     // Admittance(mho)
31 Y_21 = Y_12                       // Admittance(mho)
32 Y_22 = Y_P12+Y_P23+Y_S12+Y_S23   // Admittance(mho)
33 Y_23 = -Y_S23                     // Admittance(mho)
34 Y_31 = Y_13                       // Admittance(mho)
35 Y_32 = Y_23                       // Admittance(mho)
36 Y_33 = Y_P13+Y_P23+Y_S23+Y_S13   // Admittance(mho)
37 Y_bus = [[Y_11, Y_12, Y_13],      ]
38           [Y_21, Y_22, Y_23],      ]
39           [Y_31, Y_32, Y_33]]
40
41 // Results
42 disp("PART II - EXAMPLE : 9.1 : SOLUTION :-")
43 printf("\n[ Y_bus ] = \n"); disp(Y_bus)

```

---

### Scilab code Exa 16.3 Voltage values at different buses

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 9: LOAD FLOW STUDY USING COMPUTER
    TECHNIQUES
8
9 // EXAMPLE : 9.3 :
10 // Page number 236–237
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V_a = 1.0 // Voltage(p.u)
15 V_b = 1.0*exp(%i*-36.87*pi/180) // Voltage(p.u)
16 V_c = 1.0 // Voltage(p.u)
17 Z_1 = complex(0,1) // Reactance(p.u)
18 Z_2 = complex(0,1) // Reactance(p.u)
19 Z_3 = complex(0,1) // Reactance(p.u)
20 Z_13 = complex(0,0.4) // Reactance(p.u)
21 Z_23 = complex(0,0.4) // Reactance(p.u)
22 Z_14 = complex(0,0.2) //
```

```

        Reactance(p.u)
23 Z_24 = complex(0,0.2)           //
        Reactance(p.u)
24 Z_34 = complex(0,0.2)           //
        Reactance(p.u)
25 Z_12 = complex(0,0)             //
        Reactance(p.u)

26
27 // Calculations
28 I_1 = V_a/Z_1                  // Current injection vector(p.
        u)
29 I_2 = V_b/Z_2                  // Current injection vector(p.
        u)
30 I_3 = V_c/Z_3                  // Current injection vector(p.
        u)
31 I_4 = 0.0                      // Current injection vector(p.
        u)
32 y1 = 1.0/Z_1                  // Admittance(p.u)
33 y2 = 1.0/Z_2                  // Admittance(p.u)
34 y3 = 1.0/Z_3                  // Admittance(p.u)
35 y13 = 1.0/Z_13                // Admittance(p.u)
36 y23 = 1.0/Z_23                // Admittance(p.u)
37 y14 = 1.0/Z_14                // Admittance(p.u)
38 y24 = 1.0/Z_24                // Admittance(p.u)
39 y34 = 1.0/Z_34                // Admittance(p.u)
40 y12 = 0.0                      // Admittance(p.u)
41 Y_11 = y1+y13+y14            // Equivalent admittance(p.u)
42 Y_12 = y12                      // Equivalent admittance(p.u)
43 Y_13 = -y13                     // Equivalent admittance(p.u)
44 Y_14 = -y14                     // Equivalent admittance(p.u)
45 Y_21 = Y_12                     // Equivalent admittance(p.u)
46 Y_22 = y2+y23+y24            // Equivalent admittance(p.u)
47 Y_23 = -y23                     // Equivalent admittance(p.u)
48 Y_24 = -y24                     // Equivalent admittance(p.u)
49 Y_31 = Y_13                     // Equivalent admittance(p.u)
50 Y_32 = Y_23                     // Equivalent admittance(p.u)
51 Y_33 = y3+y13+y23+y34            // Equivalent admittance(p.u)
52 Y_34 = -y34                     // Equivalent admittance(p.u)

```

```

53 Y_41 = Y_14 // Equivalent admittance(p.u)
54 Y_42 = Y_24 // Equivalent admittance(p.u)
55 Y_43 = Y_34 // Equivalent admittance(p.u)
56 Y_44 = y14+y24+y34 // Equivalent admittance(p.u)
57 Y_bus = [[Y_11, Y_12, Y_13, Y_14],
58 [Y_21, Y_22, Y_23, Y_24],
59 [Y_31, Y_32, Y_33, Y_34],
60 [Y_41, Y_42, Y_43, Y_44]] // Bus
admittance matrix
61 I_bus = [I_1,
62 I_2,
63 I_3,
64 I_4]
65 V = inv(Y_bus)*I_bus // Bus
voltage(p.u)
66
67 // Results
68 disp("PART II - EXAMPLE : 9.3 : SOLUTION :-")
69 printf("\nVoltage at bus 1, V_1 = %.4f%.4fj p.u",
real(V(1,1:1)),imag(V(1,1:1)))
70 printf("\nVoltage at bus 2, V_2 = %.4f%.4fj p.u",
real(V(2,1:1)),imag(V(2,1:1)))
71 printf("\nVoltage at bus 3, V_3 = %.4f%.4fj p.u",
real(V(3,1:1)),imag(V(3,1:1)))
72 printf("\nVoltage at bus 4, V_4 = %.4f%.4fj p.u\n",
real(V(4,1:1)),imag(V(4,1:1)))
73 printf("\nNOTE: Node equation matrix could not be
represented in a single equation. Hence, it is
not displayed")

```

---

#### Scilab code Exa 16.4 New bus admittance matrix Ybus

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.

```

```

4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 9: LOAD FLOW STUDY USING COMPUTER
    TECHNIQUES
8
9 // EXAMPLE : 9.4 :
10 // Page number 237–238
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V_a = 1.0 // Voltage(p.u)
15 V_b = 1.0*exp(%i*-36.87*pi/180) // Voltage(p.u)
16 V_c = 1.0 // Voltage(p.u)
17 Z_1 = complex(0,1) // Reactance(p.u)
18 Z_2 = complex(0,1) // Reactance(p.u)
19 Z_3 = complex(0,1) // Reactance(p.u)
20 Z_13 = complex(0,0.4) // Reactance(p.u)
21 Z_23 = complex(0,0.4) // Reactance(p.u)
22 Z_14 = complex(0,0.2) // Reactance(p.u)
23 Z_24 = complex(0,0.2) // Reactance(p.u)
24 Z_34 = complex(0,0.2) // Reactance(p.u)
25 Z_12 = complex(0,0) // Reactance(p.u)
26
27 // Calculations

```

```

28 I_1 = V_a/Z_1           // Current injection vector(p.
   u)
29 I_2 = V_b/Z_2           // Current injection vector(p.
   u)
30 I_3 = V_c/Z_3           // Current injection vector(p.
   u)
31 I_4 = 0.0                // Current injection vector(p.
   u)
32 y1 = 1.0/Z_1             // Admittance(p.u)
33 y2 = 1.0/Z_2             // Admittance(p.u)
34 y3 = 1.0/Z_3             // Admittance(p.u)
35 y13 = 1.0/Z_13           // Admittance(p.u)
36 y23 = 1.0/Z_23           // Admittance(p.u)
37 y14 = 1.0/Z_14           // Admittance(p.u)
38 y24 = 1.0/Z_24           // Admittance(p.u)
39 y34 = 1.0/Z_34           // Admittance(p.u)
40 y12 = 0.0                // Admittance(p.u)
41 Y_11 = y1+y13+y14        // Equivalent admittance(p.u)
42 Y_12 = y12                // Equivalent admittance(p.u)
43 Y_13 = -y13               // Equivalent admittance(p.u)
44 Y_14 = -y14               // Equivalent admittance(p.u)
45 Y_21 = Y_12               // Equivalent admittance(p.u)
46 Y_22 = y2+y23+y24        // Equivalent admittance(p.u)
47 Y_23 = -y23               // Equivalent admittance(p.u)
48 Y_24 = -y24               // Equivalent admittance(p.u)
49 Y_31 = Y_13               // Equivalent admittance(p.u)
50 Y_32 = Y_23               // Equivalent admittance(p.u)
51 Y_33 = y3+y13+y23+y34    // Equivalent admittance(p.u)
52 Y_34 = -y34               // Equivalent admittance(p.u)
53 Y_41 = Y_14               // Equivalent admittance(p.u)
54 Y_42 = Y_24               // Equivalent admittance(p.u)
55 Y_43 = Y_34               // Equivalent admittance(p.u)
56 Y_44 = y14+y24+y34        // Equivalent admittance(p.u)
57 Y_bus = [[Y_11, Y_12, Y_13, Y_14],
58                 [Y_21, Y_22, Y_23, Y_24],
59                 [Y_31, Y_32, Y_33, Y_34],
60                 [Y_41, Y_42, Y_43, Y_44]]          // 
                                         Bus admittance matrix

```

```

61 K = Y_bus([1,2],1:2)
62 L = Y_bus([1,2],3:4)
63 M = Y_bus([3,4],3:4)
64 N = Y_bus([3,4],1:2)
65 inv_M = inv([M(1,1:2);M(2,1:2)]) // Multiplication of matrix [L][M^(-1)][N]
66 Y_bus_new = K-L*inv_M*N // New bus admittance matrix
67
68 // Results
69 disp("PART II - EXAMPLE : 9.4 : SOLUTION :-")
70 printf("\n[Y_bus]_new = \n"); disp(Y_bus_new)
71 printf("\nNOTE: ERROR: Mistake in representing the
sign in final answer in textbook")

```

---

### Scilab code Exa 16.5 Bus admittance matrix V1 and V2

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 9: LOAD FLOW STUDY USING COMPUTER
TECHNIQUES
8
9 // EXAMPLE : 9.5 :
10 // Page number 238
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 I_1 = 2.0 // Voltage(p.u)
15 I_2 = 2.0*exp(%i*45.0*pi/180) //
```

```

    Voltage(p.u)
16  y1 = complex(0,-1.0)                                //
    Admittance(p.u)
17  y2 = complex(0,-2.0)                                //
    Admittance(p.u)
18  y12 = complex(0,-2.0)                                //
    Admittance(p.u)

19
20 // Calculations
21 E_1 = I_1*y1                                         // Voltage
    element(p.u)
22 E_2 = I_2*y2                                         // Voltage
    element(p.u)
23 Y_11 = y1+y12                                       // Self
    Admittance(p.u)
24 Y_12 = -y12                                         // Mutual
    Admittance(p.u)
25 Y_21 = Y_12                                         // Mutual
    Admittance(p.u)
26 Y_22 = y2+y12                                       // Self
    Admittance(p.u)
27 Y_bus = [[Y_11, Y_12],
28                 [Y_21, Y_22]]                           // Bus
    admittance matrix
29 I_bus = [I_1,
30           I_2]
31 V = inv(Y_bus)*I_bus
32 V_1 = V(1,1:1)                                       // Voltage(
    p.u)
33 V_2 = V(2,1:1)                                       // Voltage(
    p.u)

34
35 // Results
36 disp("PART II - EXAMPLE : 9.5 : SOLUTION :-")
37 printf("\n[Y_bus] = \n"); disp(Y_bus)
38 printf("\nV_1 = %.3 f % .1 f p.u", abs(V_1),
    phasemag(V_1))
39 printf("\nV_2 = %.3 f % .1 f p.u\n", abs(V_2),
    phasemag(V_2))

```

```
    phasemag(V_2))
40 printf("\nNOTE: ERROR: Calculation mistake in V_1 in
      textbook")
```

---

### Scilab code Exa 16.6 Bus impedance matrix Zbus

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 9: LOAD FLOW STUDY USING COMPUTER
      TECHNIQUES
8
9 // EXAMPLE : 9.6 :
10 // Page number 238
11 clear ; clc ; close ; // Clear the work space and
      console
12
13 // Given data
14 Y_bus = [[-%i*10.5, 0, %i*5.0, %i*5.0],
15           [0, -%i*8.0, %i*2.5, %i*5.0],
16           [%i*5.0, %i*2.5, -%i*18.0, %i*10.0],
17           [%i*5.0, %i*5.0, %i*10.0, -%i*20.0]] // Bus admittance matrix
18
19 // Calculations
20 Z_bus = inv(Y_bus) // Bus impedance matrix
21
22 // Results
23 disp("PART II - EXAMPLE : 9.6 : SOLUTION :-")
24 printf("\n[ Z_bus ] = \n') ; disp(Z_bus)
```

---

### Scilab code Exa 16.7 Power flow expressions

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 9: LOAD FLOW STUDY USING COMPUTER
    TECHNIQUES
8
9 // EXAMPLE : 9.7 :
10 // Page number 239
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 Y_C = complex(0,0.1) // Shunt
    admittance(mho)
15 Z_L = complex(0,0.2) // Series
    impedance(mho)
16
17 // Calculations
18 Y_L = 1.0/Z_L // Series
    admittance(mho)
19 Y_11 = Y_C+Y_C+Y_L+Y_L // Admittance (mho)
20 Y_12 = -Y_L // Admittance (mho)
21 Y_13 = -Y_L // Admittance (mho)
22 Y_21 = Y_12 // Admittance (mho)
23 Y_22 = Y_L+Y_L+Y_C+Y_C // Admittance (mho)
24 Y_23 = -Y_L // Admittance (mho)
25 Y_31 = Y_13 // Admittance (mho)
26 Y_32 = Y_23 // Admittance (mho)
27 Y_33 = Y_L+Y_L+Y_C+Y_C // Admittance (mho)
```

```

28 Y_bus = [[Y_11, Y_12, Y_13],
29             [Y_21, Y_22, Y_23],
30             [Y_31, Y_32, Y_33]]      // Bus admittance
31 S_11 = conj(Y_bus(1,1:1))
32 S_12 = conj(Y_bus(1,2:2))
33 S_13 = conj(Y_bus(1,3:3))
34 S_21 = S_12
35 S_22 = conj(Y_bus(2,2:2))
36 S_23 = conj(Y_bus(2,3:3))
37 S_31 = S_13
38 S_32 = S_23
39 S_33 = conj(Y_bus(3,3:3))
40
41 // Results
42 disp("PART II - EXAMPLE : 9.7 : SOLUTION :-")
43 printf("\nPower flow expressions are:")
44 printf("\nS_1 = %.1fj | V_1|^2 %.1fjV_1V_2* %.1fjV_3*"
        , imag(S_11), imag(S_12), imag(S_13))
45 printf("\nS_2 = %.1fjV_2V_1* + %.1fj | V_2|^2 %.1fjV_2V_3*", imag(S_21), imag(S_22), imag(S_23))
46 printf("\nS_3 = %.1fjV_3V_1* %.1fjV_3V_2* + %.1fj | V_3|^2", imag(S_31), imag(S_32), imag(S_33))

```

---

### Scilab code Exa 16.8 Voltage V2 by GS method

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti, M.L. Soni, P.V. Gupta, U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 9: LOAD FLOW STUDY USING COMPUTER
    TECHNIQUES
8

```

```

9 // EXAMPLE : 9.8 :
10 // Page number 242
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V_1 = 1.0                                // Voltage(p.u)
15 S_g2 = complex(0,1.0)                      // Complex power
    generated(p.u)
16 S_D2 = complex(0.5,1.0)                    // Complex power
    demand(p.u)
17 Z_L = complex(0,0.5)                       // Impedance(p.u)
18
19 // Calculations
20 Y_L = 1.0/Z_L                            //
    Admittance(p.u)
21 Y_22 = Y_L                                //
    Admittance(mho)
22 Y_21 = -Y_L                               //
    Admittance(mho)
23 S_2 = S_g2-S_D2
24 V_2_0 = 1.0                                //
    Initial guess
25 V_2_1 = 1.0/Y_22*((conj(S_2/V_2_0))-Y_21*V_1) //
    V_2(p.u). In 1st iteration
26 V_2_2 = 1.0/Y_22*((conj(S_2/V_2_1))-Y_21*V_1) //
    V_2(p.u). In 2nd iteration
27 V_2_3 = 1.0/Y_22*((conj(S_2/V_2_2))-Y_21*V_1) //
    V_2(p.u). In 3rd iteration
28 V_2_4 = 1.0/Y_22*((conj(S_2/V_2_3))-Y_21*V_1) //
    V_2(p.u). In 4th iteration
29 V_2_5 = 1.0/Y_22*((conj(S_2/V_2_4))-Y_21*V_1) //
    V_2(p.u). In 5th iteration
30 V_2_6 = 1.0/Y_22*((conj(S_2/V_2_5))-Y_21*V_1) //
    V_2(p.u). In 6th iteration
31
32 // Results
33 disp("PART II - EXAMPLE : 9.8 : SOLUTION :-")

```

```
34 printf("\nBy G-S method, V_2 = %.6f % .5f    p.u\n",  
        abs(V_2_6), phasemag(V_2_6))
```

---

# Chapter 17

## POWER SYSTEM STABILITY

Scilab code Exa 17.1 Operating power angle and Magnitude of P0

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.1 :
10 // Page number 270
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 Z = 0.1           // Impedance of transmission line(p.u
   )
15 M = 0.3           // Stability margin
16 X = 1.0           // Constant(p.u)
17
```

```

18 // Calculations
19 sin_delta_0 = 1-M                         // Sin( _0 )
20 delta_0 = asind(sin_delta_0)                // _0 (   )
21 P_0 = X/Z*sin_delta_0                     // Magnitude of P_0
     (p.u)
22
23 // Results
24 disp("PART II - EXAMPLE : 10.1 : SOLUTION :-")
25 printf("\nOperating power angle, _0 = %.2 f    ",
        delta_0)
26 printf("\nP_0 = %.2 f p.u", P_0)

```

---

### Scilab code Exa 17.2 Minimum value of E and VL Maximum power limit and Steady state

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.2 :
10 // Page number 270
11 clear ; clc ; close ; // Clear the work space and
                           console
12
13 // Given data
14 x_s = 0.85      // Reactance(p.u)
15 x_T1 = 0.157    // Reactance(p.u)
16 x_T2 = 0.157    // Reactance(p.u)
17 x_l1 = 0.35     // Reactance(p.u)
18 x_l2 = 0.35     // Reactance(p.u)
19 E = 1.50         // Sending end voltage(p.u)
20 V_L = 1.0         // Load voltage(p.u)

```

```

21 P_0 = 1.0           // Stable power output(p.u)
22
23 // Calculations
24 x = x_s+x_T1+x_T2+(x_11/2)           // Total
   reactance(p.u)
25 P_max = E*V_L/x                      // Maximum power
   limit(p.u)
26 M = (P_max-P_0)/P_max*100            // Steady state
   stability margin(%)
27 V_Lmin = P_0*x/E                     // Minimum value
   of V_L(p.u)
28 E_min = P_0*x/V_L                   // Minimum value
   of E(p.u)
29
30 // Results
31 disp("PART II - EXAMPLE : 10.2 : SOLUTION :-")
32 printf("\nMinimum value of |E| , |E_min| = %.3f p.u" ,
   E_min)
33 printf("\nMinimum value of |V_L| , |V_Lmin| = %.3f p.
   u" , V_Lmin)
34 printf("\nMaximum power limit , P_0 = %.2f p.u" ,
   P_max)
35 printf("\nSteady state stability margin , M = %.1f
   percent" , M)

```

---

### Scilab code Exa 17.3 Maximum power transfer if shunt inductor and Shunt capacitor

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8

```

```

9 // EXAMPLE : 10.3 :
10 // Page number 270-271
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 E_1 = 1.25      // Sending end voltage(p.u)
15 x_d = 1.0       // Reactance(p.u)
16 x_T1 = 0.2      // Reactance(p.u)
17 x_l1 = 1.0      // Reactance(p.u)
18 x_l2 = 1.0      // Reactance(p.u)
19 x_T2 = 0.2      // Reactance(p.u)
20 E_2 = 1.0       // Receiving end voltage(p.u)
21 x_L = 1.0       // Shunt inductor reactance(p.u)
22 x_C = 1.0       // Shunt capacitor reactance(p.u)
23
24 // Calculations
25 // Case(a)
26 Z_1_a = x_d+x_T1+(x_l1/2.0)           //
    Reactance(p.u)
27 Z_2_a = x_T2+x_d                         //
    Reactance(p.u)
28 Z_3_a = x_L                               //
    Reactance(p.u)
29 Z_a = Z_1_a+Z_2_a+(Z_1_a*Z_2_a/Z_3_a)  // Transfer
    reactance(p.u)
30 P_max_1 = E_1*E_2/Z_a                   // Maximum
    power transfer if shunt inductor is connected at
    bus 2(p.u)
31 // Case(b)
32 Z_1_b = x_d+x_T1+(x_l1/2.0)           //
    Reactance(p.u)
33 Z_2_b = x_T2+x_d                         //
    Reactance(p.u)
34 Z_3_b = -x_C                            //
    Reactance(p.u)
35 Z_b = Z_1_b+Z_2_b+(Z_1_b*Z_2_b/Z_3_b) // Transfer
    reactance(p.u)

```

```

36 P_max_2 = E_1*E_2/Z_b // Maximum
    power transfer if shunt capacitor is connected at
    bus 2(p.u)
37
38 // Results
39 disp("PART II - EXAMPLE : 10.3 : SOLUTION :-")
40 printf("\nCase(a): Maximum power transfer if shunt
    inductor is connected at bus 2, P_max1 = %.3f p.u
    ", P_max_1)
41 printf("\nCase(b): Maximum power transfer if shunt
    capacitor is connected at bus 2, P_max2 = %.2f p.
    u", P_max_2)

```

---

#### Scilab code Exa 17.4 Maximum power transfer and Stability margin

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.4 :
10 // Page number 271
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 400.0 // Voltage(kV)
15 L = 220.0 // Line length(km)
16 P = 0.58 // Initial real power transfer(p.
    u)
17 PF = 0.85 // Lagging power factor
18 V_L = 1.00 // Load bus voltage(p.u)

```

```

19 x_d = 0.460           // Reactance(p.u)
20 x_T1 = 0.200          // Reactance(p.u)
21 x_T2 = 0.15           // Reactance(p.u)
22 x_line = 0.7          // Reactance(p.u)
23
24 // Calculations
25 x = x_d+x_T1+x_T2+(x_line/2)           // Net
   reactance(p.u)
26 phi = acosd(PF)                      // (
   )
27 Q = P*tand(phi)                     // /
   Reactive power(p.u)
28 E = ((V_L+(Q*x/V_L))**2+(P*x/V_L)**2)**0.5 // /
   Excitation voltage of generator(p.u)
29 P_max = E*V_L/x                      // /
   Maximum power transfer(p.u)
30 M = (P_max-P)/P_max*100             // /
   Steady state stability margin(%)
31
32 // Results
33 disp("PART II - EXAMPLE : 10.4 : SOLUTION :-")
34 printf("\nMaximum power transfer , P_max = %.2f p.u" ,
   P_max)
35 printf("\nStability margin , M = %.f percent" , M)

```

---

**Scilab code Exa 17.5 QgB Phase angle of VB and What happens if QgB is made zero**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8

```

```

9 // EXAMPLE : 10.5 :
10 // Page number 271–272
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V_A = 1.0          // Voltage at bus A(p.u)
15 Z_AB = %i*0.5      // Impedance(p.u)
16 S_DA = 1.0          // p.u
17 S_DB = 1.0          // p.u
18 V_B = 1.0          // Voltage at bus B(p.u)
19
20 // Calculations
21 // Case(i) & (ii)
22 X = abs(Z_AB)           //
    Reactance(p.u)
23 sin_delta = 1.0*X/(V_A*V_B)           // Sin
24 delta = asind(sin_delta)           //
    )
25 V_2 = V_B
26 V_1 = V_A
27 Q_gB = (V_2**2/X)-(V_2*V_1*cosd(delta)/X)
28 // Case(iii)
29 V_2_3 = 1/2.0**0.5           //
    Solving quadratic equation from textbook
30 delta_3 = acosd(V_2_3)           //
    )
31
32 // Results
33 disp("PART II – EXAMPLE : 10.5 : SOLUTION :-")
34 printf("\nCase( i ) : Q_gB = %.3f", Q_gB)
35 printf("\nCase( ii ) : Phase angle of V_B,      = %.f
    ", delta)
36 printf("\nCase( iii ) : If Q_gB is equal to zero then
    amount of power transmitted is , V_2 = %.3f %
    f    ", V_2_3,delta_3)

```

---

**Scilab code Exa 17.6 Steady state stability limit with two terminal voltages const**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.6 :
10 // Page number 272
11 clear ; clc ; close ; // Clear the work space and
    console
12 funcprot(0)
13
14 // Given data
15 A = 0.98*exp(%i*0.3*pi/180)           // Constant
16 B = 82.5*exp(%i*76.0*pi/180)          // Constant(ohm)
17 C = 0.0005*exp(%i*90.0*pi/180)        // Constant(mho)
18 D = A                                     // Constant
19 V_S = 110.0                                // Sending end
    voltage(kV)
20 V_R = 110.0                                // Receiving end
    voltage(kV)
21
22 // Calculations
23 alpha = phasemag(A)

    //      (
24 beta = phasemag(B)

    //      (
25 P_max = (V_S*V_R/abs(B))-(abs(A)*V_R**2/abs(B)*cosd
```

```

        ((beta-alpha))) // Maximum power transfer (MW)
26 B_new = abs(B)*sind(beta)                                //
                                                               Constant (ohm)
27 beta_new = 90.0

// ( )
28 P_max_new = (V_S*V_R/B_new)-(V_R**2/B_new*cosd(
    beta_new)) // Maximum power transfer (MW)
)
29
30 // Results
31 disp("PART II - EXAMPLE : 10.6 : SOLUTION :-")
32 printf("\nSteady state stability limit , P_max = %.2f
    MW", P_max)
33 printf("\nSteady state stability limit if shunt
    admittance is zero & series resistance neglected ,
    P_max = %.2f MW \n", P_max_new)
34 printf("\nNOTE: Changes in the obtained answer from
    that of textbook is due to precision")

```

---

### Scilab code Exa 17.8 Power angle diagram Maximum power the line is capable of trans-

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.8 :
10 // Page number 273-275

```

Fig E10.7 . Power angle diagram

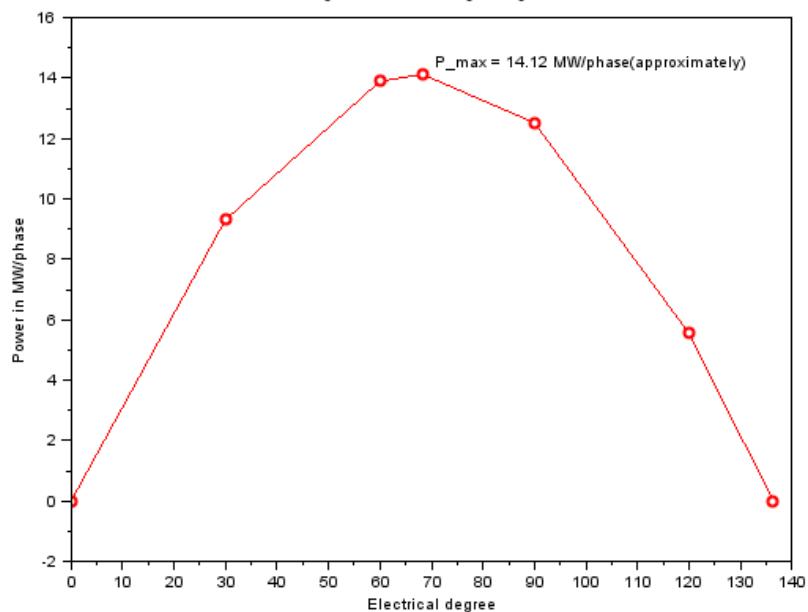


Figure 17.1: Power angle diagram Maximum power the line is capable of transmitting and Power transmitted with equal voltage at both ends

```

11 clear ; clc ; close ; // Clear the work space and
    console
12 funcprot(0)
13
14 // Given data
15 V = 33.0*10**3          // Line voltage (V)
16 R = 6.0                  // Resistance per phase (ohm)
17 X = 15.0                 // Reactance per phase (ohm)
18
19 // Calculations
20 V_S = V/3**0.5           //
                           Sending end phase voltage (V)
21 V_R = V/3**0.5           //
                           Receiving end phase voltage (V)
22 beta = atand(X/R)        //
                           (
                           )
23 Z = (R**2+X**2)**0.5      //
                           Impedance (ohm)
24 delta_0 = 0.0             //
                           (
                           )
25 P_0 = (V_R/Z**2)*(V_S*Z*cosd((delta_0-beta))-V_R*R)   //
   /10**6 // Power received (MW/phase)
26 delta_1 = 30.0            //
                           (
                           )
27 P_1 = (V_R/Z**2)*(V_S*Z*cosd((delta_1-beta))-V_R*R)   //
   /10**6 // Power received (MW/phase)
28 delta_2 = 60.0            //
                           (
                           )
29 P_2 = (V_R/Z**2)*(V_S*Z*cosd((delta_2-beta))-V_R*R)   //
   /10**6 // Power received (MW/phase)
30 delta_3 = beta

```

```

// ( )
31 P_3 = (V_R/Z**2)*(V_S*Z*cosd((delta_3-beta))-V_R*R)
      /10**6 // Power received (MW/phase)
32 delta_4 = 90.0
// ( )
33 P_4 = (V_R/Z**2)*(V_S*Z*cosd((delta_4-beta))-V_R*R)
      /10**6 // Power received (MW/phase)
34 delta_5 = 120.0
// ( )
35 P_5 = (V_R/Z**2)*(V_S*Z*cosd((delta_5-beta))-V_R*R)
      /10**6 // Power received (MW/phase)
36 delta_6 = (acosd(R/Z))+beta
// ( )
37 P_6 = (V_R/Z**2)*(V_S*Z*cosd((delta_6-beta))-V_R*R)
      /10**6 // Power received (MW/phase)
38
39
40 delta = [delta_0,delta_1,delta_2,delta_3,delta_4,
           delta_5,delta_6]
41 P = [P_0,P_1,P_2,P_3,P_4,P_5,P_6]
42 a = gca(); // sets thickness of plot
43 a.thickness = 2
44 plot(delta,P,'ro-') // labels x-axis
45 a.xlabel.text = 'Electrical degree' // labels y-axis
46 a.ylabel.text = 'Power in MW/phase' // sets thickness of axes
47 xtitle("Fig E10.7 . Power angle diagram") // sets thickness of axes
48 xset('thickness',2) // sets thickness of axes
49 xstring(70,14.12,'P_max = 14.12 MW/phase(
           approximately)')
50 P_max = V_R/Z**2*(V_S*Z-V_R*R)/10**6
// Maximum

```

```

        power transmitted (MW/ phase)
51 delta_equal = 0.0

        // With no phase shift( )
52 P_no_shift = (V_R/Z**2)*(V_S*Z*cosd((delta_equal-
    beta))-V_R*R)/10**6 // Power transmitted with
    no phase shift (MW/ phase)

53
54 // Results
55 disp("PART II - EXAMPLE : 10.8 : SOLUTION :-")
56 printf("\nPower angle diagram is plotted and is
    shown in the Figure 1")
57 printf("\nMaximum power the line is capable of
    transmitting , P_max = %.2f MW/phase" , P_max)
58 printf("\nWith equal voltage at both ends power
    transmitted = %.f MW/phase" , abs(P_no_shift))

```

---

**Scilab code Exa 17.9** Maximum steady state power that can be transmitted over the line.

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.9 :
10 // Page number 275
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 132.0*10**3           // Sending end voltage(
    V)

```

```

15 Z_line = complex(4,6)           // Line impedance per
16   phase(ohm)
17 // Calculations
18 V_S = V/3**0.5
19   // Sending end phase voltage(V)
20 V_R = V/3**0.5
21   // Receiving end phase voltage(V)
22 Z = abs(Z_line)
23   // Impedance(ohm)
24 R = real(Z_line)
25   // Resistance per phase(ohm)
26 P_max_phase = ((V_S*V_R/Z)-(R*V_R**2/Z**2))/10**6
27   // Maximum steady state power that can be
28   transmitted over the line(MW/phase)
29 P_max_total = 3.0*P_max_phase
30   // Maximum steady state
31   power that can be transmitted over the line(MW)
32
33 // Results
34 disp("PART II - EXAMPLE : 10.9 : SOLUTION :-")
35 printf("\nMaximum steady state power that can be
36 transmitted over the line , P_max = %.f MW ( total
37 3-phase)", P_max_total)

```

---

**Scilab code Exa 17.10 Maximum steady state power Value of P and Q if static capaci**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION

```

```

5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.10 :
10 // Page number 275–276
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 E_1 = 1.1      // Sending end voltage(p.u)
15 x_d1 = 1.0     // Reactance(p.u)
16 x_T1 = 0.1     // Reactance(p.u)
17 x_l1 = 0.4     // Reactance(p.u)
18 x_l2 = 0.4     // Reactance(p.u)
19 x_T2 = 0.1     // Reactance(p.u)
20 E_2 = 1.0      // Receiving end voltage(p.u)
21 x_d2 = 1.0     // Reactance(p.u)
22 x_L = 1.0      // Shunt inductor reactance(p.u)
23 x_C = 1.0      // Static capacitor reactance(p.u)
24 delta = 30.0   // ( )
25
26 // Calculations
27 // Case(a)
28 Z_1_a = x_d1+x_T1+(x_l1/2.0)
                    // Reactance(p.u)
29 X_1_a = %i*Z_1_a
30 Z_2_a = x_T2+x_d2
                    //
            Reactance(p.u)
31 X_2_a = %i*Z_2_a
32 Z_3_a = -x_C
                    //
            Reactance(p.u)
33 X_3_a = %i*Z_3_a
34 X_a = X_1_a+X_2_a+(X_1_a*X_2_a/X_3_a)
                    // Transfer reactance(p.u)
35 P_max_a = E_1*E_2/abs(X_a)

```

```

                                // Maximum steady
state power if static capacitor is connected(p.u)
36 P_a = P_max_a*sind(delta)                                // Value of P(p.u)
37 Q_a = (E_1*E_2/abs(X_a))*cosd(delta)-(E_2**2/abs(X_a
)) // Value of Q(p.u)
38 // Case(b)
39 Z_1_b = x_d1+x_T1+(x_l1/2.0)                            // Reactance(p.u)
40 X_1_b = %i*Z_1_b
41 Z_2_b = x_T2+x_d2                                         //
        Reactance(p.u)
42 X_2_b = %i*Z_2_b
43 Z_3_b = x_L                                               //
        Reactance(p.u)
44 X_3_b = %i*Z_3_b
45 X_b = X_1_b+X_2_b+(X_1_b*X_2_b/X_3_b)                  // Transfer reactance(p.u)
46 P_max_b = E_1*E_2/abs(X_b)                                // Maximum steady
state power if static capacitor is replaced by an
inductive reactor(p.u)
47 P_b = P_max_b*sind(delta)                                // Value of P(p.u)
48 Q_b = (E_1*E_2/abs(X_b))*cosd(delta)-(E_2**2/abs(X_b
)) // Value of Q(p.u)
49
50 // Results
51 disp("PART II - EXAMPLE : 10.10 : SOLUTION :-")
52 printf("\nCase(a): Maximum steady state power if
static capacitor is connected , P_max = %.3f p.u" ,
P_max_a)
53 printf("\n          Value of P = %.3f p.u" , P_a)
54 printf("\n          Value of Q = %.3f p.u" , Q_a)
55 printf("\nCase(b): Maximum steady state power if
static capacitor is replaced by an inductive

```

```

        reactor , P_max = %.3f p.u" , P_max_b)
56 printf("\n          Value of P = %.3f p.u" , P_b)
57 printf("\n          Value of Q = %.4f p.u" , Q_b)

```

---

**Scilab code Exa 17.11 Kinetic energy stored in the rotor at synchronous speed and**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.11 :
10 // Page number 303
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 f = 50.0           // Frequency(Hz)
15 G = 100.0          // Rating of generator(MVA)
16 H = 5.0            // Inertia constant(MJ/MVA)
17 P_a = 20.0          // Acceleration power(MVA)
18
19 // Calculations
20 GH = G*H           // Energy stored in rotor at
    synchronous speed(MJ)
21 M = GH/(180*f)      // Angular momentum
22 acceleration = P_a/M // Acceleration( /sec^2)
23
24 // Results
25 disp("PART II - EXAMPLE : 10.11 : SOLUTION :-")
26 printf("\nKinetic energy stored in the rotor at
    synchronous speed , GH = %.f MJ" , GH)

```

```
27 printf("\nAcceleration = %.f / sec ^2", acceleration)
```

---

**Scilab code Exa 17.12 Kinetic energy stored in the rotor at synchronous speed and**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.12 :
10 // Page number 303–304
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 f = 50.0          // Frequency (Hz)
15 P = 4.0           // Number of poles
16 G = 20.0          // Rating of generator (MVA)
17 H = 9.0           // Inertia constant (kWsec/MVA)
18 P_m = 26800.0    // Rotational loss (hp)
19 P_e = 16000.0    // Electric power developed (kW)
20
21 // Calculations
22 GH = G*H          //
    Energy stored in rotor at synchronous speed (MJ)
23 P_m_kW = P_m*0.746 //
    Rotational loss (kW)
24 P_a = P_m_kW-P_e //
    Acceleration power (kW)
25 P_a1 = P_a/1000.0 //
    Acceleration power (MW)
26 M = GH/(180*f)   //
```

```

        Angular momentum
27 acceleration = P_a1/M                                //
    Acceleration( /sec^2)
28 acceleration_1 = acceleration*%pi/180.0             //
    Acceleration( rad/sec^2)
29
30 // Results
31 disp("PART II - EXAMPLE : 10.12 : SOLUTION :-")
32 printf("\nKinetic energy stored in the rotor at
    synchronous speed , GH = %.f MJ", GH)
33 printf("\nAcceleration = %.f /sec^2 = %.2f rad/sec
    ^2 \n", acceleration,acceleration_1)
34 printf("\nNOTE: ERROR: H = 9 kW-sec/MVA, not 9 kW-
    sec/kVA as mentioned in the textbook statement")

```

---

**Scilab code Exa 17.13 Change in torque angle in that period and RPM at the end of**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.13 :
10 // Page number 304
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 f = 50.0          // Frequency(Hz)
15 P = 4.0           // Number of poles
16 alpha = 200.0     // Acceleration( /sec^2)
17 alpha_rad = 3.49  // Acceleration( rad/sec^2)

```

```

18 n = 10.0           // Number of cycle
19
20 // Calculations
21 t = 1/f*n          // Time(sec)
22 delta_rel = ((alpha_rad*2)**0.5*0.5)**2 // Relation
   of change in rotor angle with time(rad)
23 delta = delta_rel*t**2          // Change
   in torque angle(rad)
24 delta_deg = delta*180/%pi      // Change
   in torque angle in that period()
25 rpm_rad = (alpha_rad*2*delta)**0.5 // r.p.m(
   rad/sec)
26 rpm = rpm_rad*60.0/(%pi*P)      // r.p.m
27 speed_rotor = (120*f/P)+rpm    // Rotor
   speed at the end of 10 cycles(r.p.m)
28
29 // Results
30 disp("PART II - EXAMPLE : 10.13 : SOLUTION :-")
31 printf("\nChange in torque angle in that period ,"
   "= %.4f rad = %.f elect degree", delta, delta_deg)
32 printf("\nRotor speed at the end of 10 cycles = %.2f"
   " r.p.m", speed_rotor)

```

---

**Scilab code Exa 17.14 Accelerating torque at the time the fault occurs**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.14 :

```

```

10 // Page number 304
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 Power = 20.0*10**3           // Rating of generator(kVA)
15 PF = 0.8                      // Lagging power factor
16 fault = 0.5                   // Reduction in output
   under fault
17 P = 4.0                        // Number of poles
18 f = 50.0                       // Frequency(Hz)
19
20 // Calculations
21 P_m = Power*PF                // Output power before
   fault(kW)
22 P_e = fault*P_m                // Output after fault(kW)
23 P_a = P_m-P_e                 // Accelerating power(kW)
24 w_s = 4.0*pi*f/P             // Speed
25 T_a = P_a*10**3/w_s          // Accelerating torque at
   the time the fault occurs(N-m)
26
27 // Results
28 disp("PART II - EXAMPLE : 10.14 : SOLUTION :-")
29 printf("\nAccelerating torque at the time the fault
   occurs , T_a = %.2f N-m" , T_a)

```

---

### Scilab code Exa 17.16 Value of H and in 100 MVA base

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY

```

```

8
9 // EXAMPLE : 10.16 :
10 // Page number 304–305
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 S = 1000.0          // Rating of generator(MVA)
15 N = 1500.0          // Speed of alternator(r.p.m)
16 WR_sq = 5.0*10**6   // WR^2(lb.ft^2)
17
18 // Calculations
19 H = 2.31*10**-10*WR_sq*N**2/S           // Inertia
    constant (MJ/MVA)
20 H_100 = H*1000.0/100                     // Inertia
    constant on 100 MVA(MJ/MVA)
21
22 // Results
23 disp("PART II – EXAMPLE : 10.16 : SOLUTION :-")
24 printf("\nValue of inertia constant , H = %.1f MJ/MVA
    ", H)
25 printf("\nValue of inertia constant in 100 MVA base ,
    H = %.f MJ/MVA" , H_100)

```

---

**Scilab code Exa 17.17 Equivalent H for the two to common 100 MVA base**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.17 :

```

```

10 // Page number 305
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 MVA_1 = 500.0          // Rating of generator (MVA)
15 H_1 = 4.0              // Inertia constant (MJ/VA)
16 MVA_2 = 1000.0         // Rating of generator (MVA)
17 H_2 = 3.5              // Inertia constant (MJ/VA)
18 MVA = 100.0            // Base MVA
19
20 // Calculations
21 KE_T = H_1*MVA_1+H_2*MVA_2      // Total KE of the
   system (MJ)
22 H_total = KE_T/MVA             // Equivalent H for
   the two to common 100MVA base (MJ/MVA)
23
24 // Results
25 disp("PART II – EXAMPLE : 10.17 : SOLUTION :-")
26 printf("\nEquivalent H for the two to common 100 MVA
   base , H = %.f MJ/MVA" , H_total)

```

---

**Scilab code Exa 17.18 Energy stored in the rotor at the rated speed Value of H and**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.18 :
10 // Page number 305
11 clear ; clc ; close ; // Clear the work space and

```

```

        console
12
13 // Given data
14 MVA = 210.0          // Rating of generator (MVA)
15 P = 2.0              // Number of poles
16 f = 50.0             // Frequency (Hz)
17 MI = 60.0*10**3     // Moment of inertia (kg-mt^2)
18
19 // Calculations
20 N = 120.0*f/P       // Speed (r.p.m)
21 KE = 1.0/2*MI*(2*pi*N/f)**2/10**6 // Energy stored in the rotor at rated speed (MJ)
22 H = KE/MVA           // Inertia constant (MJ/MVA)
23 G = MVA
24 M = G*H/(180*f)     // Angular momentum (MJ-sec/elect. degree)
25
26 // Results
27 disp("PART II - EXAMPLE : 10.18 : SOLUTION :-")
28 printf("\nEnergy stored in the rotor at the rated speed , KE = %.2e MJ" , KE)
29 printf("\nValue of inertia constant , H = %.2f MJ/MVA" , H)
30 printf("\nAngular momentum , M = %.3f MJ-sec/elect. degree" , M)

```

---

### Scilab code Exa 17.19 Acceleration of the rotor

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5

```

```

6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.19 :
10 // Page number 305
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 P_accl = 30.0          // Acceleration power(MVA)
15 M = 0.474              // Angular momentum(MJ-sec /
    elect.degree). From Example 10.18
16
17 // Calculations
18 acceleration = P_accl/M // Acceleration of the
    rotor(elect.degree/sec^2)
19
20 // Results
21 disp("PART II – EXAMPLE : 10.19 : SOLUTION :-")
22 printf("\nAcceleration of the rotor = %.2f elect.
    degree/sec^2", acceleration)

```

---

**Scilab code Exa 17.20 Accelerating power and New power angle after 10 cycles**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.20 :
10 // Page number 305
11 clear ; clc ; close ; // Clear the work space and

```

```

        console
12
13 // Given data
14 MVA = 50.0          // Rating of alternator (MVA)
15 P = 4.0             // Number of poles
16 f = 50.0            // Frequency (Hz)
17 KE = 150.0          // Kinetic energy stored in
                      // rotor (MJ)
18 P_m = 25.0          // Machine input (MW)
19 P_e = 22.5          // Developed power (MW)
20 n = 10.0            // Number of cycles
21
22 // Calculations
23 P_a = P_m - P_e    // Accelerating power (MW)
24 H = KE/MVA          // Inertia constant (MJ/MVA)
25 G = MVA
26 M_deg = G*H/(180*f) // Angular momentum (MJ-sec /
                      // elect. degree)
27 M = G*H/(%pi*f)     // Angular momentum (MJ-sec /
                      // rad)
28 acceleration = P_a/M // Accelerating power (rad /
                      // sec ^2)
29 t = 1/f*n           // Time (sec)
30 delta = 1.309*t**2  // Term in
31
32 // Results
33 disp("PART II - EXAMPLE : 10.20 : SOLUTION :-")
34 printf("\nAccelerating power = %.3f rad/sec^2 , "
       acceleration)
35 printf("\nNew power angle after 10 cycles ,      = (%.3
       f + _0 ) rad", delta)

```

---

Scilab code Exa 17.21 Kinetic energy stored by rotor at synchronous speed and Acce

1 // A Texbook on POWER SYSTEM ENGINEERING

```

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.21 :
10 // Page number 305–306
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 f = 50.0          // Frequency (Hz)
15 P = 4.0           // Number of poles
16 G = 20.0          // Rating of turbo-generator (MVA)
17 V = 13.2          // Voltage (kV)
18 H = 9.0           // Inertia constant (kW–sec /kVA)
19 P_s = 20.0         // Input power less rotational loss (
    MW)
20 P_e = 15.0         // Output power (MW)
21
22 // Calculations
23 KE = G*H           // Kinetic energy
    stored (MJ)
24 M = G*H/(180*f)      // Angular momentum
    (MJ–sec / elect . degree )
25 P_a = P_s-P_e        // Accelerating
    power (MW)
26 alpha = P_a/M         // Acceleration (
    elect . degree / sec ^2)
27 alpha_deg = alpha/2.0      // Acceleration (
    degree / sec ^2)
28 alpha_rpm = 60.0*alpha_deg/360 // Acceleration (rpm
    / sec )
29
30 // Results
31 disp("PART II – EXAMPLE : 10.21 : SOLUTION :-")

```

```

32 printf("\nCase(a): Kinetic energy stored by rotor at
           synchronous speed , GH = %.f MJ" , KE)
33 printf("\nCase(b): Acceleration ,      = %.f degree/sec
           ^2" , alpha_deg)
34 printf("\n          Acceleration ,      = %.2f rpm/sec" ,
           alpha_rpm)

```

---

**Scilab code Exa 17.22 Change in torque angle and Speed in rpm at the end of 10 cycles**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.22 :
10 // Page number 306
11 clear ; clc ; close ; // Clear the work space and
                           console
12
13 // Given data
14 f = 50.0             // Frequency(Hz)
15 P = 4.0              // Number of poles
16 G = 20.0              // Rating of turbo-generator(MVA)
17 V = 13.2              // Voltage(kV)
18 H = 9.0                // Inertia constant(kW-sec/kVA)
19 P_s = 20.0             // Input power less rotational loss(
                           MW)
20 P_e = 15.0             // Output power(MW)
21 n = 10.0              // Number of cycles
22
23 // Calculations
24 KE = G*H                  // Kinetic energy

```

```

        stored (MJ)
25 M = G*H/(180*f)           // Angular momentum
      (MJ-sec / elect . degree)
26 P_a = P_s-P_e             // Accelerating
      power (MW)
27 alpha = P_a/M             // Acceleration (
      elect . degree/sec ^2)
28 alpha_deg = alpha/2.0       // Acceleration (
      degree/sec ^2)
29 alpha_rpm = 60.0*alpha_deg/360 // Acceleration (rpm
      /sec)
30 t = 1.0/f*n               // Time( sec)
31 delta = 1.0/2*alpha*t**2   // Change in torque
      angle(elect . degree)
32 N_s = 120*f/P             // Synchronous
      speed(rpm)
33 speed = N_s+alpha_rpm*t    // Speed at the end
      of 10 cycles(rpm)
34
35 // Results
36 disp("PART II - EXAMPLE : 10.22 : SOLUTION :-")
37 printf("\nChange in torque angle in that period ,
      = %.f elect degrees.", delta)
38 printf("\nSpeed in rpm at the end of 10 cycles = %.2
      f rpm", speed)

```

---

### Scilab code Exa 17.23 Accelerating torque at the time of fault occurrence

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY

```

```

8
9 // EXAMPLE : 10.23 :
10 // Page number 306
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 G = 20.0          // Rating of turbo-generator (MVA)
15 PF = 0.75         // Lagging power factor
16 fault = 0.5        // Fault reduces output power
17 N_s = 1500.0       // Synchronous speed (rpm). From
    Example 10.22
18
19 // Calculations
20 P_prefault = PF*G           // Pre-fault output power(
    MW)
21 P_a = P_prefault*fault     // Post-fault output power
    (MW)
22 w = 2.0*pi*N_s/60          // (rad/sec)
23 T_a = P_a*10**6/w          // Accelerating torque at
    the time of fault occurrence (N-m)
24
25 // Results
26 disp("PART II - EXAMPLE : 10.23 : SOLUTION :-")
27 printf("\nAccelerating torque at the time of fault
    occurrence , T_a = %.f N-m" , T_a)

```

---

### Scilab code Exa 17.24 Swing equation

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION

```

```

7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.24 :
10 // Page number 306–307
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 x_d = %i*0.2          // Transient reactance of
    generator(p.u)
15 P_e = 0.8              // Power delivered (p.u)
16 V_t = 1.05             // Terminal voltage (p.u)
17 H = 4.0                // Inertia constant (kW–sec/kVA)
18 x_t = %i*0.1          // Transformer reactance (p.u)
19 x_l = %i*0.4          // Transmission line reactance (p
    .u)
20 V = 1.0                // Infinite bus voltage (p.u)
21 f = 50.0               // Frequency (Hz)
22
23 // Calculations
24 x_12 = x_d+x_t+(x_l/2)           // Reactance
    b/w bus 1 & 2(p.u)
25 y_12 = 1/x_12                  //
    Admittance b/w bus 1 & 2(p.u)
26 y_21 = y_12                   //
    Admittance b/w bus 2 & 1(p.u)
27 y_10 = 0.0                     //
    // Admittance b/w bus 1 & 0(p.u)
28 y_20 = 0.0
    // Admittance b/w bus 2 & 0(p.u)
29 Y_11 = y_12+y_10            //
    // Admittance at bus 1(p.u)

```

```

30 Y_12 = -y_12 // Admittance b/w bus 1 & 2(p.u)
31 Y_21 = -y_12 // Admittance b/w bus 2 & 1(p.u)
32 Y_22 = y_21+y_20 // Admittance at bus 2(p.u)
33 x_32 = x_t+(x_1/2) // Reactance b/w bus 3 & 1(p.u)
34 theta_t = asind(P_e*abs(x_32)/V_t) // Angle( )
35 V_t1 = V_t*exp(%i*theta_t*pi/180) // Terminal voltage(p.u)
36 I = (V_t1-V)/x_32 // Current(p.u)
37 E = V_t1+I*x_d // Alternator voltage(p.u)
38 sine = poly(0,"sin")
39 P_e1 = 2.0*abs(E) // Developed power(p.u) in terms of sin
40 P_m_P_e = P_e-P_e1*sine
41 M = 2*H/(2*pi*f) // Angular momentum
42 acc = (P_e-P_e1*sine)*2*pi*f/(2*H) // Acceleration = (rad/
sec^2)
43
44 // Results
45 disp("PART II - EXAMPLE : 10.24 : SOLUTION :-")
46 printf("\nSwing equation is , %.4f* = %.1f - %.3
fsin \n" , M,P_e,P_e1)

```

```
47 printf("\nNOTE: Swing equation is simplified and  
        represented here")  
48 printf("\n        ERROR: x_d = 0.2 p.u, not 0.1 p.u as  
        mentioned in textbook statement")
```

---

### Scilab code Exa 17.26 Critical clearing angle

```
1 // A Texbook on POWER SYSTEM ENGINEERING  
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  
3 // DHANPAT RAI & Co.  
4 // SECOND EDITION  
5  
6 // PART II : TRANSMISSION AND DISTRIBUTION  
7 // CHAPTER 10: POWER SYSTEM STABILITY  
8  
9 // EXAMPLE : 10.26 :  
10 // Page number 308–309  
11 clear ; clc ; close ; // Clear the work space and  
    console  
12  
13 // Given data  
14 X_d = 0.25          // Transient reactance of  
    generator(p.u)  
15 X_t1 = 0.15         // Reactance of transformer(p.u)  
16 X_t2 = 0.15         // Reactance of transformer(p.u)  
17 X_t3 = 0.15         // Reactance of transformer(p.u)  
18 X_t4 = 0.15         // Reactance of transformer(p.u)  
19 X_l1 = 0.20          // Reactance of line(p.u)  
20 X_l2 = 0.20          // Reactance of line(p.u)  
21 X_tr = 0.15          // Reactance of transformer(p.u)  
22 P_m = 1.0            // Power delivered(p.u)  
23 E = 1.20             // Voltage behind transient  
    reactance(p.u)  
24 V = 1.0              // Infinite bus voltage(p.u)  
25
```

```

26 // Calculations
27 X_14 = X_d+((X_t1+X_t2+X_l1)/2)+X_tr
                    // Reactance before fault(p
                    .u)
28 x_1_b = X_t1+X_t2+X_l1
                    // Reactance(
                    p.u). From figure (b)
29 x_2_b = X_l2+X_t4
                    //
                    Reactance(p.u). From figure (b)
30 x_1 = x_1_b*X_t3/(x_1_b+x_2_b+X_t3)
                    // Reactance(p.u). From
                    figure (c)
31 x_2 = x_1_b*x_2_b/(x_1_b+x_2_b+X_t3)
                    // Reactance(p.u). From
                    figure (c)
32 x_3 = X_t3*x_2_b/(x_1_b+x_2_b+X_t3)
                    // Reactance(p.u). From
                    figure (c)
33 X_14_fault = x_1+X_d+x_2+X_tr+((x_1+X_d)*(x_2+X_tr)/
            x_3) // Reactance under fault(p.u)
34 X_14_after_fault = X_d+X_t1+X_l1+X_t2+X_tr
                    // Reactance after fault is
                    cleared(p.u)
35 P_max = V*E/X_14
                    //
                    Maximum power transfer(p.u)
36 gamma_1 = (V*E/X_14_fault)/P_max
                    // _1
37 gamma_2 = (V*E/X_14_after_fault)/P_max
                    // _2
38 delta_0 = asin(P_m/P_max)
                    // _0 (radians)
39 delta_0_degree = delta_0*180/%pi
                    // _0 ( )
40 delta_m = %pi-asin(P_m/(gamma_2*P_max))
                    // _1 (radians)
41 delta_m_degree = delta_m*180/%pi

```

```

                                // _1 ( )
42 delta_c = acosd((P_m/P_max*(delta_m-delta_0)+gamma_2-
    *cos(delta_m)-gamma_1*cos(delta_0))/(gamma_2-
    gamma_1)) // Clearing angle( )
43
44 // Results
45 disp("PART II - EXAMPLE : 10.26 : SOLUTION :-")
46 printf("\nCritical clearing angle , _c = %.2 f ", 
    delta_c)

```

---

### Scilab code Exa 17.27 Critical angle using equal area criterion

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.27 :
10 // Page number 309–310
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 f = 50.0                      // Frequency(Hz)
15 P_m = 1.0                      // Power delivered(p.u)
16 P_max = 1.8                     // Maximum power(p.u)
17 gamma_1_P_max = 0.4             // Reduced maximum power
    after fault(p.u)
18 gamma_2_P_max = 1.30            // Maximum power after fault
    clearance(p.u)
19
20 // Calculations

```

```

21 delta_0 = asin(P_m/P_max) // -0 (radians)
22 delta_0_degree = delta_0*180/%pi // -0 ( )
23 delta_f = %pi-asin(P_m/(gamma_2_P_max)) // -1 (radians)
24 delta_f_degree = delta_f*180/%pi // -1 ( )
25 gamma_1 = gamma_1_P_max/P_max // -1
26 gamma_2 = gamma_2_P_max/P_max // -2
27 delta_c = acosd(1.0/(gamma_2-gamma_1)*((delta_f-
    delta_0)*sin(delta_0)+(gamma_2*cos(delta_f)-
    gamma_1*cos(delta_0)))) // Clearing angle( )
28
29 // Results
30 disp("PART II - EXAMPLE : 10.27 : SOLUTION :-")
31 printf("\nCritical angle , -c = %.2 f ", delta_c)

```

---

### Scilab code Exa 17.28 Critical clearing angle

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.28 :
10 // Page number 310
11 clear ; clc ; close ; // Clear the work space and
    console
12

```

```

13 // Given data
14 sin_delta_0 = 0.45      // Supplying percent of peak
   power capacity before fault
15 x = 4.0                  // Reactance under fault
   increased
16 gamma_2 = 0.7            // Peak power delivered after
   fault clearance
17
18 // Calculations
19 delta_0 = asin(sin_delta_0)
   // _0 (radians)
20 delta_0_degree = delta_0*180/%pi
   // _0 ( )
21 gamma_1 = 1.0/x
   // _1
22 delta_m = %pi-asin(sin_delta_0/(gamma_2))
   // _m (radians)
23 delta_m_degree = delta_m*180/%pi
   // _m ( )
24 delta_c = acosd(1.0/(gamma_2-gamma_1)*((delta_m-
   delta_0)*sin(delta_0)+(gamma_2*cos(delta_m)-
   gamma_1*cos(delta_0)))) // Clearing angle( )
25
26 // Results
27 disp("PART II - EXAMPLE : 10.28 : SOLUTION :-")
28 printf("\nCritical clearing angle , _c = %.f ", 
   delta_c)

```

---

### Scilab code Exa 17.30 Power angle and Swing curve data

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5

```

```

6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 10: POWER SYSTEM STABILITY
8
9 // EXAMPLE : 10.30 :
10 // Page number 310–311
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 f = 60.0          // Frequency(Hz)
15 P = 6.0           // Number of poles
16 H = 4.0           // Inertia constant(p.u)
17 P_e = 1.0          // Power supplied by generator(p.u
    )
18 E = 1.2           // Internal voltage(p.u)
19 V = 1.0           // Infinite bus voltage(p.u)
20 X = 0.3           // Line reactance(p.u)
21 del_t = 0.05       // t = Interval step size(sec)
22
23 // Calculations
24 P_max = E*V/X           //
    Maximum power(p.u)
25 delta_0 = asind(P_e/P_max)      //
    (-)
26 G = P_e
27 M = G*H/(180*f)           //
    Angular momentum(p.u)
28 P_a_0 = 1.0/2*(P_e-0)        //
    )
29 alpha_0 = P_a_0/M           //
    (/sec^2)
30 del_w_r_1 = alpha_0*del_t     //
    _r1   (/sec)
31 w_r_1 = 0+del_w_r_1         //
    _r1   (/sec)
32 del_delta_1 = w_r_1*del_t     //
    _1   ()
33 delta_1 = delta_0+del_delta_1 // _1

```

```

        (
)
34 P_a_1 = 1.0*(P_e-0)                                // (p.u
        )
35 alpha_1 = P_a_1/M                                 // -1
        (/sec^2)
36 del_w_r_2 = alpha_1*del_t                          //
        _r2   (/sec)
37 w_r_2 = del_w_r_1+del_w_r_2                      //
        _r2   (/sec)
38 del_delta_2 = w_r_2*del_t                         //
        _2   ()
39 delta_2 = delta_1+del_delta_2                     // -2
        ()
40 del_w_r_3 = del_w_r_2                           //
        _r3   (/sec)
41 w_r_3 = w_r_2+del_w_r_3                         //
        _r3   (/sec)
42 del_delta_3 = w_r_3*del_t                         //
        _3   ()
43 delta_3 = delta_2+del_delta_3                     // -3
        ()
44 del_w_r_4 = del_w_r_2                           //
        _r4   (/sec)
45 w_r_4 = w_r_3+del_w_r_4                         //
        _r4   (/sec)
46 del_delta_4 = w_r_4*del_t                         //
        _4   ()
47 delta_4 = delta_3+del_delta_4                     // -4
        ()
48 del_w_r_5 = del_w_r_2                           //
        _r5   (/sec)
49 w_r_5 = w_r_4+del_w_r_5                         //
        _r5   (/sec)
50 del_delta_5 = w_r_5*del_t                         //
        _5   ()
51 delta_5 = delta_4+del_delta_5                     // -5
        ()

```

```

53 // Results
54 disp("PART II - EXAMPLE : 10.30 : SOLUTION :-")
55 printf("\nPower angle , _0 = %.2 f ", delta_0)
56 printf("\nValue of vs t are:")
57 printf("\n-----")
58 printf("\n t(Sec) : (degree)")
59 printf("\n-----")
60 printf("\n %.1 f : %.2 f ", 0, delta_0)
61 printf("\n %.2 f : %.2 f ", (del_t), delta_1)
62 printf("\n %.2 f : %.2 f ", (del_t+del_t),
       delta_2)
63 printf("\n %.2 f : %.2 f ", (del_t*3), delta_3
      )
64 printf("\n %.2 f : %.2 f ", (del_t*4), delta_4
      )
65 printf("\n %.2 f : %.2 f ", (del_t*5), delta_5
      )
66 printf("\n-----")

```

---

# Chapter 18

## LOAD FREQUENCY CONTROL AND LOAD SHARING OF POWER GENERATING SOURCES

Scilab code Exa 18.1 Load shared by two machines and Load at which one machine cea

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 11: LOAD FREQUENCY CONTROL AND LOAD
    SHARING OF POWER GENERATING SOURCES
8
9 // EXAMPLE : 11.1 :
10 // Page number 330
11 clear ; clc ; close ; // Clear the work space and
    console
12 funcprot(0)
13
```

```

14 // Given data
15 rating = 1000.0          // Rating of alternator (kW)
16 load = 1600.0            // Total load (kW)
17 X_f1 = 100.0             // Full load speed regulation
18 Y_f1 = 104.0             of alternator X(%)
19 X_nl = 100.0             of alternator Y(%)
20 Y_nl = 105.0             of alternator X(%)
21
22 // Calculations
23 h = poly(0,"h")
24 PB = (Y_nl-X_nl)-h
25 PR = rating/(Y_nl-X_nl)*PB           // Load shared
26 by machine X(kW) in terms of h
26 QQ = (Y_f1-X_f1)-h
27 RQ = rating/(Y_f1-X_f1)*QQ           // Load shared
27 by machine Y(kW) in terms of h
28 h_1 = roots(PR+RQ-load)
29 PB_1 = (Y_nl-X_nl)-h_1
30 PR_1 = rating/(Y_nl-X_nl)*PB_1       // Load shared
30 by machine X(kW)
31 QQ_1 = (Y_f1-X_f1)-h_1
32 RQ_1 = rating/(Y_f1-X_f1)*QQ_1       // Load shared
32 by machine Y(kW)
33 load_cease = rating/(Y_nl-X_nl)        // Y cease
33 supply load (kW)
34
35 // Results
36 disp("PART II - EXAMPLE : 11.1 : SOLUTION :-")
37 printf("\nLoad shared by machine X, PR = %.f kW",
37 PR_1)
38 printf("\nLoad shared by machine Y, RQ = %.f kW",
38 RQ_1)
39 printf("\nLoad at which machine Y ceases to supply
39 any portion of load = %.f kW", load_cease)

```

---

### Scilab code Exa 18.2 Synchronizing power and Synchronizing torque for no load and

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 11: LOAD FREQUENCY CONTROL AND LOAD
    SHARING OF POWER GENERATING SOURCES
8
9 // EXAMPLE : 11.2 :
10 // Page number 330–331
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 kVA = 5000.0          // Rating of alternator (kVA)
15 N = 1500.0            // Speed(rpm)
16 V = 6600.0            // Voltage(V)
17 f = 50.0              // Frequency(Hz)
18 PF = 0.8              // Lagging power factor
19 x = 0.15              // Short circuit reactance
20
21 // Calculations
22 E = V/3**0.5

    // Phase voltage(V)
23 I = kVA*1000/(3**0.5*V)                                // Full
    load current of alternator (A)
24 V_drop = E*x

    // Synchronous reactance drop(V)
```

```

25 X = V_drop/I
      // Synchronous reactance per phase(ohm)
26 P = 120*f/N
      // Number of poles
27 n = N/60
      // Speed(rps)
28 phi = acosd(PF)
      // ( )
29 // Case(a)
30 theta_a = 2.0
      // For a 4 pole m/c. 1 mech degree = 2 elect
      degree
31 E_s_a = E*sind(theta_a)
      // Synchronizing voltage(V)
32 I_s_a = E_s_a/X
      // Synchronizing current(A)
33 P_s_a = E*I_s_a
      // Synchronizing power per phase(W)
34 P_s_a_total = 3.0*P_s_a
      // Total synchronizing power(W)
35 P_s_a_total_kw = P_s_a_total/1000.0
      // Total synchronizing power(kW)
36 T_s_a = P_s_a_total/(2*pi*n)
      // Synchronizing torque(N-m)
37 // Case(b)
38 sin_phi = sind(phi)
39 OB = ((E*PF)**2+(E*sin_phi+V_drop)**2)**0.5

```

```

        // Voltage (V)
40 E_b = 0B

        // Voltage (V)
41 alpha_phi = atand((E*sin_phi+V_drop)/(E*PF))
                // + ( )
42 alpha = alpha_phi-phi
                // (
)
43 E_s_b = 2.0*E_b*sind(2.0/2)
                //
        Synchronizing voltage (V)
44 I_s_b = E_s_b/X

        // Synchronizing current (A)
45 P_s_b = E*I_s_b*cosd((alpha+1.0))
                // Synchronizing
        power per phase (W)
46 P_s_b_total = 3.0*P_s_b
                // Total
        synchronizing power (W)
47 P_s_b_total_kw = P_s_b_total/1000.0
                // Total
        synchronizing power (kW)
48 T_s_b = P_s_b_total/(2*%pi*n)
                //
        Synchronizing torque (N-m)
49
50 // Results
51 disp("PART II - EXAMPLE : 11.2 : SOLUTION :-")
52 printf("\nCase(a): Synchronizing power for no-load ,
        P_s = %.1f kW", P_s_a_total_kw)
53 printf("\n        Synchronizing torque for no-load ,
        T_s = %.f N-m", T_s_a)
54 printf("\nCase(b): Synchronizing power at full-load ,
        P_s = %.1f kW", P_s_b_total_kw)
55 printf("\n        Synchronizing torque at full-load
        , T_s = %.f N-m \n", T_s_b)

```

```
56 printf("\nNOTE: ERROR: Calculation mistakes in  
textbook")
```

---

### Scilab code Exa 18.3 Armature current EMF and PF of the other alternator

```
1 // A Texbook on POWER SYSTEM ENGINEERING  
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  
3 // DHANPAT RAI & Co.  
4 // SECOND EDITION  
5  
6 // PART II : TRANSMISSION AND DISTRIBUTION  
7 // CHAPTER 11: LOAD FREQUENCY CONTROL AND LOAD  
SHARING OF POWER GENERATING SOURCES  
8  
9 // EXAMPLE : 11.3 :  
10 // Page number 331–332  
11 clear ; clc ; close ; // Clear the work space and  
console  
12  
13 // Given data  
14 V = 6600.0 // Voltage(V)  
15 R = 0.045 // Resistance(ohm)  
16 X = 0.45 // Reactance(ohm)  
17 Load = 10000.0*10**3 // Total load(W)  
18 PF = 0.8 // Lagging power factor  
19 I_a = 437.5 // Armature current(A)  
20  
21 // Calculations  
22 I = Load/(3**0.5*V*PF) //  
Load current(A)  
23 I_working = PF*I //  
Working component of current(A)  
24 I_watless = (1-PF**2)**0.5*I //  
Watless component of current(A)  
25 I_second = (I_a**2+I_watless**2)**0.5 //
```

```

        Load current supplied by second alternator (A)
26 PF_second = I_a/I_second                                //
        Lagging power factor of second alternator
27 V_ph = V/3**0.5                                         //
        Terminal voltage per phase(V)
28 I_R = I_second*R                                       //
        Voltage drop due to resistance(V)
29 I_X = I_second*X                                       //
        Voltage drop due to reactance(V)
30 sin_phi_second = (1-PF_second**2)**0.5
31 E = ((V_ph+I_R*PF_second+I_X*sin_phi_second)**2+(I_X
        *PF_second-I_R*sin_phi_second)**2)**0.5 // EMF
        of the alternator(V/phase)
32 E_ll = 3**0.5*E                                       //
        Line-to-line EMF of the alternator(V)
33
34 // Results
35 disp("PART II – EXAMPLE : 11.3 : SOLUTION :-")
36 printf("\nArmature current of other alternator = %.1
        f A", I_second)
37 printf("\nem.f of other alternator = %.f V (line-to
        -line)", E_ll)
38 printf("\nPower factor of other alternator = %.3f (
        lagging)", PF_second)

```

---

#### Scilab code Exa 18.4 New value of machine current and PF Power output Current and

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 11: LOAD FREQUENCY CONTROL AND LOAD
        SHARING OF POWER GENERATING SOURCES

```

```

8
9 // EXAMPLE : 11.4 :
10 // Page number 332–333
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 X = 10.0                      // Reactance(ohm)
15 I_a = 220.0                    // Armature current(A)
16 PF = 1.0                       // Unity power factor
17 V = 11000.0                    // Phase voltage(V)
18 emf_raised = 0.2              // EMF raised by 20%
19
20 // Calculations
21 I_X = I_a*X                  // Reactance drop
    (V)
22 E_0 = (V**2+I_X**2)**0.5      // EMF(V)
23 E_00 = (1+emf_raised)*E_0     // New value of
    induced emf(V)
24 U = ((E_00**2-I_X**2)**0.5-V)/X // Current(A)
25 I_1 = (I_a**2+U**2)**0.5       // Current(A)
26 PF_1 = I_a/I_1                // Lagging power
    factor
27 I_X_2 = (E_00**2+V**2)**0.5    // Reactance drop
    (V)
28 I_2 = I_X_2/X                 // Current
    corresponding to this drop(A)
29 PF_2 = E_00/I_X_2             // Leading power
    factor
30 P_max = V*I_2*PF_2/1000       // Maximum power
    output(kW)
31
32 // Results
33 disp("PART II – EXAMPLE : 11.4 : SOLUTION :-")
34 printf("\nNew value of machine current = %.1f A",
    I_1)
35 printf("\nNew value of power factor , p.f = %.4f (",
    lagging)", PF_1)

```

```

36 printf("\nPower output at which alternator break
      from synchronism = %.f kW", P_max)
37 printf("\nCurrent corresponding to maximum load = %.
      f A", I_2)
38 printf("\nPower factor corresponding to maximum load
      = %.4f (leading) \n", PF_2)
39 printf("\nNOTE: ERROR: Calculation mistakes in the
      textbook solution")

```

---

### Scilab code Exa 18.5 Phase angle between busbar sections

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 11: LOAD FREQUENCY CONTROL AND LOAD
     SHARING OF POWER GENERATING SOURCES
8
9 // EXAMPLE : 11.5 :
10 // Page number 333
11 clear ; clc ; close ; // Clear the work space and
     console
12
13 // Given data
14 V = 10000.0           // Voltage(V)
15 rating = 10000.0       // Full load rating (kW)
16 V_drop_per = 0.2       // Voltage drop of 20% for
     10000 kW
17
18 // Calculations
19 V_drop = V_drop_per*rating          //
     Voltage drop(V)
20 sin_theta_2 = (V_drop/2)/V          // Sin

```

```

        ( /2)
21 theta_2 = asind(sin_theta_2)           // 
        /2( )
22 theta = 2.0*theta_2                   // 
    Phase angle between busbar sections , ( )
23
24 // Results
25 disp("PART II - EXAMPLE : 11.5 : SOLUTION :-")
26 printf("\nPhase angle between busbar sections , = 
        %.2 f \n", theta)
27 printf("\nNOTE: ERROR: Calculation mistakes in the 
        textbook solution")

```

---

### Scilab code Exa 18.6 Voltage and Power factor at this latter station

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 11: LOAD FREQUENCY CONTROL AND LOAD
    SHARING OF POWER GENERATING SOURCES
8
9 // EXAMPLE : 11.6 :
10 // Page number 334
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 load_1 = 20000.0          // Total load (kW)
15 V = 11000.0               // Voltage(V)
16 PF_1 = 1.0                // Unity power factor
17 load_2 = 8000.0            // Load supplied (kW)
18 PF_2 = 0.8                // Lagging power factor

```

```

19 R = 0.5 // Resistance (ohm/phase)
20 X = 0.8 // Reactance (ohm/phase)
21
22 // Calculations
23 I_1 = load_1*1000/(3**0.5*V*PF_1) // Load current (
A)
24 I_2 = load_2*1000/(3**0.5*V*PF_2)*exp(%i*-acos(PF_2)
) // Current supplied by local
generators (A)
25 I_3 = I_1-I_2

// Current through interconnector (A)
26 angle_I_3 = phasemag(I_3) //
Current through interconnector leads reference
phasor by angle( )
27 V_drop = (R+%i*X)*I_3 //
Voltage drop across interconnector (V)
28 V_ph = V/3**0.5

// Phase voltage (V)
29 V_S = V_ph+V_drop

// Sending end voltage (V/phase)
30 V_S_11 = 3**0.5*V_S //
Sending end voltage (V)
31 angle_V_S_11 = phasemag(V_S_11) //
Angle of
sending end voltage( )
32 PF_S = cosd(angle_I_3-angle_V_S_11) //
Power factor at
sending station
33
34 // Results
35 disp("PART II - EXAMPLE : 11.6 : SOLUTION :-")

```

```

36 printf("\nVoltage at this latter station = %.f % .2
         f V (line-to-line)", abs(V_S_11), angle_V_S_11)
37 printf("\nPower factor at this latter station = %.4f
         (leading)", PF_S)

```

---

### Scilab code Exa 18.7 Load received Power factor and Phase difference between volta

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 11: LOAD FREQUENCY CONTROL AND LOAD
     SHARING OF POWER GENERATING SOURCES
8
9 // EXAMPLE : 11.7 :
10 // Page number 334
11 clear ; clc ; close ; // Clear the work space and
      console
12
13 // Given data
14 V = 33000.0          // Voltage(V)
15 R = 0.7              // Resistance(ohm/phase)
16 X = 3.5              // Reactance(ohm/phase)
17 load_1 = 60.0         // Load on generator at
      station X(MW)
18 PF_1 = 0.8            // Lagging power factor
19 load_2 = 40.0          // Local load taken by
      consumer (MW)
20 PF_2 = 0.707          // Lagging power factor
21
22 // Calculations
23 V_ph = V/3**0.5

```

```

        // Phase voltage(V)
24 I_1 = load_1*10**6/(3**0.5*V*PF_1)*exp(%i*-acos(PF_1
))                                // Load current on generator at X
(A)
25 I_2 = load_2*10**6/(3**0.5*V*PF_2)*exp(%i*-acos(PF_2
))                                // Current due to local load(A)
26 I_3 = I_1-I_2

        // Current through interconnector(A)
27 angle_I_3 = phasemag(I_3)          //
                                         Current through interconnector leads reference
                                         phasor by angle(   )
28 V_drop = (R+%i*X)*I_3            //
                                         Voltage drop across interconnector(V)
29 V_Y = V_ph-V_drop

        // Voltage at Y(V)
30 angle_V_Y = phasemag(V_Y)          //
                                         Angle of voltage at Y(   )
31 phase_diff = angle_I_3-angle_V_Y    // Phase
                                         difference b/w Y_Y and I_3(   )
32 PF_Y = cosd(phase_diff)           //
                                         Power factor of current received by Y
33 P_Y = 3*abs(V_Y*I_3)*PF_Y/1000.0  // Power
                                         received by station Y(kW)
34 phase_XY = abs(angle_V_Y)          //
                                         Phase angle b/w voltages of X & Y(   )
35
36 // Results
37 disp("PART II - EXAMPLE : 11.7 : SOLUTION :-")
38 printf("\nLoad received from station X to station Y

```

```

        = %.f kW" , P_Y)
39 printf("\nPower factor of load received by Y = %.4f
        (lagging)" , PF_Y)
40 printf("\nPhase difference between voltage of X & Y
        = %.2f (lagging) \n" , phase_XY)
41 printf("\nNOTE: Changes in the obtained answer from
        that of textbook is due to more precision here")

```

---

### Scilab code Exa 18.8 Percentage increase in voltage and Phase angle difference bet

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 11: LOAD FREQUENCY CONTROL AND LOAD
    SHARING OF POWER GENERATING SOURCES
8
9 // EXAMPLE : 11.8 :
10 // Page number 335
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V_tie = 11000.0          // Tie line Voltage(V)
15 Z = (3.5+%i*7.0)         // Impedance of tie line (ohm
    /conductor)
16 V = 6600.0                // Bus bar voltage(V)
17 Z_per = (2.5+%i*7.5)      // Percentage impedance on
    1000kVA rating
18 kVA = 2500.0              // Load received by other (
    kVA)
19
20 // Calculations

```

```

21 V_ph = V/3**0.5 // Phase
    voltage (V)
22 I_f1_LV = 100.0*V_tie/V_ph // LV
    side Full load current of each transformer(A)
23 R_eq = V_ph*real(Z_per)/(100*I_f1_LV) //
    Equivalent resistance of transformer(ohm/phase)
24 X_eq = 3.0*R_eq //
    Equivalent reactance of transformer(ohm/phase)
25 R_phase = real(Z)*(V/V_tie)**2 //
    Resistance of line per phase(ohm)
26 X_phase = imag(Z)*(V/V_tie)**2 //
    Resistance of line per phase(ohm)
27 R_total = 2.0*R_eq+R_phase // Total
    resistance per phase(ohm)
28 X_total = 2.0*X_eq+X_phase // Total
    resistance per phase(ohm)
29 Z_total = R_total+%i*X_total // Total
    impedance(ohm/phase)
30 I = kVA*1000/(3**0.5*V) // Load
    current (A)
31 V_drop = I*Z_total //
    Voltage drop per phase(V)
32 V_A = V_ph
33 V_AA = V_A+V_drop //
    Sending end voltage per phase(V)
34 V_increase = abs(V_AA)-V_A //
    Increase in voltage required(V/phase)
35 percentage_increase = V_increase/V_A*100 //
    Percentage increase required(%)
36 phase_diff = phasemag(V_AA) //
    Angle at which V_A & V_B are displaced( )
37
38 // Results
39 disp("PART II – EXAMPLE : 11.8 : SOLUTION :-")
40 printf("\nCase(a): Percentage increase in voltage =
    %.2f percent", percentage_increase)
41 printf("\nCase(b): Phase angle difference between
    the two busbar voltages = %.2f \n", phase_diff)

```

```
42 printf("\nNOTE: ERROR: Several calculation mistakes  
in the textbook")
```

---

### Scilab code Exa 18.9 Station power factors and Phase angle between two busbar volt

```
1 // A Texbook on POWER SYSTEM ENGINEERING  
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  
3 // DHANPAT RAI & Co.  
4 // SECOND EDITION  
5  
6 // PART II : TRANSMISSION AND DISTRIBUTION  
7 // CHAPTER 11: LOAD FREQUENCY CONTROL AND LOAD  
SHARING OF POWER GENERATING SOURCES  
8  
9 // EXAMPLE : 11.9 :  
10 // Page number 335–336  
11 clear ; clc ; close ; // Clear the work space and  
console  
12  
13 // Given data  
14 X = 2.80 // Combined reactance (ohm/phase  
)  
15 load_1 = 7000.0 // Consumer load at station A(  
kW)  
16 PF_1 = 0.9 // Lagging power factor  
17 V = 11000.0 // Voltage(V)  
18 load_2 = 10000.0 // Load supplied by station B(  
kW)  
19 PF_2 = 0.75 // Lagging power factor  
20  
21 // Calculations  
22 V_ph = V/3**0.5  
  
// Phase voltage(V)  
23 I_1 = load_1*10**3/(3**0.5*V*PF_1)*exp(%i*-acos(PF_1
```

```

        ))                                // Current at A due to local
        load(A)
24 I_2 = load_2*10**3/(3**0.5*V*PF_2)*exp(%i*-acos(PF_2
        ))                                // Current at B due to local
        load(A)
25 IA_X = 0.5*(load_1+load_2)*1000/(3**0.5*V)
                                         // Current(A)
26 Y_1 = 220.443/V_ph

        // Solved manually referring textbook
27 X_1 = (1-Y_1**2)**0.5
28 angle_1 = atand(Y_1/X_1)
                                         //
        Phasor lags by an angle( )
29 IA_Y = (6849.09119318-V_ph*X_1)/X
                                         // Current(
        A)
30 Y_X = IA_Y/IA_X
31 angle_2 = atand(Y_X)

        // Angle by which I_A lags behind V_A( )
32 PF_A = cosd(angle_2)

        // Power factor of station A
33 angle_3 = acosd(PF_2)+angle_1
                                         //
        Angle by which I_2 lags V_A( )
34 I_22 = load_2*10**3/(3**0.5*V*PF_2)*exp(%i*-angle_3*
        %pi/180)                            // Current(A)
35 I = 78.7295821622-%i*(IA_Y-177.942225747)
                                         // Current(A)
36 I_B = I_22-I

        // Current(A)
37 angle_4 = abs(phasemag(I_B))-angle_1
                                         // Angle by
        which I_B lags behind V_B( )
38 PF_B = cosd(angle_4)

```

```

        // Power factor of station B
39
40 // Results
41 disp("PART II - EXAMPLE : 11.9 : SOLUTION :-")
42 printf("\nPower factor of station A = %.4f (lagging)
      ", PF_A)
43 printf("\nPower factor of station B = %.4f (lagging)
      ", PF_B)
44 printf("\nPhase angle between two bus bar voltages =
      %.f (V_B lagging V_A)", angle_1)

```

---

### Scilab code Exa 18.10 Constants of the second feeder

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 11: LOAD FREQUENCY CONTROL AND LOAD
     SHARING OF POWER GENERATING SOURCES
8
9 // EXAMPLE : 11.10 :
10 // Page number 336
11 clear ; clc ; close ; // Clear the work space and
     console
12
13 // Given data
14 load_1 = 10000.0          // Total balanced load (kW)
15 V = 33000.0               // Voltage (V)
16 PF_1 = 0.8                // Lagging power factor
17 R = 1.6                   // Resistance of feeder (ohm/
     phase)
18 X = 2.5                  // Reactance of feeder (ohm/

```

```

    phase)
19 load_2 = 4460.0           // Load delivered by feeder (kW
)
20 PF_2 = 0.72                // Lagging power factor
21
22 // Calculations
23 I = load_1*1000/(3**0.5*V*PF_1)*exp(%i*-acos(PF_1))
    // Total line current(A)
24 I_1 = load_2*1000/(3**0.5*V*PF_2)*exp(%i*-acos(PF_2)
)      // Line current of first feeder(A)
25 I_2 = I-I_1
                    //
        Line current of first feeder(A)
26 Z_1 = complex(R,X)
                    //
        Impedance of first feeder(ohm)
27 Z_2 = I_1*Z_1/I_2
                    //
        Impedance of second feeder(ohm)
28
29 // Results
30 disp("PART II - EXAMPLE : 11.10 : SOLUTION :-")
31 printf("\nImpedance of second feeder , Z_2 = %.2 f %
.1 f ohm \n", abs(Z_2),phasemag(Z_2))
32 printf("\nNOTE: ERROR: Changes in the obtained
answer from that of textbook is due to wrong
values of substitution")

```

---

### Scilab code Exa 18.11 Necessary booster voltages

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5

```

```

6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 11: LOAD FREQUENCY CONTROL AND LOAD
    SHARING OF POWER GENERATING SOURCES
8
9 // EXAMPLE : 11.11 :
10 // Page number 337
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 P = 9.0                                // Load supplied from
    substation (MW)
15 V = 33000.0                             // Voltage(V)
16 PF_1 = 1.0                               // Unity power factor
17 Z_A = complex(2.0,8.0)                   // Impedance of circuit A(
    ohm)
18 Z_B = complex(4.0,4.0)                   // Impedance of circuit B(
    ohm)
19
20 // Calculations
21 V_ph = V/3**0.5                         //
    Voltage at receiving end per phase(V)
22 P_A = 1.0/3*P                            //
    Power supplied by line A(MW)
23 P_B = 2.0/3*P                            //
    Power supplied by line B(MW)
24 I_A = P_A*10**6/(3**0.5*V)              //
    Current through line A(A)
25 I_B = P_B*10**6/(3**0.5*V)              //
    Current through line B(A)
26 IA_ZA_drop = I_A*Z_A                     //
    I_A Z_A drop (V/phase)
27 IB_ZB_drop = I_B*Z_B                     //
    I_B Z_B drop (V/phase)
28 phase_boost = real(IB_ZB_drop)-real(IA_ZA_drop) //
    Voltage in phase boost (V/phase)
29 quad_boost = imag(IB_ZB_drop)-imag(IA_ZA_drop) //
    Voltage in quadrature boost (V/phase)

```

```

30 constant_P = V_ph+IA_ZA_drop //  

    Assumed that sending end voltage at P is kept  

    constant(V/phase)  

31  

32 // Results  

33 disp("PART II - EXAMPLE : 11.11 : SOLUTION :-")  

34 printf("\nVoltage in-phase boost = %.2f V/phase",  

    phase_boost)  

35 printf("\nVoltage in quadrature boost = %.f V/phase"  

    , quad_boost)

```

---

**Scilab code Exa 18.12 Load on C at two different conditions of load in A and B**

```

1 // A Texbook on POWER SYSTEM ENGINEERING  

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  

3 // DHANPAT RAI & Co.  

4 // SECOND EDITION  

5  

6 // PART II : TRANSMISSION AND DISTRIBUTION  

7 // CHAPTER 11: LOAD FREQUENCY CONTROL AND LOAD  

    SHARING OF POWER GENERATING SOURCES  

8  

9 // EXAMPLE : 11.12 :  

10 // Page number 337  

11 clear ; clc ; close ; // Clear the work space and  

    console  

12  

13 // Given data  

14 cap_A = 15000.0 // Capacity of station  

    A(kW)  

15 cap_B = 10000.0 // Capacity of station  

    B(kW)  

16 cap_C = 2000.0 // Capacity of station  

    C(kW)  

17 speed_reg_A = 2.4/100 // Speed regulation of

```

```

A
18 speed_reg_B = 3.2/100           // Speed regulation of
      B
19 slip_C = 4.5/100               // Full load slip
20 local_load_B_a = 10000.0       // Local load on
      station B(kW)
21 local_load_A_a = 0             // Local load on
      station A(kW)
22 local_load_both = 10000.0      // Local load on both
      station (kW)
23
24 // Calculations
25 // Case(a)
26 speed_A = speed_reg_A/cap_A
                           // % of
      speed drop for A
27 speed_C = slip_C/cap_C
                           // %
      of speed drop for C
28 speed_B = speed_reg_B/cap_B
                           // % of
      speed drop for B
29 X = local_load_B_a*speed_B/(speed_A+speed_B+speed_C)
                           // Load on C when local load of B
      is 10000 kW and A has no load (kW)
30 // Case(b)
31 Y = local_load_both*(speed_B-speed_A)/(speed_A+
      speed_B+speed_C) // Load on C when both station
      have local loads of 10000 kW(kW)
32
33 // Results
34 disp("PART II - EXAMPLE : 11.12 : SOLUTION :-")
35 printf("\nCase(a): Load on C when local load of B is
      10000 kW and A has no load , X = %.f kW" , X)
36 printf("\nCase(b): Load on C when both station have
      local loads of 10000 kW, Y = %.f kW" , Y)

```

---

### Scilab code Exa 18.13 Loss in the interconnector as a percentage of power received

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 11: LOAD FREQUENCY CONTROL AND LOAD
    SHARING OF POWER GENERATING SOURCES
8
9 // EXAMPLE : 11.13 :
10 // Page number 337–338
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 l = 20.0                      // Length of cable(km)
15 r = 0.248                      // Resistance(ohm/km)
16 x = 0.50*10**-3                // Inductance(H/m)
17 V_gen = 6600.0                 // Generation voltage(V)
18 f = 50.0                        // Frequency(Hz)
19 V = 33000.0                    // Transmission voltage(V)
20 rating = 10.0                  // Transformer rating(MVA)
21 loss_cu = 100.0                // Copper loss at full load
    (kW)
22 x_tr = 2.5/100                // Transformer reactance
23 load = 7.5                     // Load to be transmitted(
    MW)
24 PF = 0.71                      // Lagging power factor
25
26 // Calculations
27 R = l*r
```

```

        // Resistance of the cable(ohm)
28 I_fl = rating*10**6/(3**0.5*V)                                // Transformer
        current at full load(A)
29 R_eq = loss_cu*1000/(3*I_fl**2)                                 // Equivalent
        resistance per phase of transformer(ohm)
30 R_total_hv = R+2.0*R_eq                                         // Total
        resistance per conductor in terms of hv side(ohm)
31 X = 2.0*%pi*f*l*x                                              //
        Reactance of cable per conductor(ohm)
32 per_X_tr = V/3**0.5*x_tr/I_fl                                    // % reactance
        of transformer(ohm)
33 X_total_hv = X+2.0*per_X_tr                                     // Total
        reactance per conductor in terms of hv side(ohm)
34 I = load*10**6/(3**0.5*V*PF)                                     // Line
        current at receiving end(A)
35 IR = I*R_total_hv                                                 //
        IR drop(V)
36 IX = I*X_total_hv                                               //
        IX drop(V)
37 E_r = V/3**0.5

        // Phase voltage at station B(V)
38 cos_phi_r = PF
39 sin_phi_r = (1-PF**2)**0.5
40 E_s = ((E_r*cos_phi_r+IR)**2+(E_r*sin_phi_r+IX)**2) **0.5/1000 // Sending end voltage(kV)
41 E_s_ll = 3**0.5*E_s                                              //
        Sending end line voltage(kV)

```

```

42 V_booster = 3**0.5*(E_s-E_r/1000) // Booster voltage
    between lines (kV)
43 tan_phi_s = (E_r*sin_phi_r+IX)/(E_r*cos_phi_r+IR)
    // tan _s
44 phi_s = atand(tan_phi_s) // _s (
    )
45 cos_phi_s = cosd(phi_s) // cos _s
46 P_s = 3.0*E_s*I*cos_phi_s // Power at
    sending end (kW)
47 loss = P_s-load*1000 // Loss (kW)
48 loss_per = loss/(load*1000)*100 // loss
    percentage
49
50 // Results
51 disp("PART II - EXAMPLE : 11.13 : SOLUTION :-")
52 printf("\nLoss in the interconnector as a percentage
        of power received = %.3f percent", loss_per)
53 printf("\nRequired voltage of the booster = %.3f kV
        (in terms of H.V) \n", V_booster)
54 printf("\nNOTE: Changes in the obtained answer from
        that of textbook is due to more precision here")
55 printf("\n      kVA rating of booster is not
        calculated in textbook and here")

```

---

# Chapter 20

## WAVE PROPAGATION ON TRANSMISSION LINES

Scilab code Exa 20.4 Reflected and Transmitted wave of Voltage and Current at the

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 13: WAVE PROPAGATION ON TRANSMISSION
    LINES
8
9 // EXAMPLE : 13.4 :
10 // Page number 366
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 R_1 = 60.0      // Surge impedance of underground
    cable(ohm)
15 R_2 = 400.0     // Surge impedance of overhead line(
    ohm)
```

```

16 e = 100.0      // Maximum value of surge (kV)
17
18 // Calculations
19 i = e*1000/R_1           // Current (A)
20 k = (R_2-R_1)/(R_2+R_1)
21 e_ref = k*e             // Reflected voltage (
    kV)
22 e_trans = e+e_ref       // Transmitted voltage
    (kV)
23 e_trans_alt = (1+k)*e   // Transmitted voltage
    (kV). Alternative method
24 i_ref = -k*i            // Reflected current (A
    )
25 i_trans = e_trans*1000/R_2 // Transmitted current
    (A)
26 i_trans_alt = (1-k)*i   // Transmitted current
    (A). Alternative method
27
28 // Results
29 disp("PART II – EXAMPLE : 13.4 : SOLUTION :-")
30 printf("\nReflected voltage at the junction = %.f kV
    ", e_ref)
31 printf("\nTransmitted voltage at the junction = %.f
    kV", e_trans)
32 printf("\nReflected current at the junction = %.f A"
    , i_ref)
33 printf("\nTransmitted current at the junction = %.f
    A\n", i_trans)
34 printf("\nNOTE: ERROR: Calculation mistake in
    textbook in finding Reflected current")

```

---

### Scilab code Exa 20.5 First and Second voltages impressed on C

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar

```

```

3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 13: WAVE PROPAGATION ON TRANSMISSION
    LINES
8
9 // EXAMPLE : 13.5 :
10 // Page number 366
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 R_A = 500.0      // Surge impedance of line A(ohm)
15 R_B = 70.0       // Surge impedance of line B(ohm)
16 R_C = 600.0      // Surge impedance of line C(ohm)
17 e = 20.0         // Rectangular voltage wave(kV)
18
19 // Calculations
20 E_2 = e*(1+((R_B-R_A)/(R_B+R_A)))           //
    Transmitted wave(kV)
21 E_4 = E_2*((1+((R_C-R_B)/(R_C+R_B)))        //
    voltage impressed on C(kV))
22 E_3 = E_2*(R_C-R_B)/(R_C+R_B)                 //
    Reflected
    wave(kV)
23 E_5 = E_3*(R_A-R_B)/(R_A+R_B)                 //
    Reflected
    wave(kV)
24 E_6 = E_5*((1+((R_C-R_B)/(R_C+R_B)))        //
    Transmitted wave(kV))
25 second = E_4+E_6                                //
    Second
    voltage impressed on C(kV)
26
27 // Results
28 disp("PART II – EXAMPLE : 13.5 : SOLUTION :-")
29 printf("\nFirst voltage impressed on C = %.1f kV" ,
    E_4)
30 printf("\nSecond voltage impressed on C = %.1f kV" ,
    second)

```

---

### Scilab code Exa 20.6 Voltage and Current in the cable and Open wire lines

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 13: WAVE PROPAGATION ON TRANSMISSION
    LINES
8
9 // EXAMPLE : 13.6 :
10 // Page number 367
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 Z = 100.0          // Surge impedance of cable(ohm)
15 Z_1 = 600.0        // Surge impedance of open wire(
    ohm)
16 Z_2 = 1000.0       // Surge impedance of open wire(
    ohm)
17 e = 2.0            // Steep fronted voltage(kV)
18
19 // Calculations
20 Z_t = Z_1*Z_2/(Z_1+Z_2)      // Resultant surge
    impedance(ohm)
21 E = e*(1+((Z_t-Z)/(Z_t+Z))) // Transmitted voltage
    (kV)
22 I_1 = E*1000/Z_1           // Current(A)
23 I_2 = E*1000/Z_2           // Current(A)
24 E_ref = e*(Z_t-Z)/(Z_t+Z)  // Reflected voltage(
    kV)
25 I_ref = -E_ref*1000/Z      // Reflected current(A)
```

```
    )
26
27 // Results
28 disp("PART II – EXAMPLE : 13.6 : SOLUTION :-")
29 printf("\nVoltage in the cable = %.3f kV", E)
30 printf("\nCurrent in the cable , I_1 = %.2f A", I_1)
31 printf("\nCurrent in the cable , I_2 = %.3f A", I_2)
32 printf("\nVoltage in the open-wire lines i.e
        Reflected voltage = %.3f kV", E_ref)
33 printf("\nCurrent in the open-wire lines i.e
        Reflected current = %.2f A", I_ref)
```

---

# Chapter 21

## LIGHTNING AND PROTECTION AGAINST OVERVOLTAGES DUE TO LIGHTNING

Scilab code Exa 21.1 Ratio of voltages appearing at the end of a line when line is

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 14: LIGHTNING AND PROTECTION AGAINST
    OVERVOLTAGES DUE TO LIGHTNING
8
9 // EXAMPLE : 14.1 :
10 // Page number 382
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
```

```

14 RI_072 = 72000.0      // Characteristic of lightning
   arrester
15 Z_c = 500.0           // Surge impedance(ohm)
16 V = 500.0             // Surge voltage(kV)
17
18 // Calculations
19 // Case(a)
20 V_a = 2.0*V           // Voltage at the end of line
   at open-circuit(kV)
21 ratio_a = V_a/V       // Ratio of voltage when line
   is open-circuited
22 // Case(b)
23 I = V*1000/Z_c       // Surge current(A)
24 R = RI_072/(I)**0.72  // Resistance of LA(ohm)
25 ratio_b = R/Z_c       // Ratio of voltage when line
   is terminated by arrester
26
27 // Results
28 disp("PART II – EXAMPLE : 14.1 : SOLUTION :-")
29 printf("\nCase(a): Ratio of voltages appearing at
   the end of a line when line is open-circuited = %
   .f", ratio_a)
30 printf("\nCase(b): Ratio of voltages appearing at
   the end of a line when line is terminated by
   arrester = %.f", ratio_b)

```

---

### Scilab code Exa 21.2 Choosing suitable arrester rating

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 14: LIGHTNING AND PROTECTION AGAINST

```

## OVERVOLTAGES DUE TO LIGHTNING

```
8
9 // EXAMPLE : 14.2 :
10 // Page number 383
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 rating = 5000.0          // Rating of transformer (kVA)
15 V_hv = 66.0              // HV voltage (kV)
16 V_lv = 11.0              // LV voltage (kV)
17 V = 66.0                 // System voltage (kV)
18 fluctuation = 0.1        // Voltage fluctuations
19 BIL = 350.0               // BIL for 66kV(kV)
20 dynamic_ov = 1.3          // Dynamic over-voltage = 1.3*
   system operating voltage
21 V_power_freq = 1.5        // Power frequency breakdown
   voltage of arrester = 1.5*arrester rating (kV)
22 lower_limit = 0.05         // Margin of lower limit of
   arrester rating
23
24 // Calculation & Result
25 disp("PART II – EXAMPLE : 14.2 : SOLUTION :-")
26 V_rating = V*(1+fluctuation)*0.8*(1+lower_limit)
   // Voltage rating of arrester (kV)
27 if(round(V_rating)==51) then
28   V_rating_choosen = 50.0
   // Arrester
   rating choosen (kV)
29   V_discharge = 176.0
   //
   Discharge voltage for 50kV arrester (kV)
30   protective_margin = BIL-V_discharge
   // Protective margin
   available (kV)
31   V_power_frequency_bd = V_rating_choosen*
   V_power_freq // Power frequency breakdown
   voltage (kV)
```

```

32     Over_voltage_dynamic = dynamic_ov*V/3**0.5
33             // Dynamic overvoltage (kV)
34     if(V_power_frequency_bd>Over_voltage_dynamic)
35         then
36             printf("\nFirst arrester with rating 50 kV (
37             rms) & discharge voltage 176 kV chosen is
38             suitable")
39         end
40     elseif(round(V_rating)==61) then
41         V_rating_choosen = 60.0
42                     // Arrester
43                     rating choosen (kV)
44         V_discharge = 220.0
45                     //
46                     Discharge voltage for 50kV arrester(kV)
47         protective_margin = BIL-V_discharge
48                     // Protective margin
49                     available (kV)
50         V_power_frequency_bd = V_rating_choosen*
51             V_power_freq // Power frequency breakdown
52             voltage(kV)
53         Over_voltage_dynamic = dynamic_ov*V/3**0.5
54             // Dynamic overvoltage(kV)
55         if(V_power_frequency_bd>Over_voltage_dynamic)
56             then
57                 printf("\nSecond arrester with rating 60 kV
58                 (rms) & discharge voltage 220 kV chosen
59                 is suitable")
60             end
61     else(round(V_rating)==74) then
62         V_rating_choosen = 73.0
63                     // Arrester
64                     rating choosen (kV)
65         V_discharge = 264.0
66                     //
67                     Discharge voltage for 50kV arrester(kV)
68         protective_margin = BIL-V_discharge
69                     // Protective margin
70                     available (kV)

```

```
49     V_power_frequency_bd = V_rating_choosen*
          V_power_freq // Power frequency breakdown
          voltage(kV)
50     Over_voltage_dynamic = dynamic_ov*V/3**0.5
          // Dynamic overvoltage(kV)
51     if(V_power_frequency_bd>Over_voltage_dynamic)
          then
52         printf("\nThird arrester with rating 73 kV (
          rms) & discharge voltage 264 kV chosen is
          suitable")
53     end
54 end
```

---

# Chapter 22

## INSULATION COORDINATION

Scilab code Exa 22.1 Highest voltage to which the transformer is subjected

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 15: INSULATION CO-ORDINATION
8
9 // EXAMPLE : 15.1 :
10 // Page number 398–399
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 L = 30.0          // Height of arrester located (m)
15 BIL = 650.0       // BIL(kV)
16 de_dt = 1000.0    // Rate of rising surge wave front
    (kV/ -sec)
17 V = 132.0         // Transformer voltage at HV side(
```

```

        kV)
18 E_a = 400.0          // Discharge voltage of arrester(
        kV)
19 v = 3.0*10**8         // Velocity of surge propagation(m
        /sec)
20
21 // Calculations
22 E_t = E_a+(2.0*de_dt*L/300) // Highest voltage the
        transformer is subjected (kV)
23
24 // Results
25 disp("PART II - EXAMPLE : 15.1 : SOLUTION :-")
26 printf("\nHighest voltage to which the transformer
        is subjected , E_t = %.f kV" , E_t)

```

---

### Scilab code Exa 22.2 Rating of LA and Location with respect to transformer

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 15: INSULATION CO-ORDINATION
8
9 // EXAMPLE : 15.2 :
10 // Page number 399
11 clear ; clc ; close ; // Clear the work space and
        console
12
13 // Given data
14 V_hv = 132.0          // Voltage at the HV side of
        transformer(kV)
15 V_lv = 33.0           // Voltage at the LV side of
        transformer(kV)

```

```

16 V = 860.0          // Insulator allowable voltage (kV)
17 Z = 400.0          // Line surge impedance(ohm)
18 BIL = 550.0        // BIL(kV)
19
20 // Calculations
21 V_rating_LA = V_hv*1.1*0.8           //
   Voltage rating of LA(kV)
22 E_a = 351.0           //
   Discharge voltage at 5 kA(kV)
23 I_disc = (2*V-E_a)*1000/Z           //
   Discharge current(A)
24 L_1 = 37.7             //
   Separation distance in current b/w arrester tap
   and power transformer tap(m)
25 dist = 11.0              //
   length from tap point to ground level(m)
26 de_dt = 500.0            //
   Maximum rate of rise of surge(kV/ -sec)
27 Inductance = 1.2          //
   Inductance( H /metre)
28 di_dt = 5000.0           //
   A/ -sec)
29 lead_drop = Inductance*dist*di_dt/1000    //
   Drop
   in the lead(kV)
30 E_d = E_a+lead_drop         //
   (kV)
31 V_tr_terminal = E_d+2*de_dt*L_1/300       //
   Voltage at transformer terminals(kV)
32 E_t = BIL/1.2             //
   Highest voltage the transformer is subjected(kV)
33 L = (E_t-E_a)/(2*de_dt)*300           //
   Distance at which lightning arrester located from
   transformer(m)
34 L_lead = (E_t-E_a*1.1)/(2*de_dt)*300     //
   Distance at which lightning arrester located from
   transformer taken 10% lead drop(m)
35
36 // Results
37 disp("PART II – EXAMPLE : 15.2 : SOLUTION :-")

```

```
38 printf("\nRating of L.A = %.1f kV" , V_rating_LA)
39 printf("\nLocation of L.A, L = %.f m" , L)
40 printf("\nLocation of L.A if 10 percent lead drop is
        considered , L = %.1f m" , L_lead)
41 printf("\nMaximum distance at which a lightning
        arrester is usually connected from transformer is
        %.f-%.f m" , L-2,L+3)
```

---

# Chapter 23

## POWER SYSTEM GROUNDING

Scilab code Exa 23.1 Inductance and Rating of arc suppression coil

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 16: POWER SYSTEM GROUNDING
8
9 // EXAMPLE : 16.1 :
10 // Page number 409
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 V = 132.0*10**3          // Voltage(V)
15 n = 3.0                  // Number of phase
16 f = 50.0                 // Frequency(Hz)
17 l = 50.0                 // Line length(km)
18 C = 0.0157*10**-6       // Capacitance to earth(F/km)
```

```

19
20 // Calculations
21 L = 1/(n*(2*pi*f)**2*C*l)           // Inductance (H)
22 X_L = 2*pi*f*L                      // Reactance (ohm)
23 I_F = V/(3**0.5*X_L)                 // Current (A)
24 rating = I_F*V/(3**0.5*1000)         // Rating of arc
    suppression coil (kVA)

25
26 // Results
27 disp("PART II – EXAMPLE : 16.1 : SOLUTION :-")
28 printf("\nInductance , L = %.1f Henry", L)
29 printf("\nRating of arc suppression coil = %.f kVA \
    n", rating)
30 printf("\nNOTE: Changes in the obtained answer from
    that of textbook is due to more approximation in
    the textbook")

```

---

# Chapter 24

## ELECTRIC POWER SUPPLY SYSTEMS

Scilab code Exa 24.1 Weight of copper required for a three phase transmission system

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 17: ELECTRIC POWER SUPPLY SYSTEMS
8
9 // EXAMPLE : 17.1 :
10 // Page number 422–423
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 no_phase = 3.0          // Number of phases in ac
    transmission system
15 V = 380.0*10**3        // Voltage b/w lines (V)
16 load = 100.0            // Load (MW)
17 PF = 0.9                // Power factor
```

```

18 l = 150.0 // Line length (km)
19 n = 0.92 // Efficiency
20 r = 0.045 // Resistance (ohm/km/sq.cm)
21 w_cu_1 = 0.01 // Weight of 1 cm^3 copper(
    kg)

22
23 // Calculations
24 // Case(i)
25 P_loss = (1-n)*load // Power loss
    in the line(MW)
26 I_L = load*10**6/(3**0.5*V*PF) // Line current
    (A)
27 loss_cu = P_loss/no_phase*10**6 // I^2*R loss
    per conductor(W)
28 R = loss_cu/I_L**2 // Resistance
    per conductor(ohm)
29 R_km = R/l // Resistance
    per conductor per km(ohm)
30 area = r/R_km // Conductor
    area(Sq.cm)
31 volume = area*100.0 // Volume of
    copper per km run(cm^3)
32 W_cu_km = volume*w_cu_1 // Weight of
    copper per km run(kg)
33 W_cu = no_phase*l*1000*W_cu_km // Weight of
    copper for 3 conductors of 150 km(kg)
34 // Case(ii)
35 W_cu_dc = 1.0/2*PF**2*W_cu // Weight of
    copper conductor in dc(kg)

36
37 // Results
38 disp("PART II - EXAMPLE : 17.1 : SOLUTION :-")
39 printf("\nWeight of copper required for a three-
    phase transmission system = %.f kg", W_cu)
40 printf("\nWeight of copper required for the d-c
    transmission system = %.f kg \n", W_cu_dc)
41 printf("\nNOTE: Changes in the obtained answer from
    that of textbook is due to more precision")

```

---

### Scilab code Exa 24.2 Percentage increase in power transmitted

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 17: ELECTRIC POWER SUPPLY SYSTEMS
8
9 // EXAMPLE : 17.2 :
10 // Page number 423
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 P_1 = 1.0      // Assume P1 to be 1
15
16 // Calculations
17 P_2 = (3.0*2)**0.5          // 3-phase power
    transmitted in terms of P_1
18 inc_per = (P_2-P_1)/P_1*100 // Increase in power
    transmitted(%)
19
20 // Results
21 disp("PART II – EXAMPLE : 17.2 : SOLUTION :-")
22 printf("\nPercentage increase in power transmitted =
    %.f percent", inc_per)
```

---

### Scilab code Exa 24.3 Percentage additional balanced load

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 17: ELECTRIC POWER SUPPLY SYSTEMS
8
9 // EXAMPLE : 17.3 :
10 // Page number 424
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 PF = 0.95      // Lagging power factor
15
16 // Calculations
17 P_1 = 1.0          //
   Power in terms of V*I_1
18 P_2 = 2.0*PF**2          //
   Power in terms of V*I_1
19 P_additional_percentage = (P_2-P_1)/P_1*100    //
   Percentage additional power transmitted in a 3-
   phase 3-wire system
20
21 // Results
22 disp("PART II - EXAMPLE : 17.3 : SOLUTION :-")
23 printf("\nPercentage additional power transmitted in
   a 3-phase 3-wire system = %.f percent",
   P_additional_percentage)

```

---

**Scilab code Exa 24.4** Amount of copper required for 3 phase 4 wire system with that

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar

```

```

3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 17: ELECTRIC POWER SUPPLY SYSTEMS
8
9 // EXAMPLE : 17.4 :
10 // Page number 424–425
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 n = 3.0          // 3-phase 4 wire ac system
15
16 // Calculations
17 a2_a1 = 1.0/6           // Ratio of cross-sectional
   area of 2 wire dc to 3-phase 4-wire system
18 ratio_cu = 3.5/2*a2_a1 // Copper for 3 phase 4
   wire system to copper for 2 wire dc system
19
20 // Results
21 disp("PART II – EXAMPLE : 17.4 : SOLUTION :-")
22 printf("\nCopper for 3-phase 4-wire system/Copper
   for 2-wire dc system = %.3f : 1", ratio_cu)

```

---

**Scilab code Exa 24.5 Weight of copper required and Reduction of weight of copper p**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 17: ELECTRIC POWER SUPPLY SYSTEMS
8

```

```

9 // EXAMPLE : 17.5 :
10 // Page number 425
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 L = 60.0          // Line length (km)
15 P = 5.0           // Load (MW)
16 PF = 0.8          // Lagging power factor
17 V = 33.0*10***3  // Voltage (V)
18 n = 0.85          // Transmission efficiency
19 rho = 1.73*10**-8 // Specific resistance of copper
    (ohm-mt)
20 density = 8900.0  // Density (kg/mt^3)
21
22 // Calculations
23 I = P*10**6/(3**0.5*V*PF)          // Line
    current (A)
24 line_loss = (1-n)*P*1000/n          // Line loss
    (kW)
25 line_loss_phase = line_loss/3.0      // Line loss
    /phase (kW)
26 R = line_loss_phase*1000/I**2       //
    Resistance/phase(ohm)
27 a = rho*L*1000/R                   // Area of
    cross section of conductor(m^2)
28 volume = 3.0*a*L*1000              // Volume of
    copper(m^3)
29 W_cu = volume*density             // Weight of
    copper in 3-phase system(kg)
30 I_1 = P*10**6/V                  // Current
    in single phase system(A)
31 R_1 = line_loss*1000/(2*I_1**2)   //
    Resistance in single phase system(ohm)
32 a_1 = rho*L*1000/R_1              // Area of
    cross section of conductor in single phase system
    (m^2)
33 volume_1 = 2.0*a_1*L*1000        // Volume of

```

```

        copper(m^3)
34 W_cu_1 = volume_1*density           // Weight of
    copper in 1-phase system(kg)
35 reduction_cu = (W_cu-W_cu_1)/W_cu*100 // Reduction
    in copper(%)
36
37 // Results
38 disp("PART II - EXAMPLE : 17.5 : SOLUTION :-")
39 printf("\nWeight of copper required for 3-phase 2-
    wire system = %.2e kg", W_cu)
40 printf("\nReduction of weight of copper possible = %
    .1f percent \n", reduction_cu)
41 printf("\nNOTE: ERROR: Calculation mistakes in the
    textbook solution")

```

---

#### Scilab code Exa 24.6 Economical cross section of a 3 core distributor cable

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 17: ELECTRIC POWER SUPPLY SYSTEMS
8
9 // EXAMPLE : 17.6 :
10 // Page number 427-428
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 L = 250.0                  // Cable length(m)
15 P = 80.0*10**3              // Load(W)
16 V = 400.0                  // Voltage(V)
17 PF = 0.8                   // Lagging power factor

```

```

18 time = 4000.0 // Time of operation (hours
                  /annum)
19 a = poly(0, 'a') // Area of each conductor (
                     Sq.cm)
20 cost_instal = 15.0*a+25 // Cost of cable including
                           installation (Rs/m)
21 interest_per = 0.1 // Interest & depreciation
22 cost_waste_per = 0.1 // Cost of energy wasted (
                         Rs/unit)
23 r = 0.173 // Resistance per km of 1
              cm^2(ohm)
24
25 // Calculations
26 I = P/(3**0.5*V*PF) // Line
                          current (A)
27 energy_waste = 3.0*I**2*r/a*L*10**-3*time*10**-3
                  // Energy wasted per annum(kWh)
28 cost_energy_waste = cost_waste_per*energy_waste
                  // Annual cost of energy wasted as losses
                  (Rs)
29 capital_cost_cable = cost_instal*L
                  // Capital cost of cable(Rs)
30 annual_cost_cable = capital_cost_cable*
                      cost_waste_per // Annual cost on cable(Rs)
31 area = (1081.25/375)**0.5
                  // Area = a(Sq.cm).
                  Simplified and taken final answer
32
33 // Results
34 disp("PART II – EXAMPLE : 17.6 : SOLUTION :-")
35 printf("\nEconomical cross-section of a 3-core
          distributor cable , a = %.1f cm^2" , area)

```

---

**Scilab code Exa 24.7 Most economical cross section**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 17: ELECTRIC POWER SUPPLY SYSTEMS
8
9 // EXAMPLE : 17.7 :
10 // Page number 428
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 V = 110.0*10**3 // Voltage(V)
15 l_1 = 24.0*10**6 // Load(MW)
16 t_1 = 6.0 // Time(hours)
17 l_2 = 8.0*10**6 // Load(MW)
18 t_2 = 6.0 // Time(hours)
19 l_3 = 4.0*10**6 // Load(MW)
20 t_3 = 12.0 // Time(hours)
21 PF = 0.8 // Lagging power
   factor
22 a = poly(0, 'a')
   each conductor(Sq.cm) // Cross-section of
23 cost_line = 12000.0+8000*a // Cost of line
   including erection(Rs/km)
24 R = 0.19/a // Resistance per km
   of each conductor(ohm)
25 cost_energy = 8.0/100 // Energy cost(Rs/unit
   )
26 interest_per = 0.1 // Interest &
   depreciation. Assumption
27
28 // Calculations
29 annual_charge = interest_per*cost_line // Total
   annual charge(Rs)
30 I_1 = l_1/(3**0.5*V*PF) // Line

```

```

        current for load 1(A)
31 I_2 = l_2/(3**0.5*V*PF)           // Line
        current for load 2(A)
32 I_3 = l_3/(3**0.5*V*PF)           // Line
        current for load 3(A)
33 I_2_t = I_1**2*t_1+I_2**2*t_2+I_3**2*t_3    // I^2*t
34 annual_energy = 3.0*R*365/1000*I_2_t          // Annual
        energy consumption on account of losses (kWh)
35 cost_waste = annual_energy*cost_energy         // Cost
        of energy wasted per annum(Rs)
36 area = (2888.62809917355/800.0)**0.5          //
        Economical cross-section = a(Sq.cm). Simplified
        and taken final answer
37
38 // Results
39 disp("PART II - EXAMPLE : 17.7 : SOLUTION :-")
40 printf("\nMost economical cross-section , a = %.2f cm
^2", area)

```

---

### Scilab code Exa 24.8 Most economical current density for the transmission line

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 17: ELECTRIC POWER SUPPLY SYSTEMS
8
9 // EXAMPLE : 17.8 :
10 // Page number 428-429
11 clear ; clc ; close ; // Clear the work space and
        console
12
13 // Given data

```

```

14 cost_km_cu = 2800.0           // Cost per km for each
     copper conductor of sq.cm(Rs)
15 LF_I = 80.0/100              // Load factor of load
     current
16 LF_loss = 65.0/100            // Load factor of losses
17 interest_per = 10.0/100       // Rate of interest and
     depreciation
18 cost_energy = 5.0/100         // Cost of energy (Rs/kWh
     )
19 rho = 1.78*10**-8             // Resistivity (ohm-m)
20
21 // Calculations
22 P_2 = cost_km_cu*interest_per          // 
     Cost in terms of L(Rs)
23 time_year = 365.0*24                  //
     Total hours in a year
24 P_3 = cost_energy*rho*10**4*time_year*LF_loss // 
     Cost in terms of I^2 & L(Rs)
25 delta = (P_2/P_3)**0.5                //
     Economical current density for the transmission
     line(A/sq.cm)
26
27 // Results
28 disp("PART II - EXAMPLE : 17.8 : SOLUTION :-")
29 printf("\nMost economical current density for the
     transmission line ,      = %.f A/sq.cm" , delta)

```

---

### Scilab code Exa 24.9 Most economical cross section of the conductor

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION

```

```

7 // CHAPTER 17: ELECTRIC POWER SUPPLY SYSTEMS
8
9 // EXAMPLE : 17.9 :
10 // Page number 429
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 MD = 1000.0 // Maximum demand (kW)
15 energy_cons = 5.0*10**6 // Annual energy
    consumption (kWh)
16 PF = 0.85 // Power factor
17 capital_cost = 80000.0 // Capital cost of cable
    (Rs/km)
18 cost_energy = 5.0/100 // Energy cost (Rs/kWh)
19 interest_per = 10.0/100 // Rate of interest and
    depreciation
20 r_specific = 1.72*10**-6 // Specific resistance
    of copper (ohm/cubic.cm)
21 V = 11.0 // Voltage (kV)
22
23 // Calculations
24 I = MD/(3**0.5*V*PF) // Line current corresponding to maximum demand(A)
25 hours_year = 365.0*24 // Total hours in a year
26 LF = energy_cons/(MD*hours_year) // Load factor
27 loss_LF = 0.25*LF+0.75*LF**2 // Loss load
    factor
28 P_2 = capital_cost*interest_per // Cost in terms
    of L(Rs)
29 P_3 = 3.0*I**2*r_specific*10**4*hours_year*loss_LF*
    cost_energy // Cost in terms of I^2 & L(Rs)

```

```
30 a = (P_3/P_2)**0.5
                                //
      Most economical cross-section of conductor (sq.cm)
31
32 // Results
33 disp("PART II - EXAMPLE : 17.9 : SOLUTION :-")
34 printf("\nMost economical cross-section of the
         conductor , a = %.2f cm^2 \n", a)
35 printf("\nNOTE: ERROR: Calculation mistake in the
         textbook solution")
```

---

# Chapter 25

## POWER DISTRIBUTION SYSTEMS

Scilab code Exa 25.1 Potential of 0 and Current leaving each supply point

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 18: POWER DISTRIBUTION SYSTEMS
8
9 // EXAMPLE : 18.1 :
10 // Page number 437
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 V_A = 225.0      // Potential at point A(V)
15 R_A = 5.0        // Resistance of line A(ohm)
16 V_B = 210.0      // Potential at point B(V)
17 R_B = 1.0        // Resistance of line B(ohm)
18 V_C = 230.0      // Potential at point C(V)
```

```

19 R_C = 1.0      // Resistance of line C(ohm)
20 V_D = 230.0    // Potential at point D(V)
21 R_D = 2.0      // Resistance of line D(ohm)
22 V_E = 240.0    // Potential at point E(V)
23 R_E = 2.0      // Resistance of line E(ohm)
24
25 // Calculations
26 V_0 = ((V_A/R_A)+(V_B/R_B)+(V_C/R_C)+(V_D/R_D)+(V_E/
R_E))/((1/R_A)+(1/R_B)+(1/R_C)+(1/R_D)+(1/R_E))
// Potential at point O(V)
27 I_A = (V_A-V_0)/R_A      // Current leaving supply
point A(A)
28 I_B = (V_B-V_0)/R_B      // Current leaving supply
point B(A)
29 I_C = (V_C-V_0)/R_C      // Current leaving supply
point C(A)
30 I_D = (V_D-V_0)/R_D      // Current leaving supply
point D(A)
31 I_E = (V_E-V_0)/R_E      // Current leaving supply
point E(A)
32
33 // Results
34 disp("PART II – EXAMPLE : 18.1 : SOLUTION :-")
35 printf("\nPotential of point O, V_0 = %.f V", V_0)
36 printf("\nCurrent leaving supply point A, I_A = %.f
A", I_A)
37 printf("\nCurrent leaving supply point B, I_B = %.f
A", I_B)
38 printf("\nCurrent leaving supply point C, I_C = %.f
A", I_C)
39 printf("\nCurrent leaving supply point D, I_D = %.2f
A", I_D)
40 printf("\nCurrent leaving supply point E, I_E = %.2f
A", I_E)

```

---

### Scilab code Exa 25.2 Point of minimum potential along the track and Currents supplied

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 18: POWER DISTRIBUTION SYSTEMS
8
9 // EXAMPLE : 18.2 :
10 // Page number 437–438
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 I = 600.0          // Constant current drawn(A)
15 D = 8.0           // Distance b/w two sub-stations (
    km)
16 V_A = 575.0        // Potential at point A(V)
17 V_B = 590.0        // Potential at point B(V)
18 R = 0.04           // Track resistance (ohm/km)
19
20 // Calculations
21 x = poly(0, 'x')           // x(
    km)
22 I_A = ((-V_B+R*I*D+V_A)-(R*I)*x)/(D*R)           //
    Simplifying
23 V_P = V_A-I_A*R*x           // /
    Potential at P in terms of x(V)
24 dVP_dx = derivat(V_P)           // /
    dV_P/dx
25 x_sol = roots(dVP_dx)           // /
    Value of x(km)
26 I_A_1 = ((-V_B+R*I*D+V_A)-(R*I)*x_sol)/(D*R)           //
    Current drawn from end A(A)
27 I_B = I-I_A_1           // /
    Current drawn from end B(A)
```

```

28
29 // Results
30 disp("PART II - EXAMPLE : 18.2 : SOLUTION :-")
31 printf("\nPoint of minimum potential along the track
32 , x = %.2f km", x_sol)
33 printf("\nCurrent supplied by station A, I_A = %.f A
34 ", I_A_1)
33 printf("\nCurrent supplied by station B, I_B = %.f A
34 \n", I_B)
34 printf("\nNOTE: ERROR: Calculation mistake in the
34 textbook solution")

```

---

### Scilab code Exa 25.3 Position of lowest run lamp and its Voltage

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 18: POWER DISTRIBUTION SYSTEMS
8
9 // EXAMPLE : 18.3 :
10 // Page number 438-439
11 clear ; clc ; close ; // Clear the work space and
11 console
12
13 // Given data
14 l = 400.0          // Length of cable(m)
15 i = 1.0            // Load(A/m)
16 I_1 = 120.0         // Current at 40m from end A(A)
17 l_1 = 40.0          // Distance from end A(A)
18 I_2 = 72.0           // Current at 72m from end A(A)
19 l_2 = 120.0         // Distance from end A(A)
20 I_3 = 48.0           // Current at 200m from end A(A)

```

```

21 l_3 = 200.0          // Distance from end A(A)
22 I_4 = 120.0          // Current at 320m from end A(A)
23 l_4 = 320.0          // Distance from end A(A)
24 r = 0.15              // Cable resistance (ohm/km)
25 V_A = 250.0          // Voltage at end A(A)
26 V_B = 250.0          // Voltage at end A(A)
27
28 // Calculations
29 I = poly(0,"I")

        // Current from end A(A)
30 A_A1 = l_1*r*(I-(1.0/2)*i*l_1)                                // Drop
over length(V)
31 I_d_1 = 40.0

        // Distributed tapped off current(A)
32 I_A1_A2 = I-l_1-l_2

        // Current fed in over length(A)
33 A1_A2 = (l_2-l_1)*r*(I_A1_A2-(1.0/2)*i*(l_2-l_1))           // Drop over length(V)
34 I_d_2 = 80.0

        // Distributed tapped off current(A)
35 I_A2_A3 = I_A1_A2-(I_2+I_d_2)                                     // Current
fed in over length(A)
36 A2_A3 = (l_3-l_2)*r*(I_A2_A3-(1.0/2)*i*(l_3-l_2))           // Drop over length(V)
37 I_d_3 = 80.0

        // Distributed tapped off current(A)
38 I_A3_A4 = I_A2_A3-(I_3+I_d_3)                                     // Current
fed in over length(A)
39 A3_A4 = (l_4-l_3)*r*(I_A3_A4-(1.0/2)*i*(l_4-l_3))           // Drop over length(V)

```

```

40 I_d_4 = 120.0

        // Distributed tapped off current(A)
41 I_A4_B = I_A3_A4-(I_4+I_d_4)                                //
        Current fed in over length(A)
42 A4_B = (l-l_4)*r*(I_A4_B-(1.0/2)*i*(l-l_4))           // Drop over length(V)
43 V_drop = A_A1+A1_A2+A2_A3+A3_A4+A4_B                      // Total voltage
        drop in terms of I
44 I = roots(V_drop)

        // Current(A)
45 I_total = 760.0

        // Total load current(A)
46 I_B = I_total-I

        // Current from B(A)
47 A_A3 = 2.0*r/1000*(l_1*(I-20)+(l_2-l_1)*(I-200)+(l_3
        -l_2)*(I-352)) // Potential drop over length
        A_A3(V)
48 V_A3 = V_A-A_A3

        // Voltage at the lowest run lamp(V)
49
50 // Results
51 disp("PART II - EXAMPLE : 18.3 : SOLUTION :-")
52 printf("\nPosition of lowest-run lamp, A_3 = %.f m", l_3)
53 printf("\nVoltage at the lowest-run lamp = %.1f V", V_A3)

```

---

**Scilab code Exa 25.4 Point of minimum potential and its Potential**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 18: POWER DISTRIBUTION SYSTEMS
8
9 // EXAMPLE : 18.4 :
10 // Page number 439
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 l = 450.0           // Length of wire(m)
15 V_A = 250.0          // Voltage at end A(V)
16 V_B = 250.0          // Voltage at end A(V)
17 r = 0.05             // Conductor resistance (ohm/km)
18 i = 1.5              // Load(A/m)
19 I_C = 20.0            // Current at C(A)
20 l_C = 60.0            // Distance to C from A(m)
21 I_D = 40.0            // Current at D(A)
22 l_D = 100.0           // Distance to D from A(m)
23 l_E = 200.0           // Distance to E from A(m)
24
25 // Calculations
26 x = poly(0,"x")           //
   Current to point D from end A(A)
27 AD = (I_C+x)*r*l_C+x*r*(l_D-l_C)           //
   Drop in length AD
28 BD = (i*r*V_A**2/2)+(I_D-x)*r*(450-l_D)           //
   Drop in length BD
29 x_sol = roots(AD-BD)           //
   Current(A)
30 I_F = x_sol-I_D           //
   Current supplied to load from end A(A)
31 l_F = l_E+(I_F/i)           //
   Point of minimum potential at F from A(m)

```

```

32 V_F = V_B - (375.0 - I_F) * (250 - (l_F - 200)) * r / 1000      //
   Potential at F from end B(V)
33
34 // Results
35 disp("PART II - EXAMPLE : 18.4 : SOLUTION :-")
36 printf("\nPoint of minimum potential occurs at F
   from A = %.2f metres", l_F)
37 printf("\nPotential at point F = %.2f V", V_F)

```

---

**Scilab code Exa 25.6 Ratio of weight of copper with and without interconnector**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 18: POWER DISTRIBUTION SYSTEMS
8
9 // EXAMPLE : 18.6 :
10 // Page number 440–441
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 l_AB = 100.0          // Length between A & B(m)
15 l_BC = 150.0          // Length between B & C(m)
16 l_CD = 200.0          // Length between C & D(m)
17 l_AD = 350.0          // Length between A & D(m)
18 l_AE = 200.0          // Length between A & E(m)
19 l_ED = 250.0          // Length between E & D(m)
20 I_B = 10.0            // Current at B(A)
21 I_C = 20.0            // Current at C(A)
22 I_D = 50.0            // Current at D(A)
23 I_E = 39.0            // Current at E(A)

```

```

24
25 // Calculations
26 x = poly(0 , "x")

    // Current in section AB(A)
27 ABCDEA = x*l_AB+(x-I_B)*l_BC+(x-I_B-I_C)*l_CD+(x-I_B
    -I_C-I_D)*l_ED+(x-I_B-I_C-I_D-I_E)*l_AE // KVL
    around loop ABCDEA
28 x_sol = roots(ABCDEA)

    //
    Current in section AB(A)
29 V_AD = x_sol*l_AB+(x_sol-I_B)*l_BC+(x_sol-I_B-I_C)*
    l_CD // Voltage drop from A to D in
    terms of /a_1(V)
30 R_AD = (l_AB+l_BC+l_CD)*(l_AE+l_ED)/(l_AB+l_BC+l_CD+
    l_AE+l_ED) // Resistance of n/w across
    terminals AD in terms of /a
31 I_AD = V_AD/(R_AD+l_AD)

    //
    Current in interconnector AD(A)
32 V_A_D = I_AD*l_AD

    // Voltage drop between A & D in terms of /a_2
33 a2_a1 = V_A_D/V_AD
34 length_with = (l_AB+l_BC+l_CD+l_AE+l_ED+l_AD)
    // Length of conductor with
    interconnector(m)
35 length_without = (l_AB+l_BC+l_CD+l_AE+l_ED)
    // Length of conductor
    without interconnector(m)
36 volume_with = a2_a1*length_with/length_without
    // Weight of copper with
    interconnector

37
38 // Results
39 disp("PART II - EXAMPLE : 18.6 : SOLUTION :-")
40 printf("\nRatio of weight of copper with & without
    interconnector = %.3f : 1 (or) 1 : %.2f",

```

```
volume_with ,1/volume_with)
```

---

### Scilab code Exa 25.7 Potential difference at each load point

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 18: POWER DISTRIBUTION SYSTEMS
8
9 // EXAMPLE : 18.7 :
10 // Page number 441–442
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 r_out = 0.05           // Resistance of each outer
    per 100 metre length(ohm)
15 r_neutral = 0.10       // Resistance of each
    neutral per 100 metre length(ohm)
16 V_A = 200.0            // Potential at point A(V)
17 V_B = 200.0            // Potential at point B(V)
18 l_AC = 100.0           // Length between A & C(m)
19 l_CD = 150.0           // Length between C & D(m)
20 l_DB = 200.0           // Length between D & B(m)
21 l_AF = 200.0           // Length between A & F(m)
22 l_FE = 100.0           // Length between F & E(m)
23 l_EB = 150.0           // Length between E & B(m)
24 I_C = 20.0              // Current at point C(A)
25 I_D = 30.0              // Current at point D(A)
26 I_F = 60.0              // Current at point F(A)
27 I_E = 40.0              // Current at point E(A)
28
```

```

29 // Calculations
30 x = poly(0,"x")

    // Current in positive outer alone(A)
31 equ_1 = r_out*(l_DB*(I_D-x))-r_out*(l_AC*(I_C+x) +
    l_CD*x)
32 x_sol = roots(equ_1)
    //

        Current in positive outer alone(A)
33 y = poly(0,"y")

    // Current in negative outer alone(A)
34 equ_2 = r_out*((I_E-y)*l_FE+(I_E+I_F-y)*l_AF)-r_out *
    *(l_EB*y)
35 y_sol = roots(equ_2)
    //

        Current in negative outer alone(A)
36 I_pos_out = I_C+x_sol
    //

        Current entering positive outer(A)
37 I_neg_out = I_E+I_F-y_sol
    //

        Current returning via negative outer(A)
38 I_middle = I_neg_out-I_pos_out
    // Current in
        the middle wire towards G(A)
39 r_CD = r_out*l_CD/100.0
    //

        Resistance between C & D(ohm)
40 r_D = r_out*l_DB/100.0
    //

        Resistance between D & B(ohm)
41 r_IH = r_neutral*l_FE*0.5/100.0
    // Resistance
        between I & H(ohm)
42 r_IJ = r_neutral*l_FE*0.5/100.0
    // Resistance
        between I & J(ohm)

```

```

43 r_GH = r_neutral*l_AF*0.5/100.0
        // Resistance
        between G & H(ohm)
44 r_AF = r_out*l_AF/100.0
        //
        Resistance between A & F(ohm)
45 I_CD = x_sol
        //
        Current flowing into D from C(A)
46 I_out_D = I_D-x_sol
        //
        Current flowing into D from outer side(A)
47 I_GH = I_C+I_middle
        //
        Current flowing into H from G(A)
48 I_IH = I_F-I_GH
        //
        Current flowing into H from I(A)
49 I_BJ = I_E-(I_D-I_IH)
        //
        Current flowing into J from B(A)
50 I_FE = y_sol-I_E
        //
        Current flowing into E from F(A)
51 I_IJ = I_D-I_IH
        //
        Current flowing into J from I(A)
52 V_C = V_A-(I_pos_out*r_out-I_middle*r_neutral)
        //
        Potential at load point C(A)
53 V_D = V_C-(I_CD*r_CD+I_IH*r_IH-I_GH*r_GH)
        //
        Potential at load
        point D(A)
54 V_F = V_A-(I_middle*r_neutral+I_GH*r_neutral+
        I_neg_out*r_AF)      //
        Potential at load point F
        (A)
55 V_E = V_F-(-I_IH*r_IH+I_IJ*r_IJ-I_FE*r_out)
        //
        Potential at load point

```

E(A)

```
56
57 // Results
58 disp("PART II - EXAMPLE : 18.7 : SOLUTION :-")
59 printf("\nPotential difference at load point C = %.3
       f V", V_C)
60 printf("\nPotential difference at load point D = %.3
       f V", V_D)
61 printf("\nPotential difference at load point E = %.3
       f V", V_E)
62 printf("\nPotential difference at load point F = %.3
       f V", V_F)
```

---

**Scilab code Exa 25.8** Load on the main generators and On each balancer machine

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 18: POWER DISTRIBUTION SYSTEMS
8
9 // EXAMPLE : 18.8 :
10 // Page number 442-443
11 clear ; clc ; close ; // Clear the work space and
                           console
12
13 // Given data
14 V = 440.0           // Voltage between outer (V)
15 I_pos = 210.0       // Ligting load current on
                         positive side (A)
16 I_neg = 337.0       // Ligting load current on
                         negative side (A)
17 I_power = 400.0     // Power load current (A)
```

```

18 P_loss = 1.5           // Loss in each balancer
    machine (kW)
19
20 // Calculations
21 P = I_power*V/1000.0      //
    Power (kW)
22 load_pos = I_pos*V*0.5/1000.0      //
    Load on positive side (kW)
23 load_neg = I_neg*V*0.5/1000.0      //
    Load on negative side (kW)
24 loss_total = 2*P_loss      //
    Total loss on rotary balancer set (kW)
25 load_main = P+load_pos+load_neg+loss_total      //
    Load on main machine (kW)
26 I = load_main*1000/V      //
    Current (A)
27 I_M = I-610.0      //
    Current through balancer machine (A)
28 I_G = 127.0-I_M      //
    Current through generator (A)
29 output_G = I_G*V*0.5/1000.0      //
    Output of generator (kW)
30 input_M = I_M*V*0.5/1000.0      //
    Input to balancer machine (kW)
31
32 // Results
33 disp("PART II - EXAMPLE : 18.8 : SOLUTION :-")
34 printf("\nLoad on the main machine = %.2f kW",
        load_main)
35 printf("\nOutput of generator = %.2f kW", output_G)
36 printf("\nInput to balancer machine = %.2f kW",
        input_M)

```

---

**Scilab code Exa 25.9 Currents in various sections and Voltage at load point C**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART II : TRANSMISSION AND DISTRIBUTION
7 // CHAPTER 18: POWER DISTRIBUTION SYSTEMS
8
9 // EXAMPLE : 18.9 :
10 // Page number 444
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 V_a = 11.0*10**3           // Line voltage at A(V)
15 Z_AB = complex(1.0,0.8)    // Impedance between A
   & B(ohm)
16 Z_AC = complex(3.0,2.0)    // Impedance between A
   & C(ohm)
17 Z_BD = complex(3.0,4.0)    // Impedance between B
   & D(ohm)
18 Z_CD = complex(1.0,0.7)    // Impedance between C
   & D(ohm)
19 I_B = 60.0                 // Current at B(A)
20 I_C = 30.0                 // Current at C(A)
21 I_D = 50.0                 // Current at D(A)
22 pf_B = 0.8                 // Power factor at B
23 pf_C = 0.9                 // Power factor at C
24 pf_D = 0.707               // Power factor at D
25
26 // Calculations
27 sin_phi_B = (1-pf_B**2)**0.5
28 I_B1 = I_B*(pf_B-%i*sin_phi_B) // Load current(
   A)
29 sin_phi_C = (1-pf_C**2)**0.5
30 I_C1 = I_C*(pf_C-%i*sin_phi_C) // Load current(
   A)
31 sin_phi_D = (1-pf_D**2)**0.5

```

```

32 I_D1 = I_D*(pf_D-%i*sin_phi_D)           // Load current (A)
33 V_A = V_a/3**0.5                          // Phase voltage at A(V)
34 I_AC = I_C1                                // Current in section AC when C & D is removed(A)
35 I_BD = I_D1                                // Current in section BD when C & D is removed(A)
36 I_AB = I_B1+I_D1                           // Current in section AB when C & D is removed(A)
37 V_AC_drop = I_AC*Z_AC                      // Voltage drop at section AC(V)
38 V_AB_drop = I_AB*Z_AB                      // Voltage drop at section AB(V)
39 V_BD_drop = I_BD*Z_BD                      // Voltage drop at section BD(V)
40 V_drop_D = V_BD_drop+V_AB_drop             // Total drop upto D(V)
41 pd_CD = V_drop_D-V_AC_drop                // Potential difference between C & D(V)
42 Z_CD = Z_AB+Z_BD+Z_AC                      // Impedance of network looking from terminal C & D(ohm)
43 I_CD = pd_CD/(Z_CD+Z_CD)                  // Current flowing in section CD(A)
44 I_AC = I_CD+I_C1                           // Current flowing in section AC(A)
45 I_BD = I_D1-I_CD                          // Current flowing in section BD(A)
46 I_AB = I_BD+I_B1                           // Current flowing in section AB(A)
47 V_drop_AC = I_AC*Z_AC                      // Drop caused by current flowing in section AC(V/phase)
48 V_drop_AC_line = V_drop_AC*3**0.5          // Drop caused by current flowing in section AC(V)
49 V_C = V_a-V_drop_AC_line                 // Voltage at C(V)
50
51 // Results

```

```
52 disp("PART II – EXAMPLE : 18.9 : SOLUTION :-")
53 printf("\nCurrent in section CD, I_CD = (%.2f%.2fj )
      A", real(I_CD), imag(I_CD))
54 printf("\nCurrent in section AC, I_AC = (%.2f%.2fj )
      A", real(I_AC), imag(I_AC))
55 printf("\nCurrent in section BD, I_BD = (%.2f%.2fj )
      A", real(I_BD), imag(I_BD))
56 printf("\nCurrent in section AB, I_AB = (%.2f%.2fj )
      A", real(I_AB), imag(I_AB))
57 printf("\nVoltage at load point C = %.2 f % .2 f     kV
      ", abs(V_C)/1000, phasemag(V_C))
```

---

# Chapter 27

## SYMMETRICAL SHORT CIRCUIT CAPACITY CALCULATIONS

Scilab code Exa 27.1 Per unit current

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 1: SYMMETRICAL SHORT CIRCUIT CAPACITY
    CALCULATIONS
8
9 // EXAMPLE : 1.1 :
10 // Page number 466–467
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 500.0          // Generator voltage(V)
15 rating = 10.0      // Rating of the
```

```

        generator (kVA)
16 n_up = 1.0/2           // Turns ratio of step-
    up transformer
17 Z_line = complex(1.0,2.0) // Transmission line
    impedance(ohm)
18 n_down = 10.0/1         // Turns ratio of step-
    down transformer
19 load = complex(2.0,4.0)  // Load(ohm)
20
21 // Calculations
22 V_base_gen = V          // Base voltage(V)
23 kVA_base_gen = rating   // Base rating(kVA)
24 I_base_gen = kVA_base_gen*1000/V_base_gen // Base current(A)
25 Z_base_gen = V_base_gen/I_base_gen          // Base impedance(ohm)
26 V_base_line = V_base_gen/n_up               // Voltage base of the transmission line(V)
27 kVA_base_line = rating                     // Base rating of transmission line(kVA)
28 I_base_line = kVA_base_line*1000/V_base_line // Base current of transmission line(A)
29 Z_base_line = V_base_line/I_base_line       // Base impedance of transmission line(ohm)
30 Z_line_1 = Z_line/Z_base_line               // Impedance of transmission line(p.u)
31 V_base_load = V_base_line/n_down            // Base voltage at the load(V)
32 kVA_base_load = rating                     // Base rating of load(kVA)
33 I_base_load = kVA_base_load*1000/V_base_load // Base current of load(A)
34 Z_base_load = V_base_load/I_base_load       // Base impedance of load(ohm)
35 Z_load = load/Z_base_load                  // Load impedance(p.u)

```

```

36 Z_total = Z_line_1+Z_load //  

    Total impedance(p.u)  

37 I = 1.0/Z_total //  

    Current(p.u)  

38  

39 // Results  

40 disp("PART III - EXAMPLE : 1.1 : SOLUTION :-")  

41 printf("\nCurrent , I = %.3 f % .2 f p.u" , abs(I) ,  

    phasemag(I))

```

---

**Scilab code Exa 27.2 kVA at a short circuit fault between phases at the HV terminal**

```

1 // A Texbook on POWER SYSTEM ENGINEERING  

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  

3 // DHANPAT RAI & Co.  

4 // SECOND EDITION  

5  

6 // PART III : SWITCHGEAR AND PROTECTION  

7 // CHAPTER 1: SYMMETRICAL SHORT CIRCUIT CAPACITY  

    CALCULATIONS  

8  

9 // EXAMPLE : 1.2 :  

10 // Page number 467-468  

11 clear ; clc ; close ; // Clear the work space and  

    console  

12  

13 // Given data  

14 kV = 33.0 // Transmission line  

    operating voltage(kV)  

15 R = 5.0 // Transmission line  

    resistance(ohm)  

16 X = 20.0 // Transmission line  

    reactance(ohm)  

17 kVA_tr = 5000.0 // Rating of step-up  

    transformer(kVA)

```

```

18 X_tr = 6.0 // Reactance of
               transformer(%)
19 kVA_A = 10000.0 // Rating of alternator
                    A(kVA)
20 X_A = 10.0 // Reactance of
               alternator A(%)
21 kVA_B = 5000.0 // Rating of alternator
                    B(kVA)
22 X_B = 7.5 // Reactance of
               alternator B(%)
23
24 // Calculations
25 kVA_base = kVA_A // Base
                     rating(kVA)
26 X_gen_A = X_A*kVA_base/kVA_A // Reactance of
                     generator A(%)
27 X_gen_B = X_B*kVA_base/kVA_B // Reactance of
                     generator B(%)
28 X_trans = X_tr*kVA_base/kVA_tr // Reactance of
                     transformer(%)
29 X_per = kVA_base*X/(10*kV**2) // X(%)
30 R_per = kVA_base*R/(10*kV**2) // R(%)
31 Z_F1 = (X_gen_A*X_gen_B/(X_gen_A+X_gen_B))+X_trans
         // Impedance upto fault(%)
32 kVA_F1 = kVA_base*(100/Z_F1) // Short-circuit kVA fed
                     into the fault(kVA)
33 R_per_F2 = R_per // R(%)
34 X_per_F2 = X_per+Z_F1 // X(%)
35 Z_F2 = (R_per_F2**2+X_per_F2**2)**0.5

```

```

                // Total impedance upto F2(%)
36 kVA_F2 = kVA_base*(100/Z_F2)                                // Short-circuit kVA fed
                                                               into the fault at F2(kVA)

37
38 // Results
39 disp("PART III - EXAMPLE : 1.2 : SOLUTION :-")
40 printf("\nCase(a): kVA at a short-circuit fault
           between phases at the HV terminal of transformers
           = %.f kVA", kVA_F1)
41 printf("\nCase(b): kVA at a short-circuit fault
           between phases at load end of transmission line =
           %.f kVA \n", kVA_F2)
42 printf("\nNOTE: Changes in the obtained answer from
           that of textbook is due to more precision here &
           approximation in textbook")

```

---

### Scilab code Exa 27.3 Transient short circuit current and Sustained short circuit current

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 1: SYMMETRICAL SHORT CIRCUIT CAPACITY
CALCULATIONS
8
9 // EXAMPLE : 1.3 :
10 // Page number 468–469
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 kVA_a = 40000.0          // Capacity of transmission

```

```

        line (kVA)
15 x_a = 10.0
      transmission line (%)

16 kVA_b = 20000.0
      line (kVA)

17 x_b = 5.0
      transmission line (%)

18 kVA_c = 50000.0
      line (kVA)

19 x_c = 20.0
      transmission line (%)

20 kVA_d = 30000.0
      line (kVA)

21 x_d = 15.0
      transmission line (%)

22 kVA_e = 10000.0
      line (kVA)

23 x_e = 6.0
      transmission line (%)

24 kVA_T1 = 150000.0
      kVA)

25 x_T1 = 10.0
      (%)

26 kVA_T2 = 50000.0
      kVA)

27 x_T2 = 8.0
      (%)

28 kVA_T3 = 20000.0
      kVA)

29 x_T3 = 5.0
      (%)

30 kVA_GA = 150000.0
      kVA)

31 x_sA = 90.0
      generator (%)

32 x_tA = 30.0
      generator (%)

33 kVA_GB = 50000.0
      // Reactance of
      // Capacity of transmission
      // Reactance of
      // Capacity of transformer(
      // Reactance of transformer
      // Capacity of transformer(
      // Reactance of transformer
      // Capacity of transformer(
      // Reactance of transformer
      // Capacity of generator(
      // Synchronous reactance of
      // Transient reactance of
      // Capacity of generator(

```

```

        kVA)
34 x_sB = 50.0           // Synchronous reactance of
                           generator(%)
35 x_tB = 17.5           // Transient reactance of
                           generator(%)
36 V = 33.0              // Feeder voltage(kV)
37
38 // Calculations
39 kVA_base = 200000.0   // Base rating(
                           kVA)
40 X_a = kVA_base/kVA_a*x_a // Reactance(%)
41 X_b = kVA_base/kVA_b*x_b // Reactance(%)
42 X_c = kVA_base/kVA_c*x_c // Reactance(%)
43 X_d = kVA_base/kVA_d*x_d // Reactance(%)
44 X_e = kVA_base/kVA_e*x_e // Reactance(%)
45 X_T1 = kVA_base/kVA_T1*x_T1 // Reactance(%)
46 X_T2 = kVA_base/kVA_T2*x_T2 // Reactance(%)
47 X_T3 = kVA_base/kVA_T3*x_T3 // Reactance(%)
48 X_sA = kVA_base/kVA_GA*x_sA // Synchronous reactance
                           (%)
49 X_tA = kVA_base/kVA_GA*x_tA // Transient reactance(%)
                           )
50 X_sB = kVA_base/kVA_GB*x_sB // Synchronous reactance
                           (%)
51 X_tB = kVA_base/kVA_GB*x_tB // Transient reactance(%)

```

```

)
52 X_eq_ab = X_a+X_b                                // Equivalent
   reactance of transmission lines a & b(%)
53 X_eq_abc = X_eq_ab*X_c/(X_eq_ab+X_c)           // Equivalent reactance of
   transmission line c with series combination of a
   & b(%)
54 X_CF = (X_eq_abc+X_sA)*X_d/(X_eq_abc+X_sA+X_d) // Total reactance b/w sub-station C & F(%)
55 // Case(i)
56 X_tr_genA = kVA_base/kVA_GA*x_tA                // Reactance in transient
   state of generator A(%)
57 X_T1_tr = kVA_base/kVA_T1*x_T1                  // Reactance in transient
   state of transformer T1(%)
58 X_CF_tr = X_CF                                    // Total
   reactance in transient state b/w sub-station C &
   F(%)
59 X_eq_tAF = X_tr_genA+X_T1_tr+X_CF_tr           // Equivalent transient reactance
   from generator A to substation F(%)
60 X_tr_genB = kVA_base/kVA_GB*x_tB                // Reactance in transient
   state of generator B(%)
61 X_T2_tr = kVA_base/kVA_T2*x_T2                  // Reactance in transient
   state of transformer T2(%)
62 X_eq_tBF = X_tr_genB+X_T2_tr                    // Equivalent transient
   reactance from generator B to substation F(%)
63 X_eq_tF = X_eq_tAF*X_eq_tBF/(X_eq_tAF+X_eq_tBF) // Equivalent transient reactance upto
   substation F(%)
64 X_eq_tfault = X_eq_tF+X_T3                      // Equivalent transient

```

```

        reactance upto fault point(%)
65 kVA_t_sc = kVA_base/X_eq_tfault*100
                // Transient short circuit kVA(
                kVA)
66 I_t_sc = kVA_t_sc/(3**0.5*V)
                // Transient short
                circuit rms current(A)
67 I_t_sc_peak = 2**0.5*I_t_sc
                // Peak value of
                transient short circuit current(A)
68 // Case(ii)
69 X_S_genA = kVA_base/kVA_GA*x_sA
                // Reactance in steady state
                of generator A(%)
70 X_eq_SAF = X_S_genA+X_T1+X_CF
                // Equivalent steady state
                reactance from generator A to substation F(%)
71 X_eq_SBF = X_sB+X_T2
                // Equivalent
                steady state reactance from generator B to
                substation F(%)
72 X_eq_SF = X_eq_SAF*X_eq_SBF/(X_eq_SAF+X_eq_SBF)
                // Equivalent steady state reactance upto
                substation F(%)
73 X_eq_Sfault = X_eq_SF+X_T3
                // Equivalent steady
                state reactance upto fault point(%)
74 kVA_S_sc = kVA_base/X_eq_Sfault*100
                // Steady state short circuit
                kVA(kVA)
75 I_S_sc = kVA_S_sc/(3**0.5*V)
                // Sustained short
                circuit rms current(A)
76 I_S_sc_peak = 2**0.5*I_S_sc
                // Peak value of
                sustained short circuit current(A)
77
78 // Results

```

```

79 disp("PART III – EXAMPLE : 1.3 : SOLUTION :-")
80 printf("\nCase(i) : Transient short circuit current
     at X = %.f A (peak value)", I_t_sc_peak)
81 printf("\nCase(ii) : Sustained short circuit current
     at X = %.f A (peak value) \n", I_S_sc_peak)
82 printf("\nNOTE: Changes in the obtained answer from
     that of textbook is due to more precision here")

```

---

### Scilab code Exa 27.4 Current in the short circuit

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 1: SYMMETRICAL SHORT CIRCUIT CAPACITY
    CALCULATIONS
8
9 // EXAMPLE : 1.4 :
10 // Page number 469–470
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 kVA_gen = 21000.0          // Generator rating (kVA)
15 kV_gen = 13.8              // Voltage rating of
    generator (kV)
16 X_tr_gen = 30.0            // Transient reactance of
    generator (%)
17 kVA_trans = 7000.0          // Transformer rating (kVA)
18 kV_trans_lv = 13.8          // LV voltage rating of
    transformer (kV)
19 kV_trans_hv = 66.0          // HV voltage rating of
    transformer (kV)

```

```

20 X_trans = 8.4          // Reactance of transformer(%)
21 l = 50.0               // Tie line length(miles)
22 x = 0.848              // Reactance of tie line(ohm/mile)
23 l_fault = 20.0         // Location of fault from station A(miles)
24
25 // Calculations
26 kVA_base = kVA_gen     // Base rating(kVA)
27 X_A = X_tr_gen         // Reactance of generator A(%)
28 X_B = X_tr_gen         // Reactance of generator B(%)
29 X_T1 = 3.0*X_trans     // Reactance of transformer T1(%)
30 X_T2 = 3.0*X_trans     // Reactance of transformer T2(%)
31 X_1 = kVA_base/(10*kV_trans_hv**2)*x*l_fault // Reactance(%)
32 X_2 = X_1*(l-l_fault)/l_fault // Reactance(%)
33 X_AF = X_A+X_T1+X_1 // Resultant reactance A to F(%)
34 X_BF = X_B+X_T2+X_2 // Resultant reactance B to F(%)
35 X_eq_fault = X_AF*X_BF/(X_AF+X_BF) // Equivalent reactance upto fault(%)
36 kVA_SC = kVA_base/X_eq_fault*100 // Short circuit kVA((kVA))
37 I_SC = kVA_SC/(3**0.5*kV_trans_hv) // Short circuit current(A)
38
39 // Results
40 disp("PART III - EXAMPLE : 1.4 : SOLUTION :-")
41 printf("\nShort circuit current = %.f A \n", I_SC)
42 printf("\nNOTE: Changes in the obtained answer from

```

that of textbook is due to more precision here")

---

### Scilab code Exa 27.5 Per unit values of the single line diagram

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 1: SYMMETRICAL SHORT CIRCUIT CAPACITY
    CALCULATIONS
8
9 // EXAMPLE : 1.5 :
10 // Page number 470-471
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 MVA_G1 = 100.0          // Generator rating (MVA)
15 X_G1 = 30.0             // Reactance of generator (
    %)
16 MVA_G2 = 150.0          // Generator rating (MVA)
17 X_G2 = 20.0             // Reactance of generator (
    %)
18 MVA_G3 = 200.0          // Generator rating (MVA)
19 X_G3 = 15.0             // Reactance of generator (
    %)
20 MVA_T1 = 150.0          // Transformer rating (MVA)
21 X_T1 = 10.0              // Reactance of
    transformer(%)
22 MVA_T2 = 175.0          // Transformer rating (MVA)
23 X_T2 = 8.0               // Reactance of
    transformer(%)
24 MVA_T3 = 200.0          // Transformer rating (MVA)
```

```

25 X_T3 = 6.0 // Reactance of
               transformer(%)
26 MVA_T4 = 100.0 // Transformer rating (MVA)
27 X_T4 = 5.0 // Reactance of
               transformer(%)
28 MVA_T5 = 150.0 // Transformer rating (MVA)
29 X_T5 = 5.0 // Reactance of
               transformer(%)
30 Z_L1 = complex(0.5,1.0) // Line impedance (ohm/km)
31 L1 = 100.0 // Line length (km)
32 Z_L2 = complex(0.4,1.2) // Line impedance (ohm/km)
33 L2 = 50.0 // Line length (km)
34 Z_L3 = complex(0.4,1.2) // Line impedance (ohm/km)
35 L3 = 50.0 // Line length (km)
36 Z_L4 = complex(0.3,1.0) // Line impedance (ohm/km)
37 L4 = 60.0 // Line length (km)
38 kV_L1 = 220.0 // Voltage towards line (kV
                  )
39 kV_L2 = 220.0 // Voltage towards line (kV
                  )
40 kV_L3 = 132.0 // Voltage towards line (kV
                  )
41 kV_L4 = 132.0 // Voltage towards line (kV
                  )
42
43 // Calculations
44 MVA_base = 200.0 // Base
                     rating (MVA)
45 X_d_G1 = (MVA_base/MVA_G1)*(X_G1/100) // 
               Reactance of generator (p.u)
46 X_d_G2 = (MVA_base/MVA_G2)*(X_G2/100) // 
               Reactance of generator (p.u)
47 X_d_G3 = (MVA_base/MVA_G3)*(X_G3/100) // 
               Reactance of generator (p.u)
48 X_T_1 = (MVA_base/MVA_T1)*(X_T1/100) // 
               Reactance of transformer (p.u)
49 X_T_2 = (MVA_base/MVA_T2)*(X_T2/100) // 
               Reactance of transformer (p.u)

```

```

50 X_T_3 = (MVA_base/MVA_T3)*(X_T3/100) //  

    Reactance of transformer(p.u)  

51 X_T_4 = (MVA_base/MVA_T4)*(X_T4/100) //  

    Reactance of transformer(p.u)  

52 X_T_5 = (MVA_base/MVA_T5)*(X_T5/100) //  

    Reactance of transformer(p.u)  

53 Z_L1_base = kV_L1**2/MVA_base // L1 base  

    impedance(ohm)  

54 Z_L_1 = Z_L1*L1/Z_L1_base // Line  

    impedance(p.u)  

55 Z_L2_base = kV_L2**2/MVA_base // L2 base  

    impedance(ohm)  

56 Z_L_2 = Z_L2*L2/Z_L2_base // Line  

    impedance(p.u)  

57 Z_L3_base = kV_L3**2/MVA_base // L3 base  

    impedance(ohm)  

58 Z_L_3 = Z_L3*L3/Z_L3_base // Line  

    impedance(p.u)  

59 Z_L4_base = kV_L4**2/MVA_base // L4 base  

    impedance(ohm)  

60 Z_L_4 = Z_L4*L4/Z_L4_base // Line  

    impedance(p.u)  

61  

62 // Results  

63 disp("PART III - EXAMPLE : 1.5 : SOLUTION :-")  

64 printf("\nnp.u values of the single line diagram are  

as below")  

65 printf("\nGenerators p.u reactances :")  

66 printf("\n X_d_G1 = %.1f p.u", X_d_G1)  

67 printf("\n X_d_G2 = %.3f p.u", X_d_G2)  

68 printf("\n X_d_G3 = %.2f p.u", X_d_G3)  

69 printf("\nTransformers p.u reactances :")  

70 printf("\n X_T1 = %.3f p.u", X_T_1)  

71 printf("\n X_T2 = %.4f p.u", X_T_2)  

72 printf("\n X_T3 = %.2f p.u", X_T_3)  

73 printf("\n X_T4 = %.1f p.u", X_T_4)  

74 printf("\n X_T5 = %.3f p.u", X_T_5)  

75 printf("\nLines p.u impedances :")

```

```

76 printf("\n Z_L1 = (%.3f + %.3fj) p.u" , real(Z_L_1),
    imag(Z_L_1))
77 printf("\n Z_L2 = (%.3f + %.3fj) p.u" , real(Z_L_2),
    imag(Z_L_2))
78 printf("\n Z_L3 = (%.3f + %.3fj) p.u" , real(Z_L_3),
    imag(Z_L_3))
79 printf("\n Z_L4 = (%.3f + %.3fj) p.u \n" , real(Z_L_4)
    ),imag(Z_L_4))
80 printf("\nNOTE: ERROR: (1). Reactance of T2 is 8
percent & not 1 percent as mentioned in the
textbook problem statement")
81 printf("\n          (2). Several calculation
mistakes in the textbook")

```

---

### Scilab code Exa 27.6 Actual fault current using per unit method

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 1: SYMMETRICAL SHORT CIRCUIT CAPACITY
CALCULATIONS
8
9 // EXAMPLE : 1.6 :
10 // Page number 471
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 kVA_gen = 21000.0           // Generator rating (kVA)
15 kV_gen = 13.8               // Voltage rating of
generator (kV)
16 X_tr_gen = 30.0             // Transient reactance of

```

```

        generator (%)
17 kVA_trans = 7000.0          // Transformer rating (kVA)
18 kV_trans_lv = 13.8          // LV voltage rating of
      transformer (kV)
19 kV_trans_hv = 66.0          // HV voltage rating of
      transformer (kV)
20 X_trans = 8.4              // Reactance of transformer(
      %)
21 l = 50.0                  // Tie line length (miles)
22 x = 0.848                 // Reactance of tie line (ohm
      /mile)
23 l_fault = 20.0            // Location of fault from
      station A (miles)
24
25 // Calculations
26 kVA_base = kVA_gen
                                // Base
      rating (kVA)
27 kV_base_lv = kV_trans_lv
                                // Base voltage on
      L.V side (kV)
28 kV_base_hv = kV_trans_hv
                                // Base voltage on
      H.V side (kV)
29 Z_gen_pu = %i*X_tr_gen/100
                                // Impedance of
      generator (p.u)
30 Z_trans_pu = %i*X_trans*3/100
                                // Impedance of
      transformer (p.u)
31 Z_F_left = %i*x*l_fault*kVA_base/(kV_base_hv
      **2*1000) // Impedance of line to left of fault
      F (p.u)
32 Z_F_right = Z_F_left*(l-l_fault)/l_fault
                                // Impedance of line to right of
      fault (p.u)
33 Z_AF = Z_gen_pu+Z_trans_pu+Z_F_left
                                // Impedance (p.u)

```

```

34 Z_BF = Z_gen_pu+Z_trans_pu+Z_F_right
           // Impedance(p.u)
35 Z_eq = Z_AF*Z_BF/(Z_AF+Z_BF)
           // Equivalent impedance
           (p.u)
36 I_F = 1.0/abs(Z_eq)
           // Fault
           current(p.u)
37 I_base = kVA_base/(3**0.5*kV_base_hv)
           // Base current(A)
38 I_F_actual = I_F*I_base
           // Actual fault
           current(A)
39
40 // Results
41 disp("PART III – EXAMPLE : 1.6 : SOLUTION :-")
42 printf("\nActual fault current = %.f A \n",
        I_F_actual)
43 printf("\nNOTE: Changes in the obtained answer from
        that of textbook is due to more precision here")

```

---

### Scilab code Exa 27.7 Sub transient fault current

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 1: SYMMETRICAL SHORT CIRCUIT CAPACITY
    CALCULATIONS
8
9 // EXAMPLE : 1.7 :
10 // Page number 471–472
11 clear ; clc ; close ; // Clear the work space and

```

```

        console
12
13 // Given data
14 MVA_G1 = 50.0          // Generator rating (MVA)
15 kV_G1 = 15.0           // Voltage rating of generator
16 (kV)
17 X_G1 = 0.2             // Reactance of generator (p.u)
18 MVA_G2 = 25.0           // Generator rating (MVA)
19 kV_G2 = 15.0           // Voltage rating of generator
20 (kV)
21 X_T = 0.1               // Reactance of transformer (p.
22 u)                      // Reactance of transformer (p.
23 kV_fault = 66.0          // Voltage at fault occurrence (
24 kV)                      // Voltage at fault occurrence (
25 kv_base = 69.0           // Base voltage (kV)
26 MVA_base = 100.0          // Base MVA
27
28 // Calculations
29 X_d_G1 = X_G1*MVA_base/MVA_G1           // Sub-
30 transient reactance referred to 100 MVA(p.u)
31 E_G1 = kV_fault/kv_base                  // Voltage
32 (p.u)
33 X_d_G2 = X_G2*MVA_base/MVA_G2           // Sub-
34 transient reactance referred to 100 MVA(p.u)
35 E_G2 = kV_fault/kv_base                  // Voltage
36 (p.u)
37 X_net = X_d_G1*X_d_G2/(X_d_G1+X_d_G2)    // Net sub
38 -transient reactance (p.u)
39 E_g = (E_G1+E_G2)/2                     // Net
40 voltage (p.u). NOTE: Not sure how this comes
41 I_fault = E_g/(%i*(X_net+X_T))           // Sub-
42 transient fault current (p.u)
43
44 // Results
45 disp("PART III - EXAMPLE : 1.7 : SOLUTION :-")

```

```

37 printf("\nSub-transient fault current = %.3f j p.u \n",
38   , imag(I_fault))
39 printf("\nNOTE: Changes in the obtained answer from
40       that of textbook is due to more precision here")

```

---

### Scilab code Exa 27.8 Voltage behind the respective reactances

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 1: SYMMETRICAL SHORT CIRCUIT CAPACITY
    CALCULATIONS
8
9 // EXAMPLE : 1.8 :
10 // Page number 472
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 X_d_st = 0.2          // Sub-transient reactance(p.u)
15 X_d_t = 0.4           // Transient reactance(p.u)
16 X_d = 1.0             // Direct axis reactance(p.u)
17 I_pu = 1.0            // Load current(p.u)
18 PF = 0.80             // Lagging power factor
19
20 // Calculations
21 V = 1.0               // Terminal voltage(p.u)
22 sin_phi = (1-PF**2)**0.5
23 I = I_pu*(PF-%i*sin_phi) // Load current(p.u)
24 E_st = V+%i*I*X_d_st   // Voltage behind sub-
    transient reactance(p.u)
25 E_t = V+%i*I*X_d_t     // Voltage behind

```

```

        transient reactance(p.u)
26 E = V + %i * I * X_d           // Voltage behind direct
      axis reactance(p.u)
27
28 // Results
29 disp("PART III - EXAMPLE : 1.8 : SOLUTION :-")
30 printf("\nVoltage behind sub-transient reactance = %.
         .2 f % .2 f    p.u", abs(E_st), phasemag(E_st))
31 printf("\nVoltage behind transient reactance = %.2
         f % .2 f    p.u", abs(E_t), phasemag(E_t))
32 printf("\nVoltage behind direct axis reactance , E =
         %.2 f % .2 f    p.u", abs(E), phasemag(E))

```

---

### Scilab code Exa 27.9 Initial symmetrical rms current in the hv side and lv side

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 1: SYMMETRICAL SHORT CIRCUIT CAPACITY
      CALCULATIONS
8
9 // EXAMPLE : 1.9 :
10 // Page number 472
11 clear ; clc ; close ; // Clear the work space and
      console
12
13 // Given data
14 kVA_G = 7500.0          // Generator rating(kVA)
15 kV_G = 6.9               // Voltage rating of
      generator(kV)
16 X_d_st = 9.0/100         // Sub-transient reactance of
      generator

```

```

17 X_d_t = 15.0/100           // Transient reactance of
     generator
18 X_d = 100.0                // Synchronous reactance of
     generator(%)
19 kVA_T = 7500.0              // Transformer rating (kVA)
20 kV_T_delta = 6.9            // Voltage rating of
     transformer delta side(kV)
21 kV_T_wye = 115.0             // Voltage rating of
     transformer wye side(kV)
22 X = 10.0/100                // Transformer reactance
23
24 // Calculations
25 I_base_ht = kVA_T/(3**0.5*kV_T_wye)      // Base
     current at ht side(A)
26 I_base_lt = kVA_T/(3**0.5*kV_T_delta)      // Base
     current at lt side(A)
27 I_f_st = 1.0/(%i*(X_d_st+X))               // Sub-
     transient current after fault(p.u)
28 I_f_ht = abs(I_f_st)*I_base_ht              // Initial
     fault current in h.t side(A)
29 I_f_lt = abs(I_f_st)*I_base_lt              // Initial
     fault current in l.t side(A)
30
31 // Results
32 disp("PART III - EXAMPLE : 1.9 : SOLUTION :-")
33 printf("\nInitial symmetrical rms current in the h.v
     side = %.f A", I_f_ht)
34 printf("\nInitial symmetrical rms current in the l.v
     side = %.f A \n", I_f_lt)
35 printf("\nNOTE: Changes in the obtained answer from
     that of textbook is due to more precision here")

```

---

**Scilab code Exa 27.10 Initial symmetrical rms current at the generator terminal**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
```

```

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 1: SYMMETRICAL SHORT CIRCUIT CAPACITY
    CALCULATIONS
8
9 // EXAMPLE : 1.10 :
10 // Page number 472
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 kVA_alt = 625.0          // Alternator rating (kVA)
15 V_alt = 480.0            // Voltage rating of
    alternator (V)
16 load = 500.0              // Load (kW)
17 V_load = 480.0            // Load voltage (V)
18 X_st = 8.0/100           // Sub-transient reactance
19
20 // Calculations
21 kVA_base = 625.0          // Base kVA
22 V_base = 480.0             // Base voltage (V)
23 I_load = load/kVA_base    // Load cuurent (A)
24 V = 1.0                   // Terminal voltage (p.u)
25 E_st = V+%i*I_load*X_st // Sub-transient voltage
    (p.u)
26 I_st = E_st/(%i*X_st)     // Sub-transient current
    (p.u)
27
28 // Results
29 disp("PART III - EXAMPLE : 1.10 : SOLUTION :-")
30 printf("\nInitial symmetrical rms current at the
    generator terminal = (%.1f%.1fj) p.u", real(I_st)
    , imag(I_st))

```

---

**Scilab code Exa 27.11 Sub transient current in the fault in generator and Motor**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 1: SYMMETRICAL SHORT CIRCUIT CAPACITY
    CALCULATIONS
8
9 // EXAMPLE : 1.11 :
10 // Page number 472–473
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 X_d_st_G = 0.15          // Sub-transient reactance of
    generator(p.u)
15 X_d_st_M = 0.45          // Sub-transient reactance of
    motor(p.u)
16 X = 0.10                 // Leakage reactance of
    transformer(p.u)
17 V = 0.9                  // Terminal voltage of the
    generator(p.u)
18 I_G = 1.0                 // Output current of the
    generator(p.u)
19 PF = 0.8                  // Power factor of the load
20
21 // Calculations
22 sin_phi = (1-PF**2)**0.5
23 I = I_G*(PF+%i*sin_phi)           // Load current(p
    .u)
24 E_st_G = V+%i*I*X_d_st_G           // Sub-transient
```

```

        voltage of the generator(p.u)
25 E_st_M = V-%i*I*X_d_st_M           // Sub-transient
        voltage of the motor(p.u)
26 I_st_g = E_st_G/(%i*(X_d_st_G+X)) // Sub-transient
        current in the generator at fault(p.u)
27 I_st_m = E_st_M/(%i*(X_d_st_M-X)) // Sub-transient
        current in the motor at fault(p.u)
28
29 // Results
30 disp("PART III - EXAMPLE : 1.11 : SOLUTION :-")
31 printf("\nCase(a): Sub-transient current in the
        fault in generator = %.3 f % .3 f    p.u", abs(
        I_st_g),phasemag(I_st_g))
32 printf("\nCase(b): Sub-transient current in the
        fault in motor = %.3 f % .2 f    p.u \n", abs(
        I_st_m),180+phasemag(I_st_m))
33 printf("\nNOTE: ERROR: Sub-transient reactance of
        motor is 0.45 p.u & not 0.35 p.u as mentioned in
        textbook statement")

```

---

### Scilab code Exa 27.12 Sub transient fault current Fault current rating of generator

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 1: SYMMETRICAL SHORT CIRCUIT CAPACITY
CALCULATIONS
8
9 // EXAMPLE : 1.12 :
10 // Page number 473-474
11 clear ; clc ; close ; // Clear the work space and
console

```

```

12
13 // Given data
14 kVA_G = 625.0 // Generator rating (kVA)
15 V_G = 2.4 // Voltage rating of
16 generator (kV)
17 X_st_G = 8.0/100 // Sub-transient reactance
18 of generator
19 rating_M = 250.0 // Motor rating (HP)
20 V_M = 2.4 // Voltage rating of motor(
21 kV)
22 n = 90.0/100 // Efficiency of motor
23 X_st_M = 20.0/100 // Sub-transient reactance
24 of motor
25
26 // Calculations
27 kVA_base = 625.0 // Base kVA
28 input_M = rating_M*0.746/n // Each motor input(
29 kVA)
30 X_st_m_pu = X_st_M*kVA_base/input_M // Sub-transient reactance of
31 motor (p.u)
32 I_base = kVA_base/(3**0.5*V_M) // Base current (A)
33 Z_th = %i*X_st_m_pu/3*X_st_G/(X_st_m_pu/3+X_st_G) // Thevenin impedance (p.u)
34 I_st = 1.0/Z_th // Initial
35 symmetrical current at F(p.u)
36 I_st_g = I_st*(X_st_m_pu/3/(X_st_m_pu/3+X_st_G)) // Fault current rating of generator breaker
37 (p.u)
38 I_st_m = (I_st-I_st_g)/3 // Fault current
39 rating of each motor breaker (p.u)
40
41 // Results

```

```
33 disp("PART III – EXAMPLE : 1.12 : SOLUTION :-")
34 printf("\nSub-transient fault current at F = %.2 f j p
     .u" , imag(I_st))
35 printf("\nFault current rating of generator breaker
     = %.1 f j p.u" , imag(I_st_g))
36 printf("\nFault current rating of each motor breaker
     = %.2 f j p.u" , imag(I_st_m))
```

---

# Chapter 28

## FAULT LIMITING REACTORS

Scilab code Exa 28.1 Reactance necessary to protect the switchgear

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 2: FAULT LIMITING REACTORS
8
9 // EXAMPLE : 2.1 :
10 // Page number 479–480
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 kVA_A = 2500.0           // Rating of alternator A(
   kVA)
15 x_A = 8.0                // Reactance of alternator
   A(%)
16 kVA_B = 5000.0           // Rating of alternator B(
```

```

        kVA)
17 x_B = 6.0                      // Reactance of alternator
        B(%)
18 kVA_CB = 150000.0               // Rating of circuit
        breaker(kVA)
19 kVA_T = 10000.0                 // Rating of transformer(
        kVA)
20 x_T = 7.5                      // Reactance of transformer
        (%)
21 V = 3300.0                     // System voltage(V)
22
23 // Calculations
24 kVA_base = 10000.0              //
        Base kVA
25 X_A = kVA_base/kVA_A*x_A       //
        Reactance of generator A(%)
26 X_B = kVA_base/kVA_B*x_B       //
        Reactance of generator B(%)
27 X_eq = X_A*X_B/(X_A+X_B)       //
        Combined reactance of A & B(%)
28 kVA_SC_G = kVA_base/X_eq*100    //
        Short-circuit kVA due to generators(kVA)
29 kVA_SC_T = kVA_base/x_T*100     //
        Short-circuit kVA due to grid supply(kVA)
30 X = (kVA_base*100/(kVA_CB-kVA_SC_G))-x_T   //
        Reactance necessary to protect switchgear(%)
31 I_f1 = kVA_base*1000/(3**0.5*V)             //
        Full load current corresponding to 10000 kVA(A)
32 X_phase = X*V/(3**0.5*I_f1*100)            //
        Actual value of reactance per phase(ohm)
33
34 // Results
35 disp("PART III - EXAMPLE : 2.1 : SOLUTION :-")
36 printf("\nReactance necessary to protect the
        switchgear = %.3f ohm/phase", X_phase)

```

---

**Scilab code Exa 28.2 kVA developed under short circuit when reactors are in circuit**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 2: FAULT LIMITING REACTORS
8
9 // EXAMPLE : 2.2 :
10 // Page number 480
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 X = 10.0          // Reactance of reactor(%)
15 kVA = 30000.0    // Rating of generator(kVA)
16 X_sc = 20.0       // Short-circuit reactance(%)
17
18 // Calculations
19 X_1 = 1.0/3*(X_sc+X)           // Combined reactance
    of generator A,B,C & associated reactors(%)
20 X_2 = X_1+X                  // Combined reactance
    upto fault(%)
21 X_total_a = X_2/2.0           // Total reactance
    upto fault(%)
22 kVA_SC_a = 100/X_total_a*kVA // Short-circuit kVA(
    kVA)
23 X_total_b = 1.0/4*X_sc        // Total reactance
    upto fault when E,F,G & H are short-circuited(%)
24 kVA_SC_b = 100/X_total_b*kVA // Short-circuit kVA(
    kVA)
25
```

```

26 // Results
27 disp("PART III - EXAMPLE : 2.2 : SOLUTION :-")
28 printf("\nCase(a): kVA developed under short-circuit
           when reactors are in circuit = %.f kVA",
           kVA_SC_a)
29 printf("\nCase(b): kVA developed under short-circuit
           when reactors are short-circuited = %.f kVA",
           kVA_SC_b)

```

---

#### Scilab code Exa 28.4 Reactance of each reactor

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 2: FAULT LIMITING REACTORS
8
9 // EXAMPLE : 2.4 :
10 // Page number 481
11 clear ; clc ; close ; // Clear the work space and
                           console
12
13 // Given data
14 kVA = 20000.0      // Rating of generator (kVA)
15 f = 50.0            // Frequency (Hz)
16 V = 11.0*10***3    // Voltage of generator (V)
17 X_G = 20.0          // Generator short-circuit reactance
                           (%)
18 x = 60.0            // Reactance falls to 60% normal
                           value
19
20 // Calculations
21 kVA_base = 20000.0

```

```

// Base kVA
22 X = poly(0,"X")
// Reactance of each reactors E,F,G & H(%)
23 X_AE = X+X_G
// Reactances of A & E in series (%)
24 X_BF = X+X_G
// Reactances of B & F in series (%)
25 X_CD = X+X_G
// Reactances of C & D in series (%)
26 X_eq = X_AE/3
// X_eq
= X_AE*X_BF*X_CD/(X_BF*X_CD+X_AE*X_CD+X_AE*X_BF)
. Combined reactances of 3 groups in parallel(%)
27 X_f = X_eq+X
// Reactances of these groups to fault via tie-bar(%)
)
28 X_sol = roots(6.66666666666667-(100-x)/100*(X_f))
// Value of reactance of each reactors E,F,
G & H(%)
29 I_f1 = kVA_base*1000/(3**0.5*V)
// Full load current
corresponding to 20000 kVA & 11 kV(A)
30 X_ohm = X_sol*V/(3**0.5*100*I_f1)
// Ohmic value of reactance
X(ohm)
31
32 // Results
33 disp("PART III - EXAMPLE : 2.4 : SOLUTION :-")
34 printf("\nReactance of each reactor = %.4f ohm \n",
X_ohm)
35 printf("\nNOTE: Changes in the obtained answer from
that of textbook is due to more precision here")

```

---

**Scilab code Exa 28.5 Instantaneous symmetrical short circuit MVA for a fault at X**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 2: FAULT LIMITING REACTORS
8
9 // EXAMPLE : 2.5 :
10 // Page number 481-482
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 kVA_base = 10000.0      // Base kVA
15 V = 6.6*10***3          // Voltage of generator(V)
16 X_A = 7.5                // Reactance of generator A(%)
17 X_B = 7.5                // Reactance of generator B(%)
18 X_C = 10.0               // Reactance of generator C(%)
19 X_D = 10.0               // Reactance of generator D(%)
20 X_E = 8.0                 // Reactance of reactor E(%)
21 X_F = 8.0                 // Reactance of reactor F(%)
22 X_G = 6.5                 // Reactance of reactor G(%)
23 X_H = 6.5                 // Reactance of reactor H(%)
24
25 // Calculations
26 Z_1 = X_B*X_C/(X_H+X_B+X_C)           // Impedance(
    %). Fig E2.7
27 Z_2 = X_H*X_C/(X_H+X_B+X_C)           // Impedance(
    %). Fig E2.7
28 Z_3 = X_B*X_H/(X_H+X_B+X_C)           // Impedance(
    %). Fig E2.7
```

```

29 Z_4 = Z_2+X_F // Impedance(%). Fig E2.8 & Fig 2.9
30 Z_5 = Z_3+X_E // Impedance(%). Fig E2.8 & Fig 2.9
31 Z_6 = X_D*Z_1/(X_D+Z_1+Z_4) // Impedance(%). Fig E2.10
32 Z_7 = X_D*Z_4/(X_D+Z_1+Z_4) // Impedance(%). Fig E2.10
33 Z_8 = Z_1*Z_4/(X_D+Z_1+Z_4) // Impedance(%). Fig E2.10
34 Z_9 = Z_7+X_G // Impedance(%). Fig E2.11 & Fig 2.12
35 Z_10 = Z_8+Z_5 // Impedance(%). Fig E2.11 & Fig 2.12
36 Z_11 = Z_9*Z_10/(Z_9+Z_10) // Impedance(%). Fig 2.12 & Fig 2.13
37 Z_12 = Z_6+Z_11 // Impedance(%). Fig 2.13
38 Z_eq = X_A*Z_12/(X_A+Z_12) // Final Impedance(%). Fig 2.13 & Fig 2.14
39 MVA_SC = kVA_base*100/(Z_eq*1000) // Instantaneous symmetrical short-circuit MVA for a fault at X(MVA)
40
41 // Results
42 disp("PART III - EXAMPLE : 2.5 : SOLUTION :-")
43 printf("\nInstantaneous symmetrical short-circuit MVA for a fault at X = %.f MVA \n", MVA_SC)
44 printf("\nNOTE: Changes in the obtained answer from that of textbook is due to more approximation in the textbook")

```

---

# Chapter 29

## SYMMETRICAL COMPONENTS ANALYSIS

Scilab code Exa 29.1 Positive Negative and Zero sequence currents

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 3: SYMMETRICAL COMPONENTS' ANALYSIS
8
9 // EXAMPLE : 3.1 :
10 // Page number 487–488
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 I_R = complex(12.0,24.0)      // Line current (A)
15 I_Y = complex(16.0,-2.0)      // Line current (A)
16 I_B = complex(-4.0,-6.0)      // Line current (A)
17
18 // Calculations
```

```

19 alpha = exp(%i*120.0*%pi/180) //  

    Operator  

20 I_R0 = 1.0/3*(I_R+I_Y+I_B) // Zero  

    sequence component(A)  

21 I_R1 = 1.0/3*(I_R+alpha*I_Y+alpha**2*I_B) //  

    Positive sequence component(A)  

22 I_R2 = 1.0/3*(I_R+alpha**2*I_Y+alpha*I_B) //  

    Negative sequence component(A)  

23  

24 // Results  

25 disp("PART III - EXAMPLE : 3.1 : SOLUTION :-")  

26 printf("\nPositive sequence current , I_R1 = (%.3f +  

    %.1fj) A" , real(I_R1), imag(I_R1))  

27 printf("\nNegative sequence current , I_R2 = (%.3f +  

    %.2fj) A" , real(I_R2), imag(I_R2))  

28 printf("\nZero sequence current , I_R0 = (%.1f + %.2  

    f j) A" , real(I_R0), imag(I_R0))

```

---

#### Scilab code Exa 29.4 Sequence components of currents in the resistors and Supply 1

```

1 // A Texbook on POWER SYSTEM ENGINEERING  

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  

3 // DHANPAT RAI & Co.  

4 // SECOND EDITION  

5  

6 // PART III : SWITCHGEAR AND PROTECTION  

7 // CHAPTER 3: SYMMETRICAL COMPONENTS' ANALYSIS  

8  

9 // EXAMPLE : 3.4 :  

10 // Page number 489-490  

11 clear ; clc ; close ; // Clear the work space and  

    console  

12  

13 // Given data  

14 R_bbc = 5.0 // Resistance of resistor connected b/

```

```

    w b & c(ohm)
15 R_ca = 10.0 // Resistance of resistor connected b/
    w c & a(ohm)
16 R_ab = 20.0 // Resistance of resistor connected b/
    w a & b(ohm)
17 V = 100.0 // Voltage of balanced system(V)
18
19 // Calculations
20 E_A = -V // 
    Voltage across resistor connected b/w b & c(V)
21 angle = 60.0 // 
    Angle in delta system( )
22 E_B = V*exp(%i*60.0*pi/180) // 
    Voltage across resistor connected b/w c & a(V)
23 E_C = V*exp(%i*-60.0*pi/180) // 
    Voltage across resistor connected b/w a & b(V)
24 I_A = E_A/R_bc // 
    Current flowing across resistor connected b/w b &
    c(A)
25 I_B = E_B/R_ca // 
    Current flowing across resistor connected b/w c &
    a(A)
26 I_C = E_C/R_ab // 
    Current flowing across resistor connected b/w a &
    b(A)
27 alpha = exp(%i*120.0*pi/180) // 
    Operator
28 I_A0 = 1.0/3*(I_A+I_B+I_C) // Zero
    sequence delta current(A)
29 I_A1 = 1.0/3*(I_A+alpha*I_B+alpha**2*I_C) // 
    Positive sequence delta current(A)
30 I_A2 = 1.0/3*(I_A+alpha**2*I_B+alpha*I_C) // 
    Negative sequence delta current(A)
31 I_a0 = 0.0 // Zero
    sequence star current(A)
32 I_a1 = (alpha-alpha**2)*I_A1 // 
    Positive sequence star current(A)
33 I_a2 = (alpha**2-alpha)*I_A2 // 

```

```

        Negative sequence star current (A)
34
35 // Results
36 disp("PART III - EXAMPLE : 3.4 : SOLUTION :-")
37 printf("\nCurrent in the resistors are:")
38 printf("\n I_A = (%. f+%. fj) A", real(I_A), imag(I_A))
39 printf("\n I_B = (%. f+%. 2 fj) A", real(I_B), imag(I_B))
40 printf("\n I_C = (%. 1 f%. 2 fj) A", real(I_C), imag(I_C))
41 printf("\nSequence components of currents in the
resistors:")
42 printf("\n Zero-sequence current , I_A0 = (%.3 f+%.2 fj
) A", real(I_A0), imag(I_A0))
43 printf("\n Positive-sequence current , I_A1 = (%.2 f+%
.fj) A", real(I_A1), imag(I_A1))
44 printf("\n Negative-sequence current , I_A2 = (%.2 f%
.2 fj) A", real(I_A2), imag(I_A2))
45 printf("\nSequence components of currents in the
supply lines:")
46 printf("\n Zero-sequence current , I_a0 = %. f A",
I_a0)
47 printf("\n Positive-sequence current , I_a1 = %.1 fj A
", imag(I_a1))
48 printf("\n Negative-sequence current , I_a2 = (%.1 f+%
.2 fj) A", real(I_a2), imag(I_a2))

```

---

**Scilab code Exa 29.5 Magnitude of positive and Negative sequence components of the**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION

```

```

7 // CHAPTER 3: SYMMETRICAL COMPONENTS' ANALYSIS
8
9 // EXAMPLE : 3.5 :
10 // Page number 490-491
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 E_a = 100.0      // Line to line voltage(V)
15 E_b = 150.0      // Line to line voltage(V)
16 E_c = 200.0      // Line to line voltage(V)
17
18 // Calculations
19 e_A = 1.0          //
    100 V = 1 unit
20 e_B = 1.5          //
    150 V = 1 unit
21 e_C = 2.0          //
    200 V = 1 unit
22 cos_alpha = (e_C**2-e_A-e_B**2)/(2*e_B)
23 alpha = acosd(cos_alpha)           //
    angle( )
24 cos_beta = (e_A+e_B*cos_alpha)/e_C
25 beta = acosd(cos_beta)           //
    angle( )
26 E_A = E_a*exp(%i*180.0*pi/180)   //
    Voltage(V)
27 E_B = E_b*exp(%i*(180.0-alpha)*pi/180) //
    Voltage(V)
28 E_C = E_c*exp(%i*-beta*pi/180)     //
    Voltage(V)
29 a = exp(%i*120.0*pi/180)           //
    Operator
30 E_A0 = 1.0/3*(E_A+E_B+E_C)         //
    Zero sequence voltage(V)
31 E_A1 = 1.0/3*(E_A+a*E_B+a**2*E_C) //
    Positive sequence delta voltage(V)
32 E_A1_mag = abs(E_A1)                //

```

```

        Magnitude of positive sequence delta voltage(V)
33 E_a1 = -%i/3**0.5*E_A1                                //
        Positive sequence star voltage(V)
34 E_a1_mag = abs(E_a1)                                 //
        Magnitude of positive sequence star voltage(V)
35 E_A2 = 1.0/3*(E_A+a**2*E_B+a*E_C)                //
        Negative sequence delta voltage(V)
36 E_A2_mag = abs(E_A2)                                //
        Magnitude of negative sequence delta voltage(V)
37 E_a2 = %i/3**0.5*E_A2                            //
        Negative sequence star voltage(V)
38 E_a2_mag = abs(E_a2)                                //
        Magnitude of negative sequence star voltage(V)
39
40 // Results
41 disp("PART III – EXAMPLE : 3.5 : SOLUTION :-")
42 printf("\nMagnitude of positive sequence delta
        voltage , |E_A1| = %.f V" , E_A1_mag)
43 printf("\nMagnitude of positive sequence star
        voltage , |E_a1| = %.1f V" , E_a1_mag)
44 printf("\nMagnitude of negative sequence delta
        voltage , |E_A2| = %.f V" , E_A2_mag)
45 printf("\nMagnitude of negative sequence star
        voltage , |E_a2| = %.f V" , E_a2_mag)

```

---

**Scilab code Exa 29.6 Current in each line by the method of symmetrical components**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 3: SYMMETRICAL COMPONENTS' ANALYSIS
8

```

```

9 // EXAMPLE : 3.6 :
10 // Page number 491-492
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 V = 2300.0
   Rated voltage(V)
15 kVA = 500.0
   kVA rating
16 E_A = 2760.0*exp(%i*0*%pi/180)
   Line voltage(V)
17 E_B = 2300.0*exp(%i*-138.6*%pi/180)
   Line voltage(V)
18 E_C = 1840.0*exp(%i*124.2*%pi/180)
   Line voltage(V)
19
20 // Calculations
21 a = exp(%i*120.0*%pi/180) // Operator
22 E_A1 = 1.0/3*(E_A+a*E_B+a**2*E_C)
   Positive sequence voltage(V)
23 E_A2 = 1.0/3*(E_A+a**2*E_B+a*E_C)
   Negative sequence voltage(V)
24 E_a1 = -%i/3**0.5*E_A1
   Positive sequence star voltage(V)
25 E_a2 = %i/3**0.5*E_A2
   Negative sequence star voltage(V)
26 E_a0 = 0.0
   sequence voltage(V)
27 E_a = E_a1+E_a2+E_a0
   Symmetrical voltage component(V)
28 R = V**2/(kVA*1000)
   Resistance(ohm)
29 I_a = abs(E_a)/R
   Current in line a(A)
30 E_b = a**2*E_a1+a*E_a2+E_a0
   Symmetrical voltage component(V)

```

```

31 I_b = abs(E_b)/R //  

    Current in line b(A)  

32 E_c = a*E_a1+a**2*E_a2+E_a0 //  

    Symmetrical voltage component(V)  

33 I_c = abs(E_c)/R //  

    Current in line c(A)  

34  

35 // Results  

36 disp("PART III - EXAMPLE : 3.6 : SOLUTION :-")  

37 printf("\nCurrent in line a, |I_a| = %.1f A", I_a)  

38 printf("\nCurrent in line b, |I_b| = %.f A", I_b)  

39 printf("\nCurrent in line c, |I_c| = %.1f A \n", I_c)  

    )  

40 printf("\nNOTE: Changes in the obtained answer from  

    that of textbook is due to more precision here")

```

---

### Scilab code Exa 29.7 Symmetrical components of line current if phase 3 is only swi

```

1 // A Texbook on POWER SYSTEM ENGINEERING  

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  

3 // DHANPAT RAI & Co.  

4 // SECOND EDITION  

5  

6 // PART III : SWITCHGEAR AND PROTECTION  

7 // CHAPTER 3: SYMMETRICAL COMPONENTS' ANALYSIS  

8  

9 // EXAMPLE : 3.7 :  

10 // Page number 492-493  

11 clear ; clc ; close ; // Clear the work space and  

    console  

12  

13 // Given data  

14 V = 2300.0 //  

    Rated voltage (V)  

15 kVA = 500.0 // kVA

```

```

rating
16 I_1 = 100.0 // Line
    current(A)
17 I_2 = 100.0*exp(%i*180*%pi/180) // Line
    current(A)
18 I_3 = 0 // Line
    current(A)
19
20 // Calculations
21 a = exp(%i*120.0*%pi/180) // Operator
22 I_10 = 1.0/3*(I_1+I_2+I_3) //
    Symmetrical component of line current for phase
    1(A)
23 I_11 = 1.0/3*(I_1+a*I_2+a**2*I_3) //
    Symmetrical component of line current for phase
    1(A)
24 I_12 = 1.0/3*(I_1+a**2*I_2+a*I_3) //
    Symmetrical component of line current for phase
    1(A)
25 I_20 = I_10 //
    Symmetrical component of line current for phase
    2(A)
26 I_21 = a**2*I_11 //
    Symmetrical component of line current for phase
    2(A)
27 I_22 = a*I_12 //
    Symmetrical component of line current for phase
    2(A)
28 I_30 = I_10 //
    Symmetrical component of line current for phase
    3(A)
29 I_31 = a*I_11 //
    Symmetrical component of line current for phase
    3(A)
30 I_32 = a**2*I_12 //
    Symmetrical component of line current for phase
    3(A)
31

```

```

32 // Results
33 disp("PART III - EXAMPLE : 3.7 : SOLUTION :-")
34 printf("\nSymmetrical component of line current for
35 phase 1:")
36 printf("\n I_10 = %.1f A", abs(I_10))
37 printf("\n I_11 = %.2f % . f A", abs(I_11),
38 phasemag(I_11))
39 printf("\n I_12 = %.2f % . f A", abs(I_12),
40 phasemag(I_12))
41 printf("\n I_20 = %.1f A", abs(I_20))
42 printf("\n I_21 = %.2f % . f A", abs(I_21),
43 phasemag(I_21))
44 printf("\n I_22 = %.2f % . f A", abs(I_22),
45 phasemag(I_22))
46 printf("\n I_30 = %.1f A", abs(I_30))
47 printf("\n I_31 = %.2f % . f A", abs(I_31),
48 phasemag(I_31))
49 printf("\n I_32 = %.2f % . f A", abs(I_32),
50 phasemag(I_32))

```

---

### Scilab code Exa 29.8 Positive Negative and Zero sequence components of currents for

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 3: SYMMETRICAL COMPONENTS' ANALYSIS
8
9 // EXAMPLE : 3.8 :

```

```

10 // Page number 493
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 I_a = 1000.0
   earth(A) // Current to
15 I_b = 0 // Current(A)
16 I_c = 0 // Current(A)
17
18 // Calculations
19 a = exp(%i*120.0*%pi/180) // Operator
20 I_a0 = 1.0/3*(I_a+I_b+I_c)
   sequence component of current(A) // Zero
21 I_b0 = I_a0
   sequence component of current(A) // Zero
22 I_c0 = I_a0
   sequence component of current(A) // Zero
23 I_a1 = 1.0/3*(I_a+a*I_b+a**2*I_c)
   sequence component of current(A) // Positive
24 I_b1 = a**2*I_a1
   sequence component of current(A) // Positive
25 I_c1 = a*I_a1
   sequence component of current(A) // Positive
26 I_a2 = 1.0/3*(I_a+a**2*I_b+a*I_c)
   sequence component of current(A) // Negative
27 I_b2 = a*I_a2
   sequence component of current(A) // Negative
28 I_c2 = a**2*I_a2
   sequence component of current(A) // Negative
29
30 // Results
31 disp("PART III - EXAMPLE : 3.8 : SOLUTION :-")
32 printf("\nZero sequence component of current for all
   phases are")
33 printf("\n I_a0 = %.1 f % . f    A" , abs(I_a0),
   phasemag(I_a0))
34 printf("\n I_b0 = %.1 f % . f    A" , abs(I_b0),
   phasemag(I_b0))

```

```

    phasemag(I_b0))
35 printf("\n I_c0 = %.1 f % . f   A" , abs(I_c0) ,
    phasemag(I_c0))
36 printf("\nPositive sequence component of current for
        all phases are")
37 printf("\n I_a1 = %.1 f % . f   A" , abs(I_a1) ,
    phasemag(I_a1))
38 printf("\n I_b1 = %.1 f % . f   A" , abs(I_b1) ,360+
    phasemag(I_b1))
39 printf("\n I_c1 = %.1 f % . f   A" , abs(I_c1) ,
    phasemag(I_c1))
40 printf("\nNegative sequence component of current for
        all phases are")
41 printf("\n I_a2 = %.1 f % . f   A" , abs(I_a2) ,
    phasemag(I_a2))
42 printf("\n I_b2 = %.1 f % . f   A" , abs(I_b2) ,
    phasemag(I_b2))
43 printf("\n I_c2 = %.1 f % . f   A" , abs(I_c2) ,360+
    phasemag(I_c2))

```

---

### Scilab code Exa 29.9 Currents in all the lines and their symmetrical components

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 3: SYMMETRICAL COMPONENTS' ANALYSIS
8
9 // EXAMPLE : 3.9 :
10 // Page number 493-494
11 clear ; clc ; close ; // Clear the work space and
                           console
12

```

```

13 // Given data
14 I_A = 1000.0 // Current
15 through line A(A)
15 I_C = 0 // Current
16 through line C(A)
16
17 // Calculations
18 I_B = 1000.0*exp(%i*180.0*pi/180) // Current
19 through line B(A)
20 a = exp(%i*120.0*pi/180) // Operator
21 I_a0 = 1.0/3*(I_A+I_B+I_C) // Zero
22 sequence component of current (A)
21 I_b0 = I_a0 // Zero
22 sequence component of current (A)
22 I_c0 = I_a0 // Zero
23 sequence component of current (A)
23 I_a1 = 1.0/3*(I_A+a*I_B+a**2*I_C) // Positive
24 sequence component of current (A)
24 I_b1 = a**2*I_a1 // Positive
25 sequence component of current (A)
25 I_c1 = a*I_a1 // Positive
26 sequence component of current (A)
26 I_a2 = 1.0/3*(I_A+a**2*I_B+a*I_C) // Negative
27 sequence component of current (A)
27 I_b2 = a*I_a2 // Negative
28 sequence component of current (A)
28 I_c2 = a**2*I_a2 // Negative
29 sequence component of current (A)
29
30 // Results
31 disp("PART III – EXAMPLE : 3.9 : SOLUTION :-")
32 printf("\nCurrent in line A, I_A = %. f % . f A",
33 abs(I_A),phasemag(I_A))
33 printf("\nCurrent in line B, I_B = %. f % . f A",
34 abs(I_B),phasemag(I_B))
34 printf("\nCurrent in line C, I_C = %. f A", I_C)
35 printf("\nSymmetrical current components of line A
are :")

```

```

36 printf("\n I_a0 = %.f A", abs(I_a0))
37 printf("\n I_a1 = %.1f % . f A", abs(I_a1),
         phasemag(I_a1))
38 printf("\n I_a2 = %.1f % . f A", abs(I_a2),
         phasemag(I_a2))
39 printf("\n Symmetrical current components of line B
        are :")
40 printf("\n I_b0 = %.f A", abs(I_b0))
41 printf("\n I_b1 = %.1f % . f A", abs(I_b1),
         phasemag(I_b1))
42 printf("\n I_b2 = %.1f % . f A", abs(I_b2),
         phasemag(I_b2))
43 printf("\n Symmetrical current components of line C
        are :")
44 printf("\n I_c0 = %.f A", abs(I_c0))
45 printf("\n I_c1 = %.1f % . f A", abs(I_c1),
         phasemag(I_c1))
46 printf("\n I_c2 = %.1f % . f A", abs(I_c2),
         phasemag(I_c2))

```

---

### Scilab code Exa 29.10 Radius of voltmeter connected to the yellow line and Current

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 3: SYMMETRICAL COMPONENTS' ANALYSIS
8
9 // EXAMPLE : 3.10 :
10 // Page number 494
11 clear ; clc ; close ; // Clear the work space and
                           console
12

```

```

13 // Given data
14 R = 20000.0 // Resistance of voltmeter (ohm)
15 E_R = 100.0 // Line-to-neutral voltage (A)
16 E_Y = 200.0*exp(%i*270.0*pi/180) // Line-to-neutral voltage (A)
17 E_B = 100.0*exp(%i*120.0*pi/180) // Line-to-neutral voltage (A)
18
19 // Calculations
20 a = exp(%i*120.0*pi/180) // Operator
21 V_R0 = 1.0/3*(E_R+E_Y+E_B) // Zero
    sequence voltage(V)
22 V_R1 = 1.0/3*(E_R+a*E_Y+a**2*E_B) // Positive
    sequence voltage(V)
23 V_R2 = 1.0/3*(E_R+a**2*E_Y+a*E_B) // Negative
    sequence voltage(V)
24 I_R1 = V_R1/R // Positive
    sequence current(A)
25 I_R2 = V_R2/R // Negative
    sequence current(A)
26 V_Y1 = a**2*V_R1 // Positive
    sequence voltage of line Y(V)
27 V_Y2 = a*V_R2 // Negative
    sequence voltage of line Y(V)
28 V_Y = V_Y1+V_Y2 // Voltmeter
    reading connected to the yellow line(V)
29 I_Y = abs(V_Y)/R*1000 // Current
    through voltmeter (mA)
30
31 // Results
32 disp("PART III - EXAMPLE : 3.10 : SOLUTION :-")
33 printf("\nVoltmeter reading connected to the yellow
        line , |V_Y| = %.1f V" , abs(V_Y))
34 printf("\nCurrent through voltmeter , I_Y = %.3f mA \
        " , I_Y)
35 printf("\nNOTE: Changes in the obtained answer from

```

that of textbook is due to more precision here")

---

### Scilab code Exa 29.11 Three line currents and Wattmeter reading

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 3: SYMMETRICAL COMPONENTS' ANALYSIS
8
9 // EXAMPLE : 3.11 :
10 // Page number 495
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 400.0          // Voltage(V)
15 Z_ab = 20.0        // Resistor load(ohm)
16 Z_bc = -%i*40.0   // Capacitor load(ohm)
17 Z_ca = 5.0+%i*10.0 // Inductor and resistance
    load(ohm)
18
19 // Calculations
20 V_ab = V           // Line voltage(V)
21 V_bc = V*exp(%i*-120.0*%pi/180) // Line voltage(V)
22 V_ca = V*exp(%i*120.0*%pi/180) // Line voltage(V)
23 I_ab = V_ab/Z_ab // Current(A)
24 I_bc = V_bc/Z_bc // Current(A)
```

```

25 I_ca = V_ca/Z_ca          //  

    Current(A)  

26 I_a = I_ab-I_ca           //  

    Line current(A)  

27 I_b = I_bc-I_ab           //  

    Line current(A)  

28 I_c = I_ca-I_bc           //  

    Line current(A)  

29 phi = -120.0-phasemag(I_a) //  

    ( )  

30 P = abs(I_a*V_bc)*cosd(phi)/1000 //  

    Wattmeter reading(kW)  

31  

32 // Results  

33 disp("PART III - EXAMPLE : 3.11 : SOLUTION :-")  

34 printf("\nLine currents are:")  

35 printf("\n I_a = %.1 f % .1 f A", abs(I_a), phasemag(I_a))  

36 printf("\n I_b = %.1 f % .2 f A", abs(I_b), phasemag(I_b))  

37 printf("\n I_c = %.2 f % .f A", abs(I_c), phasemag(I_c))  

38 printf("\nWattmeter reading , P = %.2 f kW \n", P)  

39 printf("\nNOTE: ERROR: Calculation mistakes in the  

        textbook solution")

```

---

# Chapter 30

## UNSYMMETRICAL FAULTS IN POWER SYSTEMS

Scilab code Exa 30.1 Initial symmetrical rms line currents Ground wire currents and

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 4: UNSYMMETRICAL FAULTS IN POWER SYSTEMS
8
9 // EXAMPLE : 4.1 :
10 // Page number 510–512
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 MVA = 15.0      // Generator rating (MVA)
15 kV = 6.9        // Generator voltage (kV)
16 X_1 = 25.0      // Positive sequence reactance (%)
17 X_2 = 25.0      // Negative sequence reactance (%)
18 X_0 = 8.0       // Zero sequence reactance (%)
```

```

19 X = 6.0           // Reactor placed in line(%)
20
21 // Calculations
22 a = exp(%i*120.0*%pi/180)           // Operator
23 Z_1 = %i*X_1/100
24
25 // Positive sequence impedance(p.u)
24 Z_2 = %i*X_2/100
26
27 // Negative sequence impedance(p.u)
25 Z_g0 = %i*X_0/100
28
29 // Impedance(p.u)
26 Z = %i*X/100
30
31 // Zero sequence impedance(p.u)
28 E_a = 1.0
32
33 // Voltage(p.u)
29 E_b = a**2*E_a
34
35 // Case(a)
30 I_a0_a = 0
36
37 // Current(A)
32 I_a1_a_pu = 1.0/(Z_1+Z_2)
38
39 // Current(p.u)
33 I_a1_a = I_a1_a_pu*MVA*1000/(3**0.5*kV)
40
41 // Current(A)
34 I_a2_a = -I_a1_a
42
43 // Current(A)

```

```

35 I_b0_a = 0
      // Current (A)
36 I_b1_a = a**2*I_a1_a
      // Current (A)
37 I_b2_a = a*I_a2_a
      // Current (A)
38 I_a_a = I_a1_a+I_a2_a
      // Line current (A)
39 I_b_a = I_b1_a+I_b2_a
      // Line current (A)
40 I_c_a = -I_b_a
      // Line current (A)
41 I_g_a = 0
      // Ground wire current (A)
42 V_a_a = (E_a-I_a1_a*Z_1-I_a2_a*Z_2-I_a0_a*Z_0)*kV
      *1000/3**0.5      // Voltage (V)
43 V_b_a = (a**2*E_a+%i*3**0.5*I_a1_a_pu*Z_1)*kV
      *1000/3**0.5      // Voltage (V)
44 V_c_a = V_b_a
      // Voltage (V)
45 // Case (b)
46 I_a1_b_pu = E_a/(Z_1+(Z_2*Z_0/(Z_2+Z_0)))
      // Current (p.u)
47 I_a1_b = I_a1_b_pu*MVA*1000/(3**0.5*kV)
      // Current (A)
48 I_a2_b_pu = -Z_0*Z_2/(Z_2*(Z_0+Z_2))*I_a1_b_pu
      // Current (p.u)
49 I_a2_b = -Z_0*Z_2/(Z_2*(Z_0+Z_2))*I_a1_b
      // Current (A)
50 I_a0_b_pu = -Z_0*Z_2/(Z_0*(Z_0+Z_2))*I_a1_b_pu

```

```

// Current(p.u)
51 I_a0_b = -Z_0*Z_2/(Z_0*(Z_0+Z_2))*I_a1_b
                           // Current(A)
52 I_a_b = I_a0_b+I_a1_b+I_a2_b
                           // Line
                           current(A)
53 I_b_b = I_a0_b+a**2*I_a1_b+a*I_a2_b
                           // Line current(A)
54 I_c_b = I_a0_b+a*I_a1_b+a**2*I_a2_b
                           // Line current(A)
55 I_0_b = 3*I_a0_b

                           // Current in the ground resistor(A)
56 V_a_b_pu = E_a-I_a1_b_pu*Z_1-I_a2_b_pu*Z_2-I_a0_b_pu
                           *Z_0          // Voltage(p.u)
57 V_a_b = abs(V_a_b_pu)*kV*1000/(3**0.5)
                           // Voltage(V)
58 V_b_b = 0

                           // Voltage(V)
59 V_c_b = 0

                           // Voltage(V)
60
61 // Results
62 disp("PART III - EXAMPLE : 4.1 : SOLUTION :-")
63 printf("\nCase(a): Initial symmetrical rms line
           current when ground is not involved in fault , I_a
           = %.f A" , abs(I_a_a))
64 printf("\n           Initial symmetrical rms line
           current when ground is not involved in fault , I_b
           = %.f A" , real(I_b_a))
65 printf("\n           Initial symmetrical rms line
           current when ground is not involved in fault , I_c
           = %.f A" , real(I_c_a))
66 printf("\n           Ground wire current = %.f A" ,
           I_g_a)
67 printf("\n           Line to neutral voltage , V_a = %.

```

```

f V" , real(V_a_a))
68 printf("\\n Line to neutral voltage , V_b = %.
f V" , real(V_b_a))
69 printf("\\n Line to neutral voltage , V_c = %.
f V" , real(V_c_a))
70 printf("\\nCase(b): Initial symmetrical rms line
current when fault is solidly grounded , I_a = %.f
A" , abs(I_a_b))
71 printf("\\n Initial symmetrical rms line
current when fault is solidly grounded , I_b = (%.
f +%. f j ) A" , real(I_b_b) , imag(I_b_b))
72 printf("\\n Initial symmetrical rms line
current when fault is solidly grounded , I_c = (%.
f +%. f j ) A" , real(I_c_b) , imag(I_c_b))
73 printf("\\n Ground wire current = %. f j A",
imag(I_0_b))
74 printf("\\n Line to neutral voltage , V_a = %.
f V" , V_a_b)
75 printf("\\n Line to neutral voltage , V_b = %.
f V" , V_b_b)
76 printf("\\n Line to neutral voltage , V_c = %.
f V\\n" , V_c_b)
77 printf("\\nNOTE: Changes in the obtained answer from
that of textbook is due to more precision here
and approximation in textbook")

```

---

### Scilab code Exa 30.2 Current in the line with two lines short circuited

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 4: UNSYMMETRICAL FAULTS IN POWER SYSTEMS

```

```

8
9 // EXAMPLE : 4.2 :
10 // Page number 512
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 kVA = 10000.0      // Generator rating (kVA)
15 f = 50.0            // Frequency (Hz)
16 I_1 = 30.0          // Positive sequence current (%)
17 I_2 = 10.0          // Negative sequence current (%)
18 I_0 = 5.0           // Zero sequence current (%)
19 d = 1.0/100         // Diameter of conductor (m)
20 D = 5.0             // Triangular spacing (m)
21 kV = 30.0           // Generator voltage on open-
    circuit (kV)
22 l = 20.0            // Distance of line at short
    circuit occurrence (km)
23
24 // Calculations
25 a = exp(%i*120.0*%pi/180)
               //
        Operator
26 Z_g1 = kV**2*I_1*I_2/kVA
               //
        Positive phase sequence reactance of generator (
        ohm)
27 Z_g2 = Z_g1*I_2/I_1
               //
        Negative phase sequence reactance of generator (
        ohm)
28 Z_g0 = Z_g1*I_0/I_1
               //
        Zero phase sequence reactance of generator (ohm)
29 r = d/2
               //
        Radius of conductor (m)
30 Z_l1 = 2.0*%pi*f*(0.5+4.606*log10(D/r))*10**-7*l

```

```

*1000           // Positive phase sequence
reactance of line(ohm)
31 Z_12 = 2.0*%pi*f*(0.5+4.606*log10(D/r))*10**-7*1
*1000           // Negative phase sequence
reactance of line(ohm)
32 Z_1 = %i*(Z_g1+Z_11)
                           //
      Z1 upto the point of fault(ohm)
33 Z_2 = %i*(Z_g2+Z_12)
                           //
      Z2 upto the point of fault(ohm)
34 E_a = kV*1000/3**0.5
                           //
      Phase voltage(V)
35 I_a1 = E_a/(Z_1+Z_2)
                           //
      Positive sequence current in line a(A)
36 I_a2 = -I_a1
                           //
      Negative sequence current in line a(A)
37 I_a0 = 0
                           //
      Zero sequence current in line a(A)
38 I_b0 = 0
                           //
      Zero sequence current in line b(A)
39 I_c0 = 0
                           //
      Zero sequence current in line c(A)
40 I_a = I_a0+I_a1+I_a2
                           //
      Current in line a(A)
41 I_b = I_b0+a**2*I_a1+a*I_a2
                           //
      in line b(A)
42 I_c = I_c0+a*I_a1+a**2*I_a2
                           //
      in line c(A)

```

```

43
44 // Results
45 disp("PART III - EXAMPLE : 4.2 : SOLUTION :-")
46 printf("\nCurrent in line a, I_a = %.f A", abs(I_a))
47 printf("\nCurrent in line b, I_b = %.f A", real(I_b))
48 printf("\nCurrent in line c, I_c = %.f A", real(I_c))

```

---

### Scilab code Exa 30.3 Fault current Sequence component of current and Voltages of t

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 4: UNSYMMETRICAL FAULTS IN POWER SYSTEMS
8
9 // EXAMPLE : 4.3 :
10 // Page number 512-513
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 kVA = 10000.0          // Alternator rating(
    kVA)
15 Z_g1 = complex(0.5,4.7) // Positive sequence
    impedance(ohm/phase)
16 Z_g2 = complex(0.2,0.6) // Negative sequence
    impedance(ohm/phase)
17 Z_g0 = complex(0,0.43) // Zero sequence
    impedance(ohm/phase)
18 Z_l1 = complex(0.36,0.25) // Impedance(ohm)
19 Z_l2 = complex(0.36,0.25) // Impedance(ohm)

```

```

20 Z_10 = complex(2.9,0.95)           // Impedance (ohm)
21 V = 6600.0                         // Voltage (V)
22
23 // Calculations
24 a = exp(%i*120.0*pi/180)          // Operator
25 // Case(a)
26 E_a = V/3**0.5                      // Phase
27 voltage(V)
28 Z_1 = Z_g1+Z_11                     // Z1 upto
29 the point of fault (ohm)
30 Z_2 = Z_g2+Z_12                     // Z2 upto
31 the point of fault (ohm)
32 Z_0 = Z_g0+Z_10                     // Z0 upto
33 the point of fault (ohm)
34 I_a = 3*E_a/(Z_1+Z_2+Z_0)          // Fault current (A)
35 // Case(b)
36 I_a0 = abs(I_a)/3                  // Zero
37 sequence current of line a(A)
38 I_a1 = abs(I_a)/3                  // Positive
39 sequence current of line a(A)
40 I_a2 = abs(I_a)/3                  // Negative
41 sequence current of line a(A)
42 I_b0 = I_a0                         // Zero
43 sequence current of line b(A)
44 I_b1 = a**2*I_a1                   // Positive
45 sequence current of line b(A)
46 I_b2 = a*I_a2

```

```

// Negative sequence current of line b(A)
38 I_c0 = I_a0 // Zero
    sequence current of line c(A)
39 I_c1 = a*I_a1 // Positive sequence current of line c(A)
40 I_c2 = a**2*I_a2 // Negative
    sequence current of line c(A)
41 // Case(c)
42 V_b = E_a/(Z_1+Z_2+Z_0)*((a**2-a)*Z_2+(a**2-1)*Z_0)
    // Voltage of the line b(V)
43 V_c = E_a/(Z_1+Z_2+Z_0)*((a-a**2)*Z_2+(a-1)*Z_0)
    // Voltage of the line c(V)
44
45 // Results
46 disp("PART III - EXAMPLE : 4.3 : SOLUTION :-")
47 printf("\nCase(a): Fault current , | I_a | = %.f A",
        abs(I_a))
48 printf("\nCase(b): Zero sequence current of line a ,
        I_a0 = %.f A" , I_a0)
49 printf("\n          Positive sequence current of line
        a , I_a1 = %.f A" , I_a1)
50 printf("\n          Negative sequence current of line
        a , I_a2 = %.f A" , I_a2)
51 printf("\n          Zero sequence current of line b ,
        I_b0 = %.f A" , I_b0)
52 printf("\n          Positive sequence current of line
        b , I_b1 = (%.1f%.1fj) A" , real(I_b1), imag(I_b1))
53 printf("\n          Negative sequence current of line
        b , I_b2 = (%.1f%.1fj) A" , real(I_b2), imag(I_b2))
54 printf("\n          Zero sequence current of line c ,
        I_c0 = %.f A" , I_c0)
55 printf("\n          Positive sequence current of line
        c , I_c1 = (%.1f%.1fj) A" , real(I_c1), imag(I_c1))

```

```

)
56 printf("\n      Negative sequence current of line
      c , I_c2 = (%.1f%.1fj) A" , real(I_c2),imag(I_c2))
57 printf("\nCase(c): Voltage of the sound line to
      earth at fault , |V_b| = %.f V" , abs(V_b))
58 printf("\n      Voltage of the sound line to
      earth at fault , |V_c| = %.f V\n" , abs(V_c))
59 printf("\nNOTE: Changes in the obtained answer from
      that of textbook is due to more precision here")

```

---

#### Scilab code Exa 30.4 Fault currents in each line and Potential above earth attained

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 4: UNSYMMETRICAL FAULTS IN POWER SYSTEMS
8
9 // EXAMPLE : 4.4 :
10 // Page number 513–514
11 clear ; clc ; close ; // Clear the work space and
      console
12
13 // Given data
14 V = 11000.0 // Alternator voltage
      (V)
15 kVA = 50000.0 // Alternator rating(
      kVA)
16 Z_11 = complex(0.4,0.7) // Positive sequence
      impedance of feeder(ohm)
17 Z_12 = complex(0.4,0.7) // Negative sequence
      impedance of feeder(ohm)
18 Z_10 = complex(0.7,3.0) // Zero sequence

```

```

        impedance of feeder (ohm)
19 Z_g1_A = complex(0,0.6)           // Positive sequence
      reactance(ohm)
20 Z_g1_B = complex(0,0.6)           // Positive sequence
      reactance(ohm)
21 Z_g2_A = complex(0,0.4)           // Negative sequence
      reactance(ohm)
22 Z_g2_B = complex(0,0.4)           // Negative sequence
      reactance(ohm)
23 Z_g0_A = complex(0,0.2)           // Zero sequence
      reactance(ohm)
24 Z_g0_B = complex(0,0.2)           // Zero sequence
      reactance(ohm)
25 Z_n_A = complex(0,0.2)           // Neutral reactance(
      ohm)
26 Z_n_B = complex(0,0.2)           // Neutra reactance(
      ohm)
27
28 // Calculations
29 a = exp(%i*120.0*%pi/180)         //
      Operator
30 Z_g1 = 1.0/((1/Z_g1_A)+(1/Z_g1_B)) // 
      Equivalent positive sequence impedance(ohm)
31 Z_g2 = 1.0/((1/Z_g2_A)+(1/Z_g2_B)) // 
      Equivalent negative sequence impedance(ohm)
32 Z_g0 = 1.0/((1/Z_g0_A)+(1/Z_g0_B)) // 
      Equivalent zero sequence impedance(ohm)
33 Z_n = 1.0/((1/Z_n_A)+(1/Z_n_B))    //
      Equivalent neutral impedance(ohm)
34 Z_1 = Z_11+Z_g1                   //
      Positive sequence impedance(ohm)
35 Z_2 = Z_12+Z_g2                   //
      Negative sequence impedance(ohm)
36 Z_0 = Z_10+Z_g0+3*Z_n             // Zero
      sequence impedance(ohm)
37 Z = Z_0*Z_2/(Z_0+Z_2)             //
      Impedance(ohm)
38 E_R = V/3**0.5                    //

```

```

    Phase voltage (V)
39 I_R1 = E_R/(Z_1+Z)                                //
    Postive sequence current (A)
40 I_R2 = -Z*I_R1/Z_2                                 //
    Negative sequence current (A)
41 I_R0 = -Z*I_R1/Z_0                                 // Zero
    sequence current (A)
42 I_R = I_R0+I_R1+I_R2                               //
    Fault current in line(A)
43 I_Y = I_R0+a**2*I_R1+a*I_R2                      //
    Fault current in line(A)
44 I_B = I_R0+a*I_R1+a**2*I_R2                      //
    Fault current in line(A)
45 I_earth = 3.0*I_R0                                  //
    Current through earth reactance(A)
46 V_neutral = abs(I_earth*Z_n)                       //
    Magnitude of potential above earth attained by
    generator neutral(V)

47
48 // Results
49 disp("PART III - EXAMPLE : 4.4 : SOLUTION :-")
50 printf("\nFault current in the line R, I_R = %.f A",
      abs(I_R))
51 printf("\nFault current in the line Y, I_Y = (%.f%.
      f j) A", real(I_Y), imag(I_Y))
52 printf("\nFault current in the line B, I_B = (%.f+%.
      f j) A", real(I_B), imag(I_B))
53 printf("\nPotential above earth attained by the
      alternator neutrals = %.f V\n", V_neutral)
54 printf("\nNOTE: ERROR: Voltage is 11000 not 11000 kV
      as given in textbook statement")
55 printf("\n      Changes in the obtained answer from
      that of textbook is due to more precision here")

```

---

### Scilab code Exa 30.5 Fault currents

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 4: UNSYMMETRICAL FAULTS IN POWER SYSTEMS
8
9 // EXAMPLE : 4.5 :
10 // Page number 514–515
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 6600.0           // Alternator voltage(V)
15 kVA = 10000.0        // Alternator rating(kVA)
16 x_1 = 15.0           // Reactance to positive
    sequence current(%)
17 x_2 = 75.0           // Reactance to negative
    sequence current(%)
18 x_0 = 30.0           // Reactance to zero sequence
    current(%)
19 R_earth = 0.3         // Earth resistance(ohm)
20
21 // Calculations
22 a = exp(%i*120.0*%pi/180)          // Operator
23 E_g = V/3**0.5                   // Phase
    voltage(V)
24 // Case(a)
25 I = kVA*1000/(3**0.5*V)          // Full load
    current of each alternator(A)
26 X = x_1*V/(100*3**0.5*I)         // Positive
    sequence reactance(ohm)
27 Z_g1 = %i*X                      //
    Equivalent positive sequence impedance(ohm)
28 Z_g2 = Z_g1*x_2/100                //
    Equivalent negative sequence impedance(ohm)
29 Z_g0 = Z_g1*x_0/100                //

```

```

        Equivalent zero sequence impedance(ohm)
30 Z_1 = Z_g1/3                                // Positive
      sequence impedance(ohm)
31 Z_2 = Z_g2/3                                // Negative
      sequence impedance(ohm)
32 Z_0 = Z_g0/3                                // Zero
      sequence impedance(ohm)
33 I_a_a = 3*E_g/(Z_1+Z_2+Z_0)                // Fault
      current(A)
34 // Case(b)
35 Z_0_b = Z_g0                                 // Impedance
      (ohm)
36 I_a_b = 3*E_g/(Z_1+Z_2+Z_0_b)              // Fault
      current(A)
37 // Case(c)
38 Z_0_c = R_earth*3+Z_g0                      // Impedance
      (ohm)
39 I_a_c = 3*E_g/(Z_1+Z_2+Z_0_c)              // Fault
      current(A)
40
41 // Results
42 disp("PART III - EXAMPLE : 4.5 : SOLUTION :-")
43 printf("\nCase(a): Fault current if all the
      alternator neutrals are solidly earthed , I_a = %.
      f j A", imag(I_a_a))
44 printf("\nCase(b): Fault current if only one of the
      alternator neutrals is solidly earthed & others
      isolated = %.f j A", imag(I_a_b))
45 printf("\nCase(c): Fault current if one of
      alternator neutrals is earthed through resistance
      & others isolated = %.f A\n", abs(I_a_c))
46 printf("\nNOTE: Changes in the obtained answer from
      that of textbook is due to more precision here")

```

---

**Scilab code Exa 30.6 Fault current for line fault and Line to ground fault**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 4: UNSYMMETRICAL FAULTS IN POWER SYSTEMS
8
9 // EXAMPLE : 4.6 :
10 // Page number 515–516
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 kVA_G = 2000.0          // Generator rating (kVA)
15 X_G = 10.0              // Generator reactance(%)
16 kVA_T1 = 2000.0         // Transformer rating (kVA)
17 lv_T1 = 6.6              // LV side voltage(kV)
18 hv_T1 = 11.0             // HV side voltage(kV)
19 X_T1 = 5.0              // Transformer reactance(%)
20 X_cable = 0.5            // Cable reactance(ohm)
21 V_cable = 11.0           // Cable voltage(V)
22 kVA_T2 = 2000.0         // Transformer rating (kVA)
23 lv_T2 = 6.6              // LV side voltage(kV)
24 hv_T2 = 11.0             // HV side voltage(kV)
25 X_T2 = 5.0              // Transformer reactance(%)
26
27 // Calculations
28 a = exp(%i*120.0*pi/180) // Operator
29 kVA_base = 2000.0        // Base
    kVA
30 kV = 6.6                 // Base
    voltage (kV)
31 X_1 = X_G*kV**2*10/kVA_base // 10%
    reactance at 6.6 kV(ohm)
32 X_2 = X_T1*kV**2*10/kVA_base // 5%
    reactance at 6.6 kV(ohm)

```

```

33 X_3 = (kV/hv_T1)**2*X_cable // 0.5
    ohm at 11kV when referred to 6.6kV(ohm)
34 Z_g1 = %i*X_1 // Positive sequence impedance of generator(ohm)
35 Z_g2 = Z_g1*0.7 // Negative sequence impedance of generator equal to
    70% of +ve sequence impedance(ohm)
36 T1_Z_T1_1 = %i*X_2 // Positive sequence impedance of transformer(ohm)
37 T1_Z_T1_2 = %i*X_2 // Negative sequence impedance of transformer(ohm)
38 Z_C1 = %i*X_3 // Positive sequence impedance of cable(ohm)
39 Z_C2 = %i*X_3 // Negative sequence impedance of cable(ohm)
40 T2_Z_T2_1 = %i*X_2 // Positive sequence impedance of transformer(ohm)
41 T2_Z_T2_2 = %i*X_2 // Negative sequence impedance of transformer(ohm)
42 Z_1 = Z_g1+T1_Z_T1_1+Z_C1+T2_Z_T2_1 // Positive sequence impedance(ohm)
43 Z_2 = Z_g2+T1_Z_T1_2+Z_C2+T2_Z_T2_2 // Negative sequence impedance(ohm)
44 Z_0 = %i*X_2 // Zero sequence impedance(ohm)
45 E_a = kV*1000/3**0.5 // Phase voltage(V)
46 // Case(a)
47 I_a1 = E_a/(Z_1+Z_2) // Positive sequence current(A)
48 I_a2 = -I_a1 // Negative sequence current(A)
49 I_a0 = 0 // Zero sequence current(A)
50 I_a = I_a1+I_a2+I_a0 // Fault current in line a(A)
51 I_b = (a**2-a)*I_a1 // Fault current in line b(A)

```

```

52 I_c = -I_b // Fault current in line c(A)
53 // Case(b)
54 I_a_b = 3*E_a/(Z_1+Z_2+Z_0) // Fault current for line to ground fault (A)
55
56 // Results
57 disp("PART III – EXAMPLE : 4.6 : SOLUTION :-")
58 printf("\nCase(a): Fault current for line fault are"
      )
59 printf("\n          I_a = %.f A", abs(I_a))
60 printf("\n          I_b = %.f A", abs(I_b))
61 printf("\n          I_c = %.f A", abs(I_c))
62 printf("\nCase(b): Fault current for line to ground
      fault , | I_a | = %.f A\n", abs(I_a_b))
63 printf("\nNOTE: Changes in the obtained answer from
      that of textbook is due to more precision here")

```

---

### Scilab code Exa 30.7 Fault current for a LG fault at C

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 4: UNSYMMETRICAL FAULTS IN POWER SYSTEMS
8
9 // EXAMPLE : 4.7 :
10 // Page number 516–518
11 clear ; clc ; close ; // Clear the work space and
      console
12
13 // Given data
14 MVA_G1 = 40.0 // Generator rating (MVA)

```

```

15 kV_G1 = 13.2          // Generator voltage (kV)
16 X_st_G1 = 0.15        // Sub-transient reactance (p.u)
17 X_2_G1 = 0.15          // Negative sequence reactance (p.
                           u)
18 X_0_G1 = 0.08          // Zero sequence reactance (p.u)
19 MVA_G3 = 60.0          // Generator rating (MVA)
20 kV_G3 = 13.8           // Generator voltage (kV)
21 X_st_G3 = 0.20        // Sub-transient reactance (p.u)
22 X_2_G3 = 0.20          // Negative sequence reactance (p.
                           u)
23 X_0_G3 = 0.08          // Zero sequence reactance (p.u)
24 MVA_T1 = 40.0          // Transformer rating (MVA)
25 kV_lv_T1 = 13.8        // Transformer low voltage (kV)
26 kV_hv_T1 = 138         // Transformer high voltage (kV)
27 X_1_T1 = 0.10          // Positive sequence reactance (p.
                           u)
28 X_2_T1 = 0.10          // Negative sequence reactance (p.
                           u)
29 X_0_T1 = 0.08          // Zero sequence reactance (p.u)
30 MVA_T5 = 30.0          // Transformer rating (MVA)
31 kV_lv_T5 = 13.8        // Transformer low voltage (kV)
32 kV_hv_T5 = 138         // Transformer high voltage (kV)
33 X_1_T5 = 0.10          // Positive sequence reactance (p.
                           u)
34 X_2_T5 = 0.10          // Negative sequence reactance (p.
                           u)
35 X_0_T5 = 0.08          // Zero sequence reactance (p.u)
36 X_neutral = 0.05        // Reactance of reactor
                           connected to generator neutral (p.u)
37
38 // Calculations
39 MVA_base = 100.0
                           // Base MVA
40 kV_line = 138.0
                           // Base voltage for line (kV)
41 kV_G = 13.8

```

```

        // Base voltage for generator (kV)
42 X_st_G1_pu = %i*X_st_G1*(kV_G1/kV_G)**2*MVA_base/
    MVA_G1           // Impedance of G1 & G2(p.u)
43 X_2_G1_pu = %i*X_2_G1*(kV_G1/kV_G)**2*MVA_base/
    MVA_G1           // Impedance of G1 & G2(p.u)
44 X_g0_G1_pu = %i*X_0_G1*(kV_G1/kV_G)**2*MVA_base/
    MVA_G1           // Impedance of G1 & G2(p.u)
45 X_gn_G1_pu = %i*X_neutral*(kV_G1/kV_G)**2*MVA_base/
    MVA_G1           // Impedance of G1 & G2(p.u)
46 X_st_G3_pu = %i*X_st_G3*(kV_G3/kV_G)**2*MVA_base/
    MVA_G3           // Impedance of G3(p.u)
47 X_2_G3_pu = %i*X_2_G3*(kV_G3/kV_G)**2*MVA_base/
    MVA_G3           // Impedance of G3(p.u)
48 X_g0_G3_pu = %i*X_0_G3*(kV_G3/kV_G)**2*MVA_base/
    MVA_G3           // Impedance of G3(p.u)
49 X_gn_G3_pu = %i*X_neutral*(kV_G3/kV_G)**2*MVA_base/
    MVA_G3           // Impedance of G3(p.u)
50 X_1_T1_pu = %i*X_1_T1*MVA_base/MVA_T1
                           // Impedance of T1,T2
                           ,T3 & T4(p.u)
51 X_2_T1_pu = %i*X_2_T1*MVA_base/MVA_T1
                           // Impedance of T1,T2
                           ,T3 & T4(p.u)
52 X_0_T1_pu = %i*X_0_T1*MVA_base/MVA_T1
                           // Impedance of T1,T2
                           ,T3 & T4(p.u)
53 X_1_T5_pu = %i*X_1_T5*MVA_base/MVA_T5
                           // Impedance of T5 &
                           T6(p.u)
54 X_2_T5_pu = %i*X_2_T5*MVA_base/MVA_T5
                           // Impedance of T5 &
                           T6(p.u)
55 X_0_T5_pu = %i*X_0_T5*MVA_base/MVA_T5
                           // Impedance of T5 &
                           T6(p.u)
56 X_1_line_20 = %i*20.0*100/kV_line**2
                           // Impedance of 20

```

```

        ohm line(p.u)
57 X_2_line_20 = %i*20.0*100/kV_line**2
                           // Impedance of 20
        ohm line(p.u)
58 X_0_line_20 = 3.0*X_1_line_20
                           // Impedance
                           of 20 ohm line(p.u)
59 X_1_line_10 = %i*10.0*100/kV_line**2
                           // Impedance of 10
        ohm line(p.u)
60 X_2_line_10 = %i*10.0*100/kV_line**2
                           // Impedance of 10
        ohm line(p.u)
61 X_0_line_10 = 3.0*X_1_line_10
                           // Impedance
                           of 10 ohm line(p.u)
62 // Positive , negative and zero sequence network
63 Z_1_1 = X_1_T1_pu+X_1_T1_pu+X_1_line_20
                           // Impedance(p.u)
64 Z_2_1 = X_1_T1_pu+X_1_T5_pu+X_1_line_10
                           // Impedance(p.u)
65 Z_3_1 = X_1_T1_pu+X_1_T5_pu+X_1_line_10
                           // Impedance(p.u)
66 Z_4_1 = Z_1_1*Z_2_1/(Z_1_1+Z_2_1+Z_3_1)
                           // Impedance after star
                           -delta transformation(p.u)
67 Z_5_1 = Z_3_1*Z_1_1/(Z_1_1+Z_2_1+Z_3_1)
                           // Impedance after star
                           -delta transformation(p.u)
68 Z_6_1 = Z_3_1*Z_2_1/(Z_1_1+Z_2_1+Z_3_1)
                           // Impedance after star
                           -delta transformation(p.u)
69 Z_7_1 = X_st_G1_pu+Z_4_1
                           //
                           Impedance(p.u)
70 Z_8_1 = X_st_G1_pu+Z_5_1
                           //
                           Impedance(p.u)

```

```

71 Z_9_1 = Z_7_1*Z_8_1/(Z_7_1+Z_8_1)                                // Impedance in
    parallel(p.u). Refer Fig E4.14(e) & E4.14(f)
72 Z_10_1 = Z_9_1+Z_6_1                                              //
    Impedance(p.u). Refer Fig E4.14(f) & E4.14(g)
73 Z_11_1 = Z_10_1*X_st_G3_pu/(Z_10_1+X_st_G3_pu)                  // Impedance in parallel(p.u).
    Refer Fig E4.14(g) & E4.14(h)
74 Z_1 = Z_11_1

    // Positive sequence impedance(p.u)
75 Z_2 = Z_1

    // Negative sequence impedance(p.u)
76 Z_0 = X_g0_G3_pu+3.0*X_gn_G3_pu                                  // Zero
    sequence impedance(p.u)
77 E_g = 1.0

    // Voltage(p.u)
78 I_f_pu = 3*E_g/(Z_1+Z_2+Z_0)                                     // L-G fault
    current(p.u)
79 I_f = abs(I_f_pu)*MVA_base*1000/(3**0.5*kV_G)                   // Actual fault current(A)
80 MVA_fault = abs(I_f_pu)*MVA_base                                    // Fault MVA
81
82 // Results
83 disp("PART III - EXAMPLE : 4.7 : SOLUTION :-")
84 printf("\nFault current for a L-G fault at C = %.f A\n", I_f)
85 printf("\nNOTE: Changes in the obtained answer from
        that of textbook is due to more precision here")

```

---

### Scilab code Exa 30.8 Fault current when a single phase to earth fault occurs

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 4: UNSYMMETRICAL FAULTS IN POWER SYSTEMS
8
9 // EXAMPLE : 4.8 :
10 // Page number 518-519
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 kV_G = 11.0           // Generator rating (kV)
15 X_1_G = %i*0.1        // Positive sequence
    reactance of generator (p.u)
16 X_2_G = %i*0.1        // Negative sequence
    reactance of generator (p.u)
17 X_0_G = %i*0.02       // Zero sequence reactance
    of generator (p.u)
18 Z = 1.0               // Earthing resistor (ohm)
19 X_1_T1 = %i*0.1        // Positive sequence
    reactance of 2-winding transformer (p.u)
20 X_2_T1 = %i*0.1        // Negative sequence
    reactance of 2-winding transformer (p.u)
21 X_0_T1 = %i*0.1        // Zero sequence reactanc
    of 2-winding transformere (p.u)
22 X_1_T2_hv = %i*0.05     // Positive sequence
    reactance of hv 3-winding transformer (p.u)
23 X_2_T2_hv = %i*0.05     // Negative sequence
    reactance of hv 3-winding transformer (p.u)
```

```

24 X_0_T2_hv = %i*0.05      // Zero sequence reactanc
   of hv 3-winding transformere(p.u)
25 X_1_T2_lv_1 = %i*0.02    // Positive sequence
   reactance of lv 3-winding transformer(p.u)
26 X_2_T2_lv_1 = %i*0.02    // Negative sequence
   reactance of lv 3-winding transformer(p.u)
27 X_0_T2_lv_1 = %i*0.02    // Zero sequence reactanc
   of lv 3-winding transformere(p.u)
28 X_1_T2_lv_2 = %i*0.05    // Positive sequence
   reactance of lv 3-winding transformer(p.u)
29 X_2_T2_lv_2 = %i*0.05    // Negative sequence
   reactance of lv 3-winding transformer(p.u)
30 X_0_T2_lv_2 = %i*0.05    // Zero sequence reactanc
   of lv 3-winding transformere(p.u)
31
32 // Calculations
33 MVA_b = 10.0

      // Base MVA
34 kV_b = 11.0

      // Base voltage (kV)
35 Z_n = Z*MVA_b/kV_b**2

      // Impedance (p.u)
36 Z_1 = X_1_G+X_1_T1+X_1_T2_hv+((X_1_T2_lv_1*
   X_1_T2_lv_2)/(X_1_T2_lv_1+X_1_T2_lv_2))      //
   Positive sequence impedance(p.u)
37 Z_2 = X_2_G+X_2_T1+X_2_T2_hv+((X_2_T2_lv_1*
   X_2_T2_lv_2)/(X_2_T2_lv_1+X_2_T2_lv_2))      //
   Negative sequence impedance(p.u)
38 Z_0 = ((X_0_T1+X_0_T2_hv)*X_0_T2_lv_2/(X_0_T1+
   X_0_T2_hv+X_0_T2_lv_2))+X_0_T2_lv_1+3*Z_n ////
   Zero sequence impedance(p.u)
39 E = 1.0

      // Voltage (p.u)
40 I_f_pu = 3*E/(Z_1+Z_2+Z_0)

```

```

    // Fault current(p.u)
41 I_f = MVA_b*1000*abs(I_f_pu)/(3**0.5*kV_b)           //
                                                               Fault current (A)
42
43 // Results
44 disp("PART III - EXAMPLE : 4.8 : SOLUTION :-")
45 printf("\nFault current , I_f = %.f A\n", I_f)
46 printf("\nNOTE: Changes in the obtained answer from
          that of textbook is due to more precision here")

```

---

### Scilab code Exa 30.9 Fault currents in the lines

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 4: UNSYMMETRICAL FAULTS IN POWER SYSTEMS
8
9 // EXAMPLE : 4.9 :
10 // Page number 519
11 clear ; clc ; close ; // Clear the work space and
                           console
12
13 // Given data
14 MVA_G = 10.0           // Generator rating (MVA)
15 kV_G = 11.0            // Generator rating (kV)
16 X_1_G = 27.0           // Positive sequence reactance of
                           generator(p.u)
17 X_2_G = 9.0             // Negative sequence reactance of
                           generator(p.u)
18 X_0_G = 4.5             // Zero sequence reactance of

```

```

        generator(p.u)
19 X_1_L = 9.0      // Positive sequence reactance of
                     line upto fault(p.u)
20 X_2_L = 9.0      // Negative sequence reactance of
                     line upto fault(p.u)
21 X_0_L = 0        // Zero sequence reactance of line
                     upto fault(p.u)
22
23 // Calculations
24 E_a = kV_G*1000/3**0.5          // Phase voltage(V)
25 Z_1 = %i*(X_1_G+X_1_L)         // Positive sequence
                     reactance(p.u)
26 Z_2 = %i*(X_2_G+X_2_L)         // Negative sequence
                     reactance(p.u)
27 I_b = %i*3**0.5*E_a/(Z_1+Z_2) // Fault current in
                     line b(p.u)
28 I_c = -I_b                   // Fault current in
                     line c(p.u)
29
30 // Results
31 disp("PART III - EXAMPLE : 4.9 : SOLUTION :-")
32 printf("\nFault current in line b, I_b = %.f A", abs
        (I_b))
33 printf("\nFault current in line c, I_c = %.f A",
        real(I_c))

```

---

### Scilab code Exa 30.10 Currents in the faulted phase Current through ground and Vol

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 4: UNSYMMETRICAL FAULTS IN POWER SYSTEMS

```

```

8
9 // EXAMPLE : 4.10 :
10 // Page number 519–520
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 MVA_A = 30.0           // Alternator rating (MVA)
15 kV_A = 11.0            // Alternator rating (kV)
16 X_1 = 2.5              // Reactance to positive
   sequence current(ohm)
17 X_2 = 0.8*X_1          // Reactance to negative
   sequence current(ohm)
18 X_0 = 0.3*X_1          // Reactance to zero sequence
   current(ohm)
19
20 // Calculations
21 // Case(a)
22 a = exp(%i*120.0*pi/180)           //
   Operator
23 Z_1 = %i*X_1                  //
   Positive sequence impedance(ohm)
24 Z_2 = %i*X_2                  //
   Negative sequence impedance(ohm)
25 Z_0 = %i*X_0                  //
   sequence impedance(ohm)
26 Z_02 = Z_0*Z_2/(Z_0+Z_2)           //
   Impedance(ohm)
27 E_a = kV_A*1000/3**0.5           //
   voltage(V)
28 I_a1 = E_a/(Z_1+Z_02)           //
   Positive sequence current(A)
29 I_a2 = -Z_0/(Z_0+Z_2)*I_a1      //
   Negative sequence current(A)
30 I_a0 = -Z_2/(Z_0+Z_2)*I_a1      //
   sequence current(A)
31 I_0 = I_a0                      //
   sequence current(A)

```

```

32 I_a = I_a0+I_a1+I_a2 // Line
    current(A)
33 I_b = I_0+a**2*I_a1+a*I_a2 // Line
    current(A)
34 I_c = I_0+a*I_a1+a**2*I_a2 // Line
    current(A)
35 // Case(b)
36 I_n = 3*abs(I_0) // Current
    through ground(A)
37 // Case(c)
38 V_a2 = Z_02*I_a1 // Negative sequence voltage(V)
39 V_a = 3*abs(V_a2) // Voltage
    of healthy phase to neutral(V)
40
41 // Results
42 disp("PART III - EXAMPLE : 4.10 : SOLUTION :-")
43 printf("\nCase(a): Currents in the faulted phase are
        ")
44 printf("\n          I_a = %.f A", abs(I_a))
45 printf("\n          I_b = %.f % .1 f A", abs(I_b),
        phasemag(I_b))
46 printf("\n          I_c = %.f % .1 f A", abs(I_c),
        phasemag(I_c))
47 printf("\nCase(b): Current through ground , I_n = %.f
        A", I_n)
48 printf("\nCase(c): Voltage of healthy phase to
        neutral , V_a = %.f V\n", V_a)
49 printf("\nNOTE: Changes in the obtained answer from
        that of textbook is due to more precision here")

```

---

### Scilab code Exa 30.11 Fault currents

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar

```

```

3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 4: UNSYMMETRICAL FAULTS IN POWER SYSTEMS
8
9 // EXAMPLE : 4.11 :
10 // Page number 520–521
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 n = 6.0          // Number of alternator
15 kV_A = 6.6        // Alternator rating(kV)
16 X_1 = 0.9         // Positive sequence reactance(ohm)
17 X_2 = 0.72        // Negative sequence reactance(ohm)
18 X_0 = 0.3         // Zero sequence reactance(ohm)
19 Z_n = 0.2          // Resistance of grounding resistor(
   ohm)
20
21 // Calculations
22 E_a = kV_A*1000/3**0.5           // Phase
   voltage(V)
23 // Case(a)
24 Z_1_a = %i*X_1/n                // Positive
   sequence impedance when alternators are in
   parallel(ohm)
25 Z_2_a = %i*X_2/n                // Negative
   sequence impedance when alternators are in
   parallel(ohm)
26 Z_0_a = %i*X_0/n                // Zero
   sequence impedance when alternators are in
   parallel(ohm)
27 I_a_a = 3*E_a/(Z_1_a+Z_2_a+Z_0_a) // Fault
   current assuming 'a' phase to be fault(A)
28 // Case(b)
29 Z_0_b = 3*Z_n+%i*X_0            // Zero
   sequence impedance(ohm)

```

```

30 I_a_b = 3*E_a/(Z_1_a+Z_2_a+Z_0_b)      // Fault
      current(A)
31 // Case(c)
32 Z_0_c = %i*X_0                         // Zero
      sequence impedance(ohm)
33 I_a_c = 3*E_a/(Z_1_a+Z_2_a+Z_0_c)      // Fault
      current(A)
34
35 // Results
36 disp("PART III - EXAMPLE : 4.11 : SOLUTION :-")
37 printf("\nCase(a): Fault current if all alternator
      neutrals are solidly grounded, I_a = %.f A", imag
      (I_a_a))
38 printf("\nCase(b): Fault current if one alternator
      neutral is grounded & others isolated, I_a = %.1
      f %.1f A", abs(I_a_b), phasemag(I_a_b))
39 printf("\nCase(c): Fault current if one alternator
      neutral is solidly grounded & others isolated,
      I_a = %.2f j A\n", imag(I_a_c))
40 printf("\nNOTE: ERROR: Calculation mistakes in the
      textbook solution")

```

---

### Scilab code Exa 30.12 Fault current if all 3 phases short circuited If single line

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 4: UNSYMMETRICAL FAULTS IN POWER SYSTEMS
8
9 // EXAMPLE : 4.12 :
10 // Page number 521-522
11 clear ; clc ; close ; // Clear the work space and

```

```

        console
12
13 // Given data
14 MVA_A = 30.0          // Alternator rating (MVA)
15 kV_A = 6.6            // Alternator rating (kV)
16 X_G = 10.0            // Reactance of alternator (%)
17 kV_lv_T = 6.6         // Transformer lv side rating (kV
    )
18 kV_hv_T = 33.0       // Transformer hv side rating (kV
    )
19 X_T = 6.0             // Reactance of transformer (%)
20 kV_line = 33.0        // Transmission line voltage (kV)
21 X_line = 4.0          // Transmission line reactance (
    ohm)
22 X_g2 = 70.0           // Negative sequence reactance
    is 70% of +ve sequence reactance of generator (%)

23
24 // Calculations
25 MVA_base = 30.0        // Base MVA
26 kV_base = 6.6           // Base kV
27 Z_base = kV_base**2/MVA_base // Base
    impedance(ohm)
28 Z_g1 = %i*Z_base*X_G/100 // Positive
    sequence impedance of alternator (ohm)
29 Z_T1 = %i*Z_base*X_T/100 // Positive
    sequence impedance of transformer (ohm)
30 Z_L1 = %i*(kV_base/kV_line)**2*X_line // Positive
    sequence impedance of transmission line (ohm)
31 Z_g2 = X_g2/100*Z_g1 // Negative
    sequence impedance of alternator (ohm)
32 Z_T2 = %i*Z_base*X_T/100 // Negative
    sequence impedance of transformer (ohm)
33 Z_T0 = %i*Z_base*X_T/100 // Zero
    sequence impedance of transformer (ohm)
34 Z_L2 = Z_L1 // Negative
    sequence impedance of transmission line (ohm)
35 Z_1 = Z_g1+Z_T1+Z_L1+Z_T1 // Positive
    sequence impedance (ohm)

```

```

36 Z_2 = Z_g2+Z_T2+Z_L2+Z_T2 // Negative
      sequence impedance (ohm)
37 Z_0 = Z_T0 // Zero
      sequence impedance (ohm)
38 E_a = kV_base*1000/3**0.5 // Base
      voltage (V)
39 // Case(a)
40 I_sc = E_a/Z_1 // Fault
      current if all 3 phases short circuited (A)
41 // Case(b)
42 I_a = 3*E_a/(Z_1+Z_2+Z_0) // Fault
      current if single line is grounded assuming 'a'
      to be grounded (A)
43 // Case(c)
44 I_b = %i*3**0.5*E_a/(Z_1+Z_2) // Fault
      current for a short circuit between two lines (A)
45 I_c = -%i*3**0.5*E_a/(Z_1+Z_2) // Fault
      current for a short circuit between two lines (A)
46
47 // Results
48 disp("PART III - EXAMPLE : 4.12 : SOLUTION :-")
49 printf("\nCase(a): Fault current if all 3 phases
      short circuited , I_sc = %. f % . f A", abs(I_sc)
      ,phasemag(I_sc))
50 printf("\nCase(b): Fault current if single line is
      grounded , I_a = %. f j A", imag(I_a))
51 printf("\nCase(c): Fault current for a short circuit
      between two lines , I_b = %. f A", real(I_b))
52 printf("\n      Fault current for a short circuit
      between two lines , I_c = %. f A\n", real(I_c))
53 printf("\nNOTE: ERROR: (1). Calculation mistake in
      Z_2 in the textbook solution")
54 printf("\n      (2). Transformer reactance is
      6 percent , not 5 percent as in problem statement"
      )

```

---

### Scilab code Exa 30.13 Sub transient current in the faulty phase

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 4: UNSYMMETRICAL FAULTS IN POWER SYSTEMS
8
9 // EXAMPLE : 4.13 :
10 // Page number 522
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 kV = 6.9          // Alternator rating (kV)
15 MVA = 10.0        // Alternator rating (MVA)
16 X_st = 0.15       // Sub-transient reactance (p.u)
17 X_2 = 0.15        // Negative sequence reactance (p.u
    )
18 X_0 = 0.05        // Zero sequence reactance (p.u)
19 X = 0.397         // Grounding reactor (ohm)
20
21 // Calculations
22 MVA_base = 10.0           // Base MVA
23 kV_base = 6.9            // Base kV
24 Z_base = kV_base**2/MVA_base // Base
    impedance (ohm)
25 Z_n = X/Z_base          // Grounding
    reactor (p.u)
26 Z_1 = %i*X_st           // Positive
    sequence impedance (p.u)
27 Z_2 = %i*X_2             // Negative
```

```

        sequence impedance(p.u)
28 Z_0 = %i*(X_0+3*Z_n)                                // Zero
        sequence impedance(p.u)
29 E_a = 1.0                                              // Phase
        voltage(p.u)
30 I_a_pu = 3*E_a/(Z_1+Z_2+Z_0)                         // Sub-
        transient current in the faulty phase(p.u)
31 I_base = kV_base*1000/(3**0.5*Z_base)                // Base
        current(A)
32 I_a = abs(I_a_pu)*I_base                             // Sub-
        transient current in the faulty phase(A)
33
34 // Results
35 disp("PART III - EXAMPLE : 4.13 : SOLUTION :-")
36 printf("\nSub-transient current in the faulty phase ,
        I_a = %.f A\n", I_a)
37 printf("\nNOTE: Changes in the obtained answer from
        that of textbook is due to more precision here")

```

---

**Scilab code Exa 30.14 Initial symmetrical rms current in all phases of generator**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 4: UNSYMMETRICAL FAULTS IN POWER SYSTEMS
8
9 // EXAMPLE : 4.14 :
10 // Page number 522-523
11 clear ; clc ; close ; // Clear the work space and
        console
12
13 // Given data

```

```

14 kVA = 10000.0      // Generator rating (kVA)
15 kV = 13.8          // Generator rating (kV)
16 X_st = 10.0         // Sub-transient reactance (%)
17 X_2 = 10.0          // Negative sequence reactance (%)
18 X_0 = 5.0           // Zero sequence reactance (%)
19 X = 8.0             // Grounding reactor (%)
20 X_con = 6.0          // Reactance of reactor connecting
                         generator & transformer (%)

21
22 // Calculations
23 a = exp(%i*120.0*%pi/180)                      // Operator
24 Z_1 = %i*(X_st+X_con)/100                       // Positive
                         sequence impedance (p.u)
25 Z_2 = %i*(X_2+X_con)/100                        // Negative
                         sequence impedance (p.u)
26 Z_0 = %i*X_con/100                             // Zero
                         sequence impedance (p.u)
27 E_a = 1.0                                         // Phase
                         voltage (p.u)
28 I_a1 = E_a/(Z_1+Z_2+Z_0)                        // Sub-
                         transient current in the faulty phase(p.u)
29 I_A1 = %i*I_a1                                    // Positive
                         sequence current (p.u)
30 I_A2 = -%i*I_a1                                  // Negative
                         sequence current (p.u)
31 I_A = I_A1+I_A2                                   // Initial
                         symmetrical r.m.s current in phase a(p.u)
32 I_B1 = a**2*I_A1                                 // Positive
                         sequence current (p.u)
33 I_B2 = a*I_A2                                    // Negative
                         sequence current (p.u)
34 I_B = I_B1+I_B2                                   // Initial
                         symmetrical r.m.s current in phase b(p.u)
35 I_C1 = a*I_A1                                    // Positive
                         sequence current (p.u)
36 I_C2 = a**2*I_A2                                 // Negative
                         sequence current (p.u)
37 I_C = I_C1+I_C2                                   // Initial

```

```

        symmetrical r.m.s current in phase c(p.u)
38 I_base = kVA/(3**0.5*kV)           // Base
    current(A)
39 I_A_amp = I_A*I_base             // Initial
    symmetrical r.m.s current in phase a(p.u)
40 I_B_amp = I_B*I_base             // Initial
    symmetrical r.m.s current in phase b(p.u)
41 I_C_amp = I_C*I_base             // Initial
    symmetrical r.m.s current in phase c(p.u)
42
43 // Results
44 disp("PART III - EXAMPLE : 4.14 : SOLUTION :-")
45 printf("\nInitial symmetrical r.m.s current in all
    phases of generator are ,")
46 printf("\n I_A = %.f A", abs(I_A_amp))
47 printf("\n I_B = %. f % . f     A", abs(I_B_amp),
    phasemag(I_B_amp))
48 printf("\n I_C = %. f % . f     A", abs(I_C_amp),
    phasemag(I_C_amp))

```

---

# Chapter 32

## CIRCUIT BREAKER

Scilab code Exa 32.1 Maximum restriking voltage Frequency of transient oscillation

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 6: CIRCUIT BREAKER
8
9 // EXAMPLE : 6.1 :
10 // Page number 545
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 f = 50.0          // Generator frequency (Hz)
15 kV = 7.5         // emf to neutral rms voltage (kV)
16 X = 4.0          // Reactance of generator &
    connected system (ohm)
17 C = 0.01*10**-6 // Distributed capacitance (F)
18
19 // Calculations
```

```

20 // Case(a)
21 v = 2**0.5*kV // Active
    recovery voltage i.e phase to neutral(kV)
22 V_max_restrike = v*2 // Maximum
    restriking voltage i.e phase to neutral(kV)
23 // Case(b)
24 L = X/(2.0*%pi*f) // 
    Inductance(H)
25 f_n = 1/(2.0*%pi*(L*C)**0.5*1000) // Frequency
    of transient oscillation (kHz)
26 // Case(c)
27 t = 1.0/(2.0*f_n*1000) // Time(sec)
28 avg_rate = V_max_restrike/t // Average
    rate of rise of voltage upto first peak of
    oscillation (kV/s)

29
30 // Results
31 disp("PART III - EXAMPLE : 6.1 : SOLUTION :-")
32 printf("\nCase(a): Maximum re-striking voltage (phase
    -to-neutral) = %.1f kV", V_max_restrike)
33 printf("\nCase(b): Frequency of transient
    oscillation , f_n = %.1f kHz", f_n)
34 printf("\nCase(c): Average rate of rise of voltage
    upto first peak of oscillation = %.f kV/s \n",
    avg_rate)
35 printf("\nNOTE: Changes in the obtained answer from
    that of textbook is due to more approximation in
    the textbook")

```

---

### Scilab code Exa 32.3 Rate of rise of restriking voltage

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION

```

```

5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 6: CIRCUIT BREAKER
8
9 // EXAMPLE : 6.3 :
10 // Page number 545–546
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 kV = 132.0          // Voltage(kV)
15 pf = 0.3            // Power factor of the fault
16 K3 = 0.95           // Recovery voltage was 0.95 of
    full line value
17 f_n = 16000.0       // Natural frequency of the
    restriking transient(Hz)
18
19 // Calculations
20 kV_phase = kV/3**0.5           // System
    voltage(kV)
21 sin_phi = sind(acosd(pf))      // Sin
22 K2 = 1.0
23 v = K2*K3*kV/3**0.5*2**0.5*sin_phi // Active
    recovery voltage(kV)
24 V_max_restrike = 2*v           // Maximum
    restriking voltage(kV)
25 t = 1.0/(2.0*f_n)             // Time(sec)
26 RRRV = V_max_restrike/(t*10**6) // Rate of
    rise of restriking voltage(kV/ -sec)
27
28 // Results
29 disp("PART III – EXAMPLE : 6.3 : SOLUTION :-")
30 printf("\nRate of rise of restriking voltage , R.R.R.
    V = %.2f kV/ -sec", RRRV)

```

---

### Scilab code Exa 32.5 Voltage across the pole of a CB and Resistance to be used acr

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 6: CIRCUIT BREAKER
8
9 // EXAMPLE : 6.5 :
10 // Page number 565
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 kV = 132.0          // Voltage(kV)
15 C = 0.01*10**-6    // Phase to ground capacitance(F)
16 L = 6.0             // Inductance(H)
17 i = 5.0             // Magnetizing current(A)
18
19 // Calculations
20 V_pros = i*(L/C)**0.5/1000 // Prospective value
    of voltage(kV)
21 R = 1.0/2*(L/C)**0.5/1000 // Resistance to be
    used across the contacts to eliminate the
    restriking voltage(k-ohm)
22
23 // Results
24 disp("PART III - EXAMPLE : 6.5 : SOLUTION :-")
25 printf("\nVoltage across the pole of a CB = %.1f kV"
    , V_pros)
26 printf("\nResistance to be used across the contacts
    to eliminate the restriking voltage , R = %.2f k-
    ohm\n" , R)
27 printf("\nNOTE: ERROR: Unit of final answer R is k-
    ohm, not ohm as in the textbook solution")
```

---

### Scilab code Exa 32.6 Rated normal current Breaking current Making current and Short-time rating

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 6: CIRCUIT BREAKER
8
9 // EXAMPLE : 6.6 :
10 // Page number 567
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 I = 1200.0      // Rated normal current (A)
15 MVA = 1500.0    // Rated MVA
16 kV = 33.0       // Voltage (kV)
17
18 // Calculations
19 I_breaking = MVA/(3**0.5*kV)      // Rated symmetrical
    breaking current (kA)
20 I_making = I_breaking*2.55        // Rated making
    current (kA)
21 I_short = I_breaking            // Short-time rating (
    kA)
22
23 // Results
24 disp("PART III - EXAMPLE : 6.6 : SOLUTION :-")
25 printf("\nRated normal current = %.f A", I)
26 printf("\nBreaking current = %.2f kA (rms)", I_breaking)
27 printf("\nMaking current = %.f kA", I_making)
```

```
28 printf("\nShort-time rating = %.2f kA for 3 secs",  
I_short)
```

---

Scilab code Exa 32.8 Sustained short circuit Initial symmetrical rms current Maximum

```
1 // A Texbook on POWER SYSTEM ENGINEERING  
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  
3 // DHANPAT RAI & Co.  
4 // SECOND EDITION  
5  
6 // PART III : SWITCHGEAR AND PROTECTION  
7 // CHAPTER 6: CIRCUIT BREAKER  
8  
9 // EXAMPLE : 6.8 :  
10 // Page number 569  
11 clear ; clc ; close ; // Clear the work space and  
// console  
12  
13 // Given data  
14 kVA = 7500.0 // Rated kVA  
15 X_st = 9.0 // Sub-transient reactance (%)  
16 X_t = 15.0 // Transient reactance (%)  
17 X_d = 100.0 // Direct-axis reactance (%)  
18 kV = 13.8 // Voltage(kV) . Assumption  
19  
20 // Calculations  
21 kVA_base = 7500.0 // Base kVA  
22 kVA_sc_sustained = kVA_base/X_d*100 // Sustained  
// S.C kVA  
23 I_sc_sustained = kVA_base/(3**0.5*kV) // Sustained  
// S.C current(A). rms  
24 I_st = kVA*100/(X_st*3**0.5*kV) // Initial  
// symmetrical rms current in the breaker(A)  
25 I_max_dc = 2**0.5*I_st // Maximum  
// possible dc component of the short-circuit(A)
```

```

26 I_moment = 1.6*I_st           // Momentary
    current rating of the breaker (A)
27 I_interrupt = 1.1*I_st        // Current
    to be interrupted by the breaker (A)
28 I_kVA = 3**0.5*I_interrupt*kV // 
    Interrupting kVA
29
30 // Results
31 disp("PART III - EXAMPLE : 6.8 : SOLUTION :-")
32 printf("\nCase(a): Sustained short circuit KVA in
    the breaker = %.f kVA", kVA_sc_sustained)
33 printf("\n            Sustained short circuit current
    in the breaker = %.1f A (rms)", I_sc_sustained)
34 printf("\nCase(b): Initial symmetrical rms current
    in the breaker = %.f A (rms)", I_st)
35 printf("\nCase(c): Maximum possible dc component of
    the short-circuit in the breaker = %.f A",
    I_max_dc)
36 printf("\nCase(d): Momentary current rating of the
    breaker = %.f A (rms)", I_moment)
37 printf("\nCase(e): Current to be interrupted by the
    breaker = %.f A (rms)", I_interrupt)
38 printf("\nCase(f): Interrupting kVA = %.f kVA \n",
    I_kVA)
39 printf("\nNOTE: Changes in the obtained answer from
    that of textbook due to more approximation in
    textbook")

```

---

# Chapter 33

## PROTECTIVE RELAYS

Scilab code Exa 33.1 Time of operation of the relay

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 7: PROTECTIVE RELAYS
8
9 // EXAMPLE : 7.1 :
10 // Page number 595–596
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 I_setting = 150.0      // Current setting of IDMT(%)
15 t_mult = 0.5           // Time multiplier setting
16 ratio_CT = 500.0/5    // CT ratio
17 CT_sec = 5.0           // Secondary turn
18 I_f = 6000.0          // Fault current
19
20 // Calculations
```

```

21 I_sec_fault = I_f/ratio_CT           //  

    Secondary fault current(A)  

22 PSM = I_sec_fault/(CT_sec*I_setting/100) // Plug  

    setting multiplier  

23 t = 3.15                           // Time  

    against this PSM(sec). From graph E7.1 in  

    textbook page no 595  

24 time_oper = t*t_mult                //  

    Operating time(sec)  

25  

26 // Results  

27 disp("PART III - EXAMPLE : 7.1 : SOLUTION :-")  

28 printf("\nTime of operation of the relay = %.3f sec"  

       , time_oper)

```

---

### Scilab code Exa 33.2 Time of operation of the relay

```

1 // A Texbook on POWER SYSTEM ENGINEERING  

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  

3 // DHANPAT RAI & Co.  

4 // SECOND EDITION  

5  

6 // PART III : SWITCHGEAR AND PROTECTION  

7 // CHAPTER 7: PROTECTIVE RELAYS  

8  

9 // EXAMPLE : 7.2 :  

10 // Page number 596  

11 clear ; clc ; close ; // Clear the work space and  

    console  

12  

13 // Given data  

14 ratio = 525.0/1      // CT ratio  

15 CT_sec = 1.0         // Secondary turn  

16 t_mult = 0.3          // Time multiplier setting  

17 I_f = 5250.0        // Fault current(A)

```

```

18
19 // Calculations
20 I_sec_fault = I_f/ratio           // Secondary
   fault current (A)
21 PSM = I_sec_fault/(1.25*CT_sec) // Plug setting
   multiplier
22 t = 3.15                         // Time against
   this PSM(sec). From graph E7.1 in textbook page
   no 595
23 time_oper = t*t_mult             // Operating time
   (sec)
24
25 // Results
26 disp("PART III - EXAMPLE : 7.2 : SOLUTION :-")
27 printf("\nTime of operation of the relay = %.3f sec"
   , time_oper)

```

---

### Scilab code Exa 33.3 Operating time of feeder relay Minimum plug setting of transfe

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 7: PROTECTIVE RELAYS
8
9 // EXAMPLE : 7.3 :
10 // Page number 596
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 MVA = 20.0                  // Transformer MVA
15 overload = 30.0              // Overload of transformer (%)

```

```

16 kV = 11.0           // Bus bar rating (kV)
17 CT_trans = 1000.0/5 // Transformer CT
18 CT_cb = 400.0/5    // Circuit breaker CT
19 ps = 125.0          // Plug setting (%)
20 ts = 0.3            // Time setting
21 I_f = 5000.0        // Fault current (A)
22 t_margin = 0.5      // Discriminative time margin(
                           sec)
23
24 // Calculations
25 I_sec_fault = I_f/CT_cb                         //
   Secondary fault current (A)
26 CT_cb_sec = 5.0                                  //
   Secondary turn
27 PSM = I_sec_fault/(ps/100*CT_cb_sec)           //
   Plug setting multiplier
28 t = 2.8                                         //
   Time against this PSM(sec). From graph E7.1 in
   textbook page no 595
29 time_oper = t*ts                                //
   Operating time of feeder relay (sec)
30 I_ol = (1+(overload/100))*MVA*1000/(3**0.5*kV) //
   Overload current (A)
31 I_sec_T = I_ol/CT_trans                          //
   Secondary current (A)
32 CT_T_sec = 5.0                                  //
   Secondary turn of transformer
33 PSM_T = I_sec_T/CT_T_sec                        //
   Minimum plug setting multiplier of transformer
34 I_sec_T1 = I_f/CT_trans                         //
   Secondary fault current (A)
35 ps_T1 = 1.5                                     //
   Plug setting as per standard value
36 PSM_T1 = I_sec_T1/(CT_T_sec*ps)                //
   Plug setting multiplier of transformer
37 t_T1 = 7.0                                       //
   Time against this PSM(sec). From graph E7.1 in
   textbook page no 595

```

```

38 time_setting = (time_oper+t_margin)/t_T1           //  

    Time setting of transformer  

39  

40 // Results  

41 disp("PART III - EXAMPLE : 7.3 : SOLUTION :-")  

42 printf("\nOperating time of feeder relay = %.2f sec"  

       , time_oper)  

43 printf("\nMinimum plug setting of transformer relay ,  

       P.S > %.2f " , PSM_T)  

44 printf("\nTime setting of transformer = %.3f " ,  

       time_setting)

```

---

#### Scilab code Exa 33.4 Time of operation of the two relays

```

1 // A Texbook on POWER SYSTEM ENGINEERING  

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  

3 // DHANPAT RAI & Co.  

4 // SECOND EDITION  

5  

6 // PART III : SWITCHGEAR AND PROTECTION  

7 // CHAPTER 7: PROTECTIVE RELAYS  

8  

9 // EXAMPLE : 7.4 :  

10 // Page number 596–597  

11 clear ; clc ; close ; // Clear the work space and  

    console  

12  

13 // Given data  

14 I_f = 2000.0          // Fault current (A)  

15 ratio_CT = 200.0/1    // CT ratio  

16 R_1 = 100.0           // Relay 1 set on(%)  

17 R_2 = 125.0           // Relay 2 set on(%)  

18 t_margin = 0.5         // Discriminative time margin(  

    sec)  

19 TSM_1 = 0.2           // Time setting multiplier of

```

```

        relay 1
20
21 // Calculations
22 CT_sec = 200.0                                // CT
    secondary
23 PSM_1 = I_f*100/(CT_sec*R_1)                  // PSM of
    relay 1
24 t_1 = 2.8                                     // Time
    against this PSM(sec). From graph E7.1 in
    textbook page no 595
25 time_oper_1 = TSM_1*t_1                       // Operating
    time of relay with TSM of 0.2(Sec)
26 PSM_2 = I_f*100/(CT_sec*R_2)                  // PSM of
    relay 2
27 t_2 = 3.15                                    // Time
    against this PSM(sec). From graph E7.1 in
    textbook page no 595
28 actual_time_2 = time_oper_1+t_margin          // Actual
    time of operation of relay 2(sec)
29 TSM_2 = actual_time_2/t_2                     // Time
    setting multiplier of relay 2
30
31 // Results
32 disp("PART III - EXAMPLE : 7.4 : SOLUTION :-")
33 printf("\nTime of operation of relay 1 = %.2f sec",
    time_oper_1)
34 printf("\nActual time of operation of relay 2 = %.2f
    sec", actual_time_2)
35 printf("\nT.S.M of relay 2 = %.4f", TSM_2)

```

---

**Scilab code Exa 33.6 Will the relay operate the trip of the breaker**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.

```

```

4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 7: PROTECTIVE RELAYS
8
9 // EXAMPLE : 7.6 :
10 // Page number 611
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 I_min = 0.1           // Relay minimum pick up
    current(A)
15 slope = 10.0          // Slope characteristic(%)
16 CT_ratio = 400.0/5    // CT ratio
17 I_1 = 320.0           // Current(A)
18 I_2 = 304.0           // Current(A)
19
20 // Calculations
21 I_op_coil = (I_1-I_2)/CT_ratio           // Current
    in operating coil(A)
22 I_re_coil = 1.0*(I_1+I_2)/(2*CT_ratio)   // Current
    in restraining coil(A)
23 I_re_coil_slope = I_re_coil*slope/100    // Current
    in restraining coil with slope(A)
24
25 // Results
26 disp("PART III - EXAMPLE : 7.6 : SOLUTION :-")
27 if(I_op_coil < I_re_coil_slope) then
28     printf("\nRelay will not trip the circuit
        breaker")
29 else then
30     print("\nRelay will trip the circuit breaker")
31 end

```

---

# Chapter 34

## PROTECTION OF ALTERNATORS AND AC MOTORS

Scilab code Exa 34.1 Neutral earthing reactance

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 8: PROTECTION OF ALTERNATORS AND AC
    MOTORS
8
9 // EXAMPLE : 8.1 :
10 // Page number 624
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 6600.0          // Alternator Voltage(V)
15 P = 2000.0*10***3 // Rating of alternator(W)
```

```

16 PF = 0.8 // Power factor of alternator
17 X = 12.5 // Alternator reactance(%)
18 I = 200.0 // Current protection(A)
19 per = 10.0 // Percentage of winding
    unprotected(%)
20
21 // Calculations
22 I_f1 = P/(3**0.5*V*PF) // Full load current
    of alternator(A)
23 x = X*V/(3**0.5*100*I_f1) // Reactance per
    phase of alternator(ohm)
24 x_per = per/100*x // Reactance of 10%
    of the winding(ohm)
25 NA = V/(3**0.5*per) // Voltage induced
    in winding(V)
26 r = ((NA/I)**2-x_per**2)**0.5 // Neutral earthing
    reactance(ohm)
27
28 // Results
29 disp("PART III - EXAMPLE : 8.1 : SOLUTION :-")
30 printf("\nNeutral earthing reactance , r = %.2f ohm",
    r)

```

---

### Scilab code Exa 34.2 Unprotected portion of each phase of the stator winding against

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 8: PROTECTION OF ALTERNATORS AND AC
    MOTORS
8
9 // EXAMPLE : 8.2 :

```

```

10 // Page number 624–625
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 MVA = 20.0           // Generator rating(MVA)
15 V = 11.0*10***3      // Generator voltage(V)
16 ratio_CT = 1200.0/5    // Ratio of current
   transformer
17 I_min_op = 0.75       // Minimum operating current
   of relay(A)
18 R = 6.0              // Neutral point earthing
   resistance(ohm)
19
20 // Calculations
21 I_max_fault = ratio_CT*I_min_op          // Maximum
   fault current to operate relay(A)
22 x = I_max_fault*3***0.5*100*R/V        // Unprotected
   portion for R = 6 ohm(%)
23 R_1 = 3.0                         // Neutral
   point earthing resistance(ohm)
24 x_1 = I_max_fault*3***0.5*100*R_1/V    // Unprotected
   portion for R = 3 ohm(%)
25 R_3 = 12.0                        // Neutral
   point earthing resistance(ohm)
26 x_3 = I_max_fault*3***0.5*100*R_3/V    // Unprotected
   portion for R = 12 ohm(%)
27
28 // Results
29 disp("PART III – EXAMPLE : 8.2 : SOLUTION :-")
30 printf("\nUnprotected portion of each phase of the
   stator winding against earth fault , x = %.f
   percent" , x)
31 printf("\nEffect of varying neutral earthing
   resistance keeping relay operating current the
   same :")
32 printf("\n ( i ) R = 3 ohms")
33 printf("\n      Unprotected portion = %.1f percent"

```

```

        , x_1)
34 printf("\n      Protected portion = %.1f percent",
         (100-x_1))
35 printf("\n ( i) R = 6 ohms")
36 printf("\n      Unprotected portion = %.f percent",
         x)
37 printf("\n      Protected portion = %.f percent",
         (100-x))
38 printf("\n ( ii) R = 12 ohms")
39 printf("\n      Unprotected portion = %.f percent",
         x_3)
40 printf("\n      Protected portion = %.f percent",
         (100-x_3))

```

---

### Scilab code Exa 34.3 Portion of alternator winding unprotected

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 8: PROTECTION OF ALTERNATORS AND AC
MOTORS
8
9 // EXAMPLE : 8.3 :
10 // Page number 625
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 kVA = 5000.0          // Alternator rating (kVA)
15 V = 6600.0            // Alternator voltage (V)
16 X = 2.0               // Synchronous reactance per phase
(ohm)

```

```

17 R = 0.5          // Resistance (ohm)
18 ofb = 30.0       // Out-of-balance current (%)
19 R_n = 6.5        // Resistance of resistor earthed
                     to star point (ohm)
20
21 // Calculations
22 I_f1 = kVA*1000/(3**0.5*V)           // Full
                     load current (A)
23 I_ofb = ofb/100*I_f1                 // Out-of
                     -balance current (A)
24 x = R_n/((V/(3**0.5*100*I_ofb))-(R/100)) // 
                     Portion of winding unprotected (%)
25
26 // Results
27 disp("PART III - EXAMPLE : 8.3 : SOLUTION :-")
28 printf("\nPortion of alternator winding unprotected ,
                     x = %.1f percent", x)

```

---

**Scilab code Exa 34.4 Will the relay trip the generator CB**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 8: PROTECTION OF ALTERNATORS AND AC
                     MOTORS
8
9 // EXAMPLE : 8.4 :
10 // Page number 625
11 clear ; clc ; close ; // Clear the work space and
                     console
12
13 // Given data

```

```

14 I_min = 0.15           // Minimum pick up current of
                           relay(A)
15 slope = 12.0           // Slope(%)
16 CT_ratio = 400.0/5      // CT ratio
17 I_1 = 360.0             // Current(A)
18 I_2 = 300.0             // Current(A)
19
20 // Calculations
21 i_1 = I_1/CT_ratio      //
                           Current(A)
22 i_2 = I_2/CT_ratio      //
                           Current(A)
23 percentage = (i_1-i_2)/((i_1+i_2)/2)*100    //
                           Percentage(%)
24
25 // Results
26 disp("PART III - EXAMPLE : 8.4 : SOLUTION :-")
27 if(percentage>slope) then
28     printf("\nRelay would trip the circuit breaker,
              since the point lie on +ve torque regime")
29 else then
30     printf("\nRelay would not trip the circuit
              breaker, since the point do not lie on +ve
              torque regime")
31 end

```

---

**Scilab code Exa 34.5 Winding of each phase unprotected against earth when machine**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 8: PROTECTION OF ALTERNATORS AND AC

```

## MOTORS

```
8
9 // EXAMPLE : 8.5 :
10 // Page number 625–626
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 MVA = 50.0           // Alternator rating(MVA)
15 V = 33.0*10***3     // Alternator voltage(V)
16 CT_ratio = 2000.0/5  // CT ratio
17 R = 7.5              // Resistor earthed generator
   neutral(ohm)
18 I = 0.5              // Current above which pick up
   current(A)
19
20 // Calculations
21 I_min = CT_ratio*I    // Minimum current
   required to operate relay(A)
22 x = I_min*R/(V/3**0.5)*100 // Winding unprotected
   during normal operation(%)
23
24 // Results
25 disp("PART III – EXAMPLE : 8.5 : SOLUTION :-")
26 printf("\nWinding of each phase unprotected against
   earth when machine operates at nominal voltage , x
   = %.2f percent" , x)
```

---

**Scilab code Exa 34.6 Portion of winding unprotected**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
```

```

6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 8: PROTECTION OF ALTERNATORS AND AC
MOTORS
8
9 // EXAMPLE : 8.6 :
10 // Page number 626
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 MVA = 50.0           // Alternator rating (MVA)
15 kV = 11.0            // Alternator voltage (kV)
16 X = 2.0              // Synchronous reactance per phase
(ohm)
17 R = 0.7               // Resistance per phase(ohm)
18 R_n = 5.0             // Resistance through which
alternator is earthed(ohm)
19 ofb = 25.0            // Out-of-balance current (%)
20
21 // Calculations
22 I_f1 = MVA*1000/(3**0.5*kV)          //
Full load current (A)
23 I_ofb = ofb/100*I_f1                  //
Out-of-balance current (A)
24 x = R_n/((kV*1000/(3**0.5*100*I_ofb))-(R/100)) //
Portion of winding unprotected (%)
25
26 // Results
27 disp("PART III - EXAMPLE : 8.6 : SOLUTION :-")
28 printf("\nPortion of winding unprotected , x = %.f
percent" , x)

```

---

**Scilab code Exa 34.7 Percentage of winding that is protected against earth faults**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
```

```

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 8: PROTECTION OF ALTERNATORS AND AC
    MOTORS
8
9 // EXAMPLE : 8.7 :
10 // Page number 626–627
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 kV = 11.0          // Alternator voltage(kV)
15 MVA = 5.0          // Alternator rating(MVA)
16 X = 2.0            // Reactance per phase(ohm)
17 ofb = 35.0          // Out-of-balance current(%)
18 R_n = 5.0          // Resistance through which star
    point is earthed(ohm)
19
20 // Calculations
21 I_f1 = MVA*1000/(3**0.5*kV)           // Full
    load current(A)
22 I_ofb = ofb/100*I_f1                  // Out-of-
    balance current(A)
23 x = I_ofb*R_n*100/(kV*1000/3**0.5)    // Portion
    of winding unprotected(%)
24 protected = 100.0-x                   // Winding
    that is protected against earth faults(%)
25
26 // Results
27 disp("PART III – EXAMPLE : 8.7 : SOLUTION :-")
28 printf("\nPercentage of winding that is protected
    against earth faults = %.2f percent", protected)

```

---

### Scilab code Exa 34.8 Magnitude of neutral earthing resistance

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 8: PROTECTION OF ALTERNATORS AND AC
    MOTORS
8
9 // EXAMPLE : 8.8 :
10 // Page number 627
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 kV = 11.0          // Alternator voltage (kV)
15 P = 100.0          // Alternator maximum rating (MW)
16 PF = 0.8           // Power factor
17 X = 0.1            // Reactance of alternator (pu)
18 i = 500.0          // Current (A)
19 per = 10.0          // Windings unprotected (%)
20
21 // Calculations
22 I = P*1000/(3**0.5*kV*PF)      // Rated current of
    alternator (A)
23 a = i/I                // Relay setting
24 I_n = a*I*100/per        // Current through
    neutral (A)
25 R = kV*1000/(3**0.5*I_n)      // Magnitude of
    neutral earthing resistance (ohm)
26
27 // Results
```

```
28 disp("PART III – EXAMPLE : 8.8 : SOLUTION :-")
29 printf("\nMagnitude of neutral earthing resistance ,
      R = %.2f ohm\n", R)
30 printf("\nNOTE: ERROR: Unit of resistance is not
      mentioned in textbook solution")
```

---

# Chapter 35

## PROTECTION OF TRANSFORMERS

Scilab code Exa 35.2 Ratio of CTs

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 9: PROTECTION OF TRANSFORMERS
8
9 // EXAMPLE : 9.2 :
10 // Page number 635–636
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 V_lv = 220.0           // LV side voltage of
   transformer(V)
15 V_hv = 11000.0          // HV side voltage of
   transformer(V)
16 ratio_CT = 600.0/(5/3**0.5) // CT ratio on LV side
```

```

        of transformer
17
18 // Calculations
19 CT_pri = 600.0           // Primary CT
20 CT_sec = 5.0/3**0.5     // Secondary CT
21 I_1 = V_lv/V_hv*CT_pri // Line current in
                           secondary of transformer corresponding to primary
                           winding(A)
22 I_2 = CT_sec*3**0.5    // Current in secondary of
                           CT(A)
23
24 // Results
25 disp("PART III - EXAMPLE : 9.2 : SOLUTION :-")
26 printf("\nRatio of CTs on 11000 V side = %.f : %.f \
n", I_1, I_2)
27 printf("\nNOTE: ERROR: Mistake in representing the
final answer in textbook solution")

```

---

### Scilab code Exa 35.3 Ratio of CTs on high voltage side

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 9: PROTECTION OF TRANSFORMERS
8
9 // EXAMPLE : 9.3 :
10 // Page number 636
11 clear ; clc ; close ; // Clear the work space and
                           console
12
13 // Given data
14 V_lv = 11.0*10**3      // LV side voltage of

```

```

        transformer(V)
15 V_hv = 66.0*10**3           // HV side voltage of
        transformer(V)
16 ratio_CT = 250.0/5          // CT ratio on LV side of
        transformer
17
18 // Calculations
19 V_hv_phase = V_hv/3**0.5    // HV side phase
        voltage(V)
20 ratio_main_T = V_hv_phase/V_lv    // Ratio of main
        transformer
21 I_2 = 250.0                  // Primary CT
22 I_1 = I_2/(ratio_main_T*3**0.5) // Primary line
        current(A)
23 CT_sec = 5.0                // Secondary CT
24 secondary_side = CT_sec/3**0.5 // HV side CT
        secondary
25
26 // Results
27 disp("PART III - EXAMPLE : 9.3 : SOLUTION :-")
28 printf("\nRatio of CTs on high voltage side = %.1f :
        %.1f = (%.f/.2 f 3 ) : (%.f/ 3 ) ", I_1,
        secondary_side,I_2,ratio_main_T,CT_sec)

```

---

### Scilab code Exa 35.4 Ratio of protective CTs

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 9: PROTECTION OF TRANSFORMERS
8
9 // EXAMPLE : 9.4 :

```

```

10 // Page number 636
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 V_hv = 33.0           // HV side voltage of
   transformer(kV)
15 V_lv = 6.6            // LV side voltage of
   transformer(kV)
16 ratio_CT = 100.0/1    // CT ratio on LV side of
   transformer
17
18 // Calculations
19 CT_pri = 100.0         // Primary CT
20 CT_sec = 1.0           // Secondary CT
21 I_hv = V_lv/V_hv*CT_pri // Line current on HV
   side(A)
22 I_lv = CT_sec/3**0.5   // Line current on LV
   side(A)
23
24 // Results
25 disp("PART III - EXAMPLE : 9.4 : SOLUTION :-")
26 printf("\nRatio of protective CTs on 33 kV side = %.
   f : %.f / 3 = %.f : %.f ", I_hv, CT_sec, 3**0.5*
   I_hv, I_lv*3**0.5)

```

---

### Scilab code Exa 35.5 CT ratios on high voltage side

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 9: PROTECTION OF TRANSFORMERS

```

```

8
9 // EXAMPLE : 9.5 :
10 // Page number 636–637
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 kVA = 200.0           // Transformer rating (kVA)
15 E_1 = 11000.0          // HV side voltage of
   transformer (kV)
16 E_2 = 400.0            // LV side voltage of
   transformer (kV)
17 ratio_CT = 500.0/5      // CT ratio on LV side of
   transformer
18 I_f = 750.0             // Fault current (A)
19
20 // Calculations
21 I_2 = 500.0           // Primary CT
22 I_1 = 5.0                // Secondary CT
23 I_1_T = E_2*I_2/(3**0.5*E_1) // Primary current in
   transformer (A)
24 I_hv_T = I_1_T*3**0.5        // Equivalent line
   current on HV side (A)
25 I_pilot_lv = I_1*3**0.5        // Pilot current on LV
   side (A)
26
27 // Results
28 disp("PART III – EXAMPLE : 9.5 : SOLUTION :-")
29 printf("\nCT ratios on high voltage side = %.2f : %.
2f \n", I_hv_T, I_pilot_lv)
30 printf("\nNOTE: Circulating current is not
   calculated")

```

---

### Scilab code Exa 35.6 Suitable CT ratios

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 9: PROTECTION OF TRANSFORMERS
8
9 // EXAMPLE : 9.6 :
10 // Page number 640
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 MVA = 50.0          // Transformer rating (MVA)
15 V_hv = 132.0        // HV side voltage of transformer (
   kV)
16 V_lv = 33.0         // LV side voltage of transformer (
   kV)
17 CT_sec = 1.0         // Secondary CT rating
18
19 // Calculations
20 I_FL = MVA*1000/(3**0.5*V_lv)
   // Full-load current (A)
21 CT_ratio_33kV = I_FL/CT_sec
   // CT ratio on 33 kV side
22 CT_ratio_132kV = (I_FL*V_lv/V_hv)/(CT_sec/3**0.5)
   // CT ratio on 132 kV side
23
24 // Results
25 disp("PART III - EXAMPLE : 9.6 : SOLUTION :-")
26 printf("\nCT ratio on 33 kV side = %.f : 1 ", 
   CT_ratio_33kV)
27 printf("\nCT ratio on 132 kV side = %.f : 1 = %.
   f 3 : 1 ", CT_ratio_132kV,CT_ratio_132kV
   /3**0.5)

```

---

# Chapter 36

## PROTECTION OF TRANSMISSION LINE SHUNT INDUCTORS AND CAPACITORS

Scilab code Exa 36.1 First Second and Third zone relay setting Without infeed and

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 10: PROTECTION OF TRANSMISSION LINE ,
8 // SHUNT INDUCTORS AND CAPACITORS
9
10 // EXAMPLE : 10.1 :
11 // Page number 647–648
12 clear ; clc ; close ; // Clear the work space and
13 // Given data
```

```

14 G2_per = 70.0 // G2 is fed at 70%
    distance from A in section AB(%)
15 X_T = 10.0 // Transformer reactance(%)
16 zone_1_per = 80.0 // Setting for first zone(%)
17 zone_2_per = 50.0 // Setting for second zone(%)
18 CT_ratio = 400.0/5 // CT ratio
19 PT_ratio = 166000.0/110 // PT ratio
20 Z_AB = complex(20.0,60.0) // Section AB impedance(ohm)
21 Z_BC = complex(10.0,25.0) // Section BC impedance(ohm)
22 MVA = 10.0 // Transformer rating (MVA)
23 kV_hv = 166.0 // HV side voltage (kV)
24 kV_lv = 33.0 // LV side voltage (kV)
25
26 // Calculations
27 // Case(i) Without infeed
28 Z_sec_1 = zone_1_per/100*Z_AB*CT_ratio/PT_ratio
    // First zone setting (ohm)
29 Z_BC_hv = Z_BC*(kV_hv/kV_lv)**2
    // Z_BC on 166 kV base (ohm)
30 Z_T = %i*10*X_T*kV_hv**2/(MVA*1000)
    // Transformer impedance (ohm)
31 Z_sec_2 = (Z_AB+zone_2_per/100*Z_BC_hv+Z_T)*CT_ratio
    /PT_ratio // Second zone setting (ohm)
32 Z_sec_3 = (Z_AB+Z_BC_hv+Z_T)*CT_ratio/PT_ratio
    // Third zone setting (ohm)
33 // Case(ii) With infeed
34 I_AB = 2.0
    // Current ratio
35 Z_zone_1 = (G2_per/100*Z_AB)+I_AB*(zone_1_per-G2_per

```

```

) /100*Z_AB
    // First zone impedance(ohm)
36 Z_1 = Z_zone_1*CT_ratio/PT_ratio

    // First zone setting(ohm)
37 Z_zone_2 = (G2_per/100*Z_AB)+I_AB*((zone_1_per-
    zone_2_per)/100*Z_AB)+(zone_2_per/100*Z_BC_hv)+
    Z_T) // Second zone impedance(ohm)
38 Z_2 = Z_zone_2*CT_ratio/PT_ratio

    // Second zone setting(ohm)
39 under_reach = Z_zone_2-(Z_AB+zone_2_per/100*Z_BC_hv+
    Z_T)
        // Under-reach due to infeed(ohm)
40 Z_zone_3 = (G2_per/100*Z_AB)+I_AB*((zone_1_per-
    zone_2_per)/100*Z_AB)+Z_BC_hv+Z_T)
            // Third zone impedance(ohm)
41 Z_3 = Z_zone_3*CT_ratio/PT_ratio

    // Third zone setting(ohm)
42
43 // Results
44 disp("PART III - EXAMPLE : 10.1 : SOLUTION :-")
45 printf("\nCase(i) Without infeed:")
46 printf("\n      First zone relay setting = (%.2f
    + %.2fj) ohm", real(Z_sec_1), imag(Z_sec_1))
47 printf("\n      Second zone relay setting = (%.1f
    + %.1fj) ohm", real(Z_sec_2), imag(Z_sec_2))
48 printf("\n      Third zone relay setting = (%.1f
    + %.1fj) ohm", real(Z_sec_3), imag(Z_sec_3))
49 printf("\nCase(ii) With infeed:")
50 printf("\n      First zone relay setting = (%.3f
    + %.2fj) ohm", real(Z_1), imag(Z_1))
51 printf("\n      Second zone relay setting = (%.1f
    + %.1fj) ohm", real(Z_2), imag(Z_2))
52 printf("\n      Third zone relay setting = (%.1f
    + %.1fj) ohm\n", real(Z_3), imag(Z_3))
53 printf("\nNOTE: ERROR: Calculation mistake in Z_BC.
```

Hence , changes in the obtained answer from that of textbook”)

---

**Scilab code Exa 36.2 Impedance seen by relay and Relay setting for high speed back**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART III : SWITCHGEAR AND PROTECTION
7 // CHAPTER 10: PROTECTION OF TRANSMISSION LINE ,
SHUNT INDUCTORS AND CAPACITORS
8
9 // EXAMPLE : 10.2 :
10 // Page number 648
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 CT_ratio = 300.0/5           // CT ratio
15 PT_ratio = 166000.0/110      // PT ratio
16 Z_AB = complex(40.0,160.0)   // Section AB impedance(
ohm)
17 Z_BC = complex(7.5,15.0)     // Section BC impedance(
ohm)
18 kV_hv = 166.0                // HV side voltage(kV)
19 kV_lv = 33.0                 // LV side voltage(kV)
20 MVA = 5.0                    // Transformer rating(
MVA)
21 X_T = 6.04                   // Transformer reactance
(%)
22
23 // Calculations
24 Z_T = %i*10*X_T*kV_hv**2/(MVA*1000) // Tranformer
```

```

        impedance(ohm)
25 Z_fault = Z_AB+Z_T                                // Fault
        impedance(ohm)
26 Z_sec = Z_fault*CT_ratio/PT_ratio      // Relay
        setting for primary protection(ohm)
27 Z_BC_hv = Z_BC*(kV_hv/kV_lv)**2      // Z_BC on 166
        kV base(ohm)
28 Z = Z_AB+Z_T+Z_BC_hv                      // For backup
        protection of line BC(ohm)
29 Z_sec_set = Z*CT_ratio/PT_ratio      // Relay
        setting(ohm)
30
31 // Results
32 disp("PART III - EXAMPLE : 10.2 : SOLUTION :-")
33 printf("\nImpedance seen by relay = (%.f + %.fj) ohm
        ", real(Z_fault), imag(Z_fault))
34 printf("\nRelay setting for high speed & backup
        protection = (%.1f + %.2fj) ohm", real(Z_sec_set)
        , imag(Z_sec_set))

```

---

## Chapter 39

# INDUSTRIAL APPLICATIONS OF ELECTRIC MOTORS

Scilab code Exa 39.1 Total annual cost of group drive and Individual drive

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
    MOTORS
8
9 // EXAMPLE : 1.1 :
10 // Page number 676
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 capital_cost_group = 8000.0          // Capital cost of
    group drive(Rs)
```

```

15 n_single = 5.0 // Number of
    individual drive
16 capital_cost_single = 2500.0 // Capital cost of
    individual drive(Rs)
17 energy_cons_group = 40000.0 // Annual energy
    consumption of group drive(kWh)
18 energy_cons_single = 30000.0 // Annual energy
    consumption of group drive(kWh)
19 cost_energy = 8.0/100 // Cost of energy
    per kWh(Rs)
20 dmo_group = 12.0 // Depreciation ,
    maintenance & other fixed charges for group drive
    (%)
21 dmo_single = 18.0 // Depreciation ,
    maintenance & other fixed charges for individual
    drive(%)
22
23 // Calculations
24 // Case(a)
25 annual_cost_energy_a = energy_cons_group*cost_energy
    // Annual cost of energy(Rs)
26 dmo_cost_a = capital_cost_group*dmo_group/100
    // Depreciation , maintenance & other
    fixed charges per year for group drive(Rs)
27 yearly_cost_a = annual_cost_energy_a+dmo_cost_a
    // Total yearly cost(Rs)
28 // Case(b)
29 total_cost = capital_cost_single*n_single
    // Capital cost of individual drive(
    Rs)
30 annual_cost_energy_b = energy_cons_single*
    cost_energy // Annual cost of energy(Rs)
31 dmo_cost_b = total_cost*dmo_single/100
    // Depreciation , maintenance &
    other fixed charges per year for individual drive
    (Rs)
32 yearly_cost_b = annual_cost_energy_b+dmo_cost_b
    // Total yearly cost(Rs)

```

```

33
34 // Results
35 disp("PART IV - EXAMPLE : 1.1 : SOLUTION :-")
36 printf("\nTotal annual cost of group drive = Rs. %.f
      ", yearly_cost_a)
37 printf("\nTotal annual cost of individual drive = Rs
      . %.f ", yearly_cost_b)

```

---

**Scilab code Exa 39.2 Starting torque in terms of full load torque with star delta**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
     MOTORS
8
9 // EXAMPLE : 1.2 :
10 // Page number 680
11 clear ; clc ; close ; // Clear the work space and
     console
12
13 // Given data
14 I_sc = 6.0          // Short circuit current = 6 times
     full load current
15 s_fl = 5.0          // Full load slip (%)
16 tap = 60.0          // Auto-tranformer tapping (%)
17
18 // Calculations
19 // Case(a)
20 I_s_fl_a = I_sc/3.0           // I_s/I_fl
21 T_s_fl_a = I_s_fl_a**2*s_fl/100 // Starting
     torque in terms of full-load torque with star-

```

```

        delta starter
22 // Case(b)
23 I_s_f1_b = tap/100*I_sc           // I_s / I_f1
24 T_s_f1_b = I_s_f1_b**2*s_f1/100   // Starting
        torque in terms of full-load torque with auto-
        transformer starter
25
26 // Results
27 disp("PART IV - EXAMPLE : 1.2 : SOLUTION :-")
28 printf("\nCase(a): Starting torque in terms of full-
        load torque with star-delta starter , I_s / I_f1 = %
        .1f ", T_s_f1_a)
29 printf("\nCase(b): Starting torque in terms of full-
        load torque with auto-transformer starter , I_s /
        I_f1 = %.3f ", T_s_f1_b)

```

---

### Scilab code Exa 39.3 Tapping to be provided on an auto transformer Starting torque

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
        MOTORS
8
9 // EXAMPLE : 1.3 :
10 // Page number 680-681
11 clear ; clc ; close ; // Clear the work space and
        console
12
13 // Given data
14 V = 400.0      // IM voltage(V)
15 s_f1 = 5.0     // Full-load slip(%)

```

```

16 I_f1 = 20.0 // Full load current drawn from supply
   by IM(A)
17 Z = 2.5 // Impedance per phase(ohm)
18 I_max = 50.0 // Maximum current drawn(A)
19
20 // Calculations
21 V_phase = V/3**0.5 // Normal phase
   voltage(V)
22 P = (100**2*I_max*Z/V_phase)**0.5 // Tapping to
   be provided to auto-transformer(%)
23 I_s = I_max/(P/100) // Starting
   current taken by motor(A)
24 T_s_f1 = (I_s/I_f1)**2*s_f1/100 // Starting
   torque in terms of full-load torque
25 T_s_f1_R = (I_max/I_f1)**2*s_f1/100 // Starting
   torque in terms of full-load torque when a
   resistor is used
26
27 // Results
28 disp("PART IV – EXAMPLE : 1.3 : SOLUTION :-")
29 printf("\nTapping to be provided on an auto-
   transformer , P = %.1f percent" , P)
30 printf("\nStarting torque in terms of full-load
   torque , T_s = %.3f*T_f1 " , T_s_f1)
31 printf("\nStarting torque in terms of full-load
   torque if a resistor were used in series , T_s = %
   .4f*T_f1 " , T_s_f1_R)

```

---

### Scilab code Exa 39.4 Starting torque and Starting current if motor started by Direct

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5

```

```

6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
MOTORS
8
9 // EXAMPLE : 1.4 :
10 // Page number 681–682
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 hp = 30.0          // Power of cage IM(hp)
15 V = 500.0          // Cage IM voltage(V)
16 P = 4.0            // Number of poles
17 f = 50.0           // Frequency(Hz)
18 I_f1 = 33.0         // Full load current(A)
19 s = 4.0/100        // Slip
20 Z = 3.5             // Impedance per phase(ohm)
21 tap = 60.0          // Auto-transformer tap setting(%)
22
23 // Calculations
24 // Case(1)
25 I_s_1 = 3**0.5*(V/Z)          //
Starting current taken from line(A)
26 N_s = 120*f/P                // Speed(
rpm)
27 N_f1 = N_s-N_s*s              // Full
load speed of motor(rpm)
28 T_f1 = hp*746*60/(2*pi*N_f1) // Full
load torque(N-m)
29 T_s_1 = (I_s_1/I_f1)**2*s*T_f1 // 
Starting torque(N-m)
30 // Case(2)
31 V_ph = V/3**0.5               // Phase
voltage in star(V)
32 I_s_2 = V_ph/Z                //
Starting current(A/phase)
33 T_s_2 = (I_s_2/(I_f1/3**0.5))**2*s*T_f1 // 
Starting torque(N-m)

```

```

34 // Case(3)
35 V_ph_at = V*tap/(3**0.5*100) // Phase
   voltage of auto-transformer secondary(V)
36 V_impressed = V_ph_at*3**0.5 // 
   Volatage impressed on delta-connected stator(V)
37 I_s_3 = V_impressed/Z // 
   Starting current(A/phase)
38 I_s_line = 3**0.5*I_s_3 // Motor
   starting line current from auto-transformer
   secondary(A)
39 I_s_line_3 = tap/100*I_s_line // 
   Starting current taken from supply(A)
40 T_s_3 = (I_s_3/(I_f1/3**0.5))**2*s*T_f1 // 
   Starting torque(N-m)
41 // Case(4)
42 I_s_4 = 3**0.5*V/Z // 
   Starting current from line(A)
43 T_s_4 = T_f1*s*(I_s_4/I_f1)**2 // 
   Starting torque(N-m)
44
45 // Results
46 disp("PART IV - EXAMPLE : 1.4 : SOLUTION :-")
47 printf("\nCase(1): Starting torque for direct
   switching , T_s = %.f N-m", T_s_1)
48 printf("\n      Starting current taken from
   supply line for direct switching , I_s = %.f A",
   I_s_1)
49 printf("\nCase(2): Starting torque for star-delta
   starting , T_s = %.f N-m", T_s_2)
50 printf("\n      Starting current taken from
   supply line for star-delta starting , I_s = %.1f A
   per phase", I_s_2)
51 printf("\nCase(3): Starting torque for auto-
   transformer starting , T_s = %.f N-m", T_s_3)
52 printf("\n      Starting current taken from
   supply line for auto-transformer starting , I_s =
   %.f A", I_s_line_3)
53 printf("\nCase(4): Starting torque for series-

```

```

        parallel switch , T_s = %.f N-m" , T_s_4)
54 printf("\n          Starting current taken from
           supply line for series-parallel switch , I_s = %.f
           A\n" , I_s_4)
55 printf("\nNOTE: ERROR: Calculation mistakes and more
           approximation in textbook solution")

```

---

### Scilab code Exa 39.5 Motor current per phase Current from the supply Starting torque

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
     MOTORS
8
9 // EXAMPLE : 1.5 :
10 // Page number 682
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 400.0      // IM voltage(V)
15 f = 50.0       // Frequency(Hz)
16 I_s = 5.0      // Full voltage starting current in
    terms of full load current
17 T_s = 2.0      // Full voltage starting torque in
    terms of full load torque
18 tap = 65.0     // Auto-tranformer tapping(%)
19
20 // Calculations
21 V_ph = V/3**0.5           // Phase voltage(V)
22 V_ph_motor = tap/100*V_ph // Motor phase voltage

```

```

        when auto-transformer is used(V)
23 I_ph_motor = tap/100*I_s          // Motor phase current
     in terms of full load current
24 I_1 = tap/100*I_ph_motor         // Line current from
     supply in terms of full load current
25 T = (tap/100)**2*T_s            // Starting torque in
     terms of full load current
26 V_applied = V_ph/2**0.5         // Voltage to be
     applied to develop full-load torque(V)
27 I_line = V_applied/V_ph*I_s     // Line current in
     terms of full load current
28
29 // Results
30 disp("PART IV - EXAMPLE : 1.5 : SOLUTION :-")
31 printf("\nCase(i): Motor current per phase = %.2f * "
     I_fl ", I_ph_motor)
32 printf("\nCase(ii): Current from the supply , I_1 = "
     "% .2f * I_fl ", I_1)
33 printf("\nCase(iii): Starting torque with auto-
     transformer starter , T = %.3f * T_fl ", T)
34 printf("\nVoltage to be applied if motor has to
     develop full-load torque at starting , V = %.f V",
     V_applied)
35 printf("\nLine current from the supply to develop
     full-load torque at starting = %.2f * I_fl ",
     I_line)

```

---

### Scilab code Exa 39.6 Ratio of starting current to full load current

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION

```

```

7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
   MOTORS
8
9 // EXAMPLE : 1.6 :
10 // Page number 682
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 hp = 10.0      // IM rating (hp)
15 V = 400.0      // IM voltage (V)
16 pf = 0.8       // Lagging power factor
17 n = 0.9        // Efficiency of IM
18 I_sc = 7.2     // Short-circuit current at 160V(A)
19 V_sc = 160.0    // Voltage at short-circuit (V)
20
21 // Calculations
22 I_f1 = hp*746/(3**0.5*V*pf*n)      // Full-load line
   current (A)
23 I_sc_fv = V/V_sc*I_sc               // Short-circuit
   current at full voltage (A)
24 I_s = I_sc_fv/3.0                  // Starting current
   with star-delta starter (A)
25 I_s_f1 = I_s/I_f1                 // Ratio of starting
   current to full load current
26
27 // Results
28 disp("PART IV - EXAMPLE : 1.6 : SOLUTION :-")
29 printf("\nRatio of starting current to full-load
   current , I_s / I_f1 = %.1f \n", I_s_f1)
30 printf("\nNOTE: ERROR: Calculation mistake in final
   answer in textbook solution")

```

---

**Scilab code Exa 39.7 Resistance to be placed in series with shunt field**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
    MOTORS
8
9 // EXAMPLE : 1.7 :
10 // Page number 685–686
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 230.0      // Voltage of DC shunt motor(V)
15 N_1 = 1000.0   // No load speed(rpm)
16 R_sh = 40.0    // Shunt resistance(ohm)
17 N_2 = 1200.0   // Speed with series resistance(rpm)
18
19 // Calculations
20 phi_2 = N_1/N_2           // Flux_2 in terms flux_1
21 I_N1 = V/R_sh            // Exciting current at 1000
    rpm(A)
22 phi_1 = 11.9             // Flux corresponding to I_N1
    (mWb)
23 phi_N2 = phi_1*phi_2     // Flux at 1200 rpm(mWb)
24 I_phi_N2 = 3.25          // Exciting current
    corresponding to phi_N2(A)
25 R = V/I_phi_N2          // Resistance in field
    circuit(ohm)
26 R_extra = R-R_sh         // Resistance to be placed in
    series with shunt field(ohm)
27
28 // Results
29 disp("PART IV – EXAMPLE : 1.7 : SOLUTION :-")
30 printf("\nResistance to be placed in series with
    shunt field = %.1f ohm", R_extra)

```

---

### Scilab code Exa 39.9 Speed and Current when field winding is shunted by a diverter

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
MOTORS
8
9 // EXAMPLE : 1.9 :
10 // Page number 686–687
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 I_f1 = 25.0      // Current without diverter(A)
15 N_1 = 500.0      // Speed of dc series motor without
diverter(rpm)
16
17 // Calculations
18 I_a2 = ((3.0/2)**0.5*I_f1**2*3/2)**0.5      // Field
current with diverter(A)
19 N_2 = I_f1*N_1*3/(2*I_a2)                      // Speed
with diverter(rpm)
20
21 // Results
22 disp("PART IV – EXAMPLE : 1.9 : SOLUTION :-")
23 printf("\nSpeed when field winding is shunted by a
diverter , N_2 = %.f rpm" , N_2)
24 printf("\nCurrent when field winding is shunted by a
diverter , I_a2 = %.1f A" , I_a2)
```

---

### Scilab code Exa 39.10 Additional resistance to be inserted in the field circuit to

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
    MOTORS
8
9 // EXAMPLE : 1.10 :
10 // Page number 687
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 220.0          // DC shunt motor voltage(V)
15 I_a1 = 50.0        // Armature current at 800rpm(A)
16 N_1 = 800.0        // Speed of dc shunt motor (rpm)
17 N_2 = 1000.0       // Speed of dc shunt motor with
    additional resistance(rpm)
18 I_a2 = 75.0        // Armature current with additional
    resistance(A)
19 R_a = 0.15          // Armature resistance(ohm)
20 R_f = 250.0         // Field resistance(ohm)
21
22 // Calculations
23 E_b1 = V-R_a*I_a1           // Back emf at 800
    rpm(V)
24 I_f1 = V/R_f                // Shunt field
    current(A)
25 E_b2 = V-R_a*I_a2           // Back emf at
    1000 rpm(V)
```

```

26 I_f2 = E_b2*N_1*I_f1/(E_b1*N_2)      // Shunt field
      current at 1000 rpm(A)
27 R_f2 = V/I_f2                         // Field
      resistance at 1000 rpm(ohm)
28 R_add = R_f2-R_f                      // Additional
      resistance required(ohm)
29
30 // Results
31 disp("PART IV - EXAMPLE : 1.10 : SOLUTION :-")
32 printf("\nAdditional resistance to be inserted in
      the field circuit to raise the speed = %.1f ohm\n",
      R_add)
33 printf("\nNOTE: ERROR: Calculation mistake in E_b2
      in the textbook solution")

```

---

### Scilab code Exa 39.11 Speed of motor with a diverter connected in parallel with se

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
      MOTORS
8
9 // EXAMPLE : 1.11 :
10 // Page number 687
11 clear ; clc ; close ; // Clear the work space and
      console
12
13 // Given data
14 V = 220.0          // DC series motor voltage(V)
15 I_1 = 20.0          // Armature current at 800rpm(A)
16 N_1 = 800.0         // Speed of dc series motor(rpm)

```

```

17 R_div = 0.4      // Diverter resistance (ohm)
18 R_a = 0.5        // Armature resistance (ohm)
19 R_f = 0.2        // Series field resistance (ohm)
20
21 // Calculations
22 E_b1 = V-(R_a+R_f)*I_1           // Back emf at 800
   rpm(V)
23 I_2 = I_1*R_div/(R_div+R_f)     // Series field
   current at new speed(A)
24 E_b2 = V-(R_a*I_1+R_f*I_2)      // Back emf at new
   speed(V)
25 N_2 = I_1*N_1*E_b2/(I_2*E_b1)  // New speed with
   diverter (rpm)
26
27 // Results
28 disp("PART IV – EXAMPLE : 1.11 : SOLUTION :-")
29 printf("\nSpeed of motor with a diverter connected
   in parallel with series field , N_2 = %.f rpm",
   N_2)

```

---

### Scilab code Exa 39.12 Diverter resistance as a percentage of field resistance

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
   MOTORS
8
9 // EXAMPLE : 1.12 :
10 // Page number 687–688
11 clear ; clc ; close ; // Clear the work space and
   console

```

```

12
13 // Given data
14 speed_per = 15.0 // Motor speed increased by(%)
15
16 // Calculations
17 N_2 = (100+speed_per)/100 // New speed N_2(rpm
    )
18 phi_2 = 1/N_2*100 // Flux_2 in terms
    of full load flux
19 I_sc1 = 0.75 // New series field
    current in terms of I_a1
20 I_a2 = N_2 // Armature current
    in terms of I_a1
21 R_d = I_sc1/(I_a2-I_sc1)*100 // Diverter
    resistance in terms of series field resistance(%)
22
23 // Results
24 disp("PART IV - EXAMPLE : 1.12 : SOLUTION :-")
25 printf("\nDiverter resistance , R_d = %.1f percent of
    field resistance", R_d)

```

---

### Scilab code Exa 39.13 Additional resistance to be placed in the armature circuit

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
    MOTORS
8
9 // EXAMPLE : 1.13 :
10 // Page number 689
11 clear ; clc ; close ; // Clear the work space and

```

```

        console
12
13 // Given data
14 V = 250.0      // Voltage of DC shunt motor(V)
15 N_1 = 400.0    // No load speed(rpm)
16 R_a = 0.5      // Armature resistance(ohm)
17 N_2 = 200.0    // Speed with additional resistance(
                    rpm)
18 I_a = 20.0     // Armature current(A)
19
20 // Calculations
21 k_phi = (V-I_a*R_a)/N_1      // k
22 R = (V-k_phi*N_2)/I_a       // Resistance(ohm)
23 R_add = R-R_a                // Additional resistance
                                to be placed in armature circuit(ohm)
24
25 // Results
26 disp("PART IV – EXAMPLE : 1.13 : SOLUTION :-")
27 printf("\nResistance to be placed in the armature
            circuit = %.f ohm\n", R_add)
28 printf("\nNOTE: ERROR: The given data doesnt match
            with example 1.7 as mentioned in problem
            statement")

```

---

### Scilab code Exa 39.14 Resistance to be connected in series with armature to reduce

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
MOTORS
8

```

```

9 // EXAMPLE : 1.14 :
10 // Page number 689
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 400.0          // Voltage of DC shunt motor(V)
15 hp = 20.0           // Power of DC shunt motor(hp)
16 I = 44.0            // Current drawn by motor(A)
17 N_1 = 1000.0         // Speed(rpm)
18 N_2 = 800.0          // Speed with additional resistance(
    rpm)
19 R_sh = 200.0         // Shunt field resistance(ohm)
20
21 // Calculations
22 output = hp*746          // Motor output(W)
23 I_f1 = V/R_sh           // Shunt field current(A)
24 I_a1 = I-I_f1           // Armature current(A)
25 E_b1 = output/I_a1       // Back emf(V)
26 R_a = (V-E_b1)/I_a1       // Armature resistance(
    ohm)
27 I_a2 = I_a1*(N_2/N_1)**2   // Armature current at N2
    (A)
28 E_b2 = N_2/N_1*E_b1       // Back emf at N2(V)
29 r = ((V-E_b2)/I_a2)-R_a     // Resistance connected
    in series with armature(ohm)
30
31 // Results
32 disp("PART IV – EXAMPLE : 1.14 : SOLUTION :-")
33 printf("\nResistance to be connected in series with
    armature to reduce speed , r = %.2f ohm", r)

```

---

**Scilab code Exa 39.15 Ohmic value of resistor connected in the armature circuit**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
```

```

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
MOTORS
8
9 // EXAMPLE : 1.15 :
10 // Page number 690
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 hp = 15.0          // Power of DC shunt motor(hp)
15 V = 400.0          // Voltage of DC shunt motor(V)
16 N_reduce = 20.0    // Speed is to be reduced by(%)
17 I_f = 3.0          // Field current(A)
18 R_a = 0.5          // Armature resistance(ohm)
19 n = 0.85           // Efficiency of motor
20
21 // Calculations
22 motor_input = hp*746/n          // Motor input(W)
23 I = motor_input/V              // Motor current(A)
24 I_a1 = I-I_f                  // Armature current(
A)
25 I_a2 = I_a1                   // Armature current
at new speed(A)
26 E_b1 = V-I_a1*R_a            // Back emf(V)
27 E_b2 = E_b1*(100-N_reduce)/100 // Back emf at new
speed(V)
28 r = ((V-E_b2)/I_a2)-R_a      // Ohmic value of
resistor connected in the armature circuit(ohm)
29
30 // Results
31 disp("PART IV – EXAMPLE : 1.15 : SOLUTION :-")
32 printf("\nOhmic value of resistor connected in the
armature circuit , r = %.2f ohm" , r)

```

---

**Scilab code Exa 39.16 External resistance per phase added in rotor circuit to redu**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
MOTORS
8
9 // EXAMPLE : 1.16 :
10 // Page number 697–698
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 p = 6.0          // Number of poles
15 f = 50.0         // Frequency(Hz)
16 R_2 = 0.3        // Rotor resistance per phase(ohm)
17 N_1 = 960.0      // Rotor speed(rpm)
18 N_2 = 800.0      // New rotor speed with external
resistance(rpm)
19
20 // Calculations
21 N_s = 120*f/p    // Synchronous speed(rpm)
22 S_1 = (N_s-N_1)/N_s // Slip at full load
23 S_2 = (N_s-N_2)/N_s // New slip
24 R = (S_2/S_1*R_2)-R_2 // External resistance per
phase added in rotor circuit to reduce speed(ohm)
25
26 // Results
27 disp("PART IV – EXAMPLE : 1.16 : SOLUTION :-")
28 printf("\nExternal resistance per phase added in
```

rotor circuit to reduce speed ,  $R = \%.1f$  ohm" ,  $R$ )

---

**Scilab code Exa 39.17 Braking torque and Torque when motor speed has fallen**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
MOTORS
8
9 // EXAMPLE : 1.17 :
10 // Page number 699
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 hp = 50.0          // DC shunt motor rating (hp)
15 V = 440.0          // Voltage(V)
16 I_b = 150.0         // Breaking current(A)
17 N_reduce = 40.0      // Speed of motor fallen by(%)
18 R_a = 0.1           // Armature resistance(ohm)
19 I_a_fl = 100.0        // Full-load armature current(A)
20 N_fl = 600.0          // Full-load speed(rpm)
21
22 // Calculations
23 E_b = V-I_a_fl*R_a          // Back emf of
motor(V)
24 V_a = V+E_b          // Voltage
across armature when braking starts(V)
25 R_b = V_a/I_b          // Resistance
required(ohm)
26 R_extra = R_b-R_a          // Extra
```

```

    resistance required (ohm)
27 T_f1 = hp*746*60/(2*pi*N_f1)           // Full-load
      torque (N-m)
28 T_initial_b = T_f1*I_b/I_a_f1          // Initial
      breaking torque (N-m)
29 E_b2 = E_b*(100-N_reduce)/100          // Back emf at
      new speed (V)
30 I = (V+E_b2)/R_b                      // Current (A)
31 EBT = T_f1*I/I_a_f1                  // Torque when
      motor speed reduced by 40%(N-m)
32
33 // Results
34 disp("PART IV – EXAMPLE : 1.17 : SOLUTION :-")
35 printf("\nBraking torque = %.1f N-m", T_initial_b)
36 printf("\nTorque when motor speed has fallen , E.B.T
      = %.1f N-m\n", EBT)
37 printf("\nNOTE: ERROR: Calculation mistakes in the
      textbook solution")

```

---

### Scilab code Exa 39.18 Initial plugging torque and Torque at standstill

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
      MOTORS
8
9 // EXAMPLE : 1.18 :
10 // Page number 699–700
11 clear ; clc ; close ; // Clear the work space and
      console
12

```

```

13 // Given data
14 V = 400.0 // Voltage of IM(V)
15 p = 4.0 // Number of poles
16 f = 50.0 // Frequency (Hz)
17 hp = 25.0 // Power developed (hp)
18 S = 0.04 // Slip
19 R_X_2 = 1.0/4 // Ratio of rotor resistance to
    standstill reactance i.e R2/X2
20
21 // Calculations
22 N_s = 120*f/p

    // Synchronous speed (rpm)
23 N_f1 = N_s*(1-S)

    // Full load speed (rpm)
24 T_f1 = hp*735.5*60/(2*pi*N_f1*9.81) // Full-
    load torque (kg-m)
25 S_1 = 1.0

    // Slip at standstill
26 X_R_2 = 1.0/R_X_2

    // Ratio of standstill reactance to rotor
    resistance
27 T_s_f1 = S_1/S*((1+(S*X_R_2)**2)/(1+(S_1*X_R_2)**2)) // T_standstill/T_f1
28 T_standstill = T_s_f1*T_f1

    // Standstill torque (kg-m)
29 S_instant = (N_s+N_f1)/N_s

    // Slip at instant of plugging
30 T_initial = (S_instant/S)*((1+(S*X_R_2)**2)/(1+
    S_instant*X_R_2)**2)*T_f1 // Initial plugging
    torque (kg-m)
31

```

```

32 // Results
33 disp("PART IV - EXAMPLE : 1.18 : SOLUTION :-")
34 printf("\nInitial plugging torque = %.1f kg-m",
       T_initial)
35 printf("\nTorque at standstill = %.f kg-m\n",
       T_standstill)
36 printf("\nNOTE: ERROR: Calculation mistake from full
       -load torque onwards. Hence, change in obtained
       answer from that of textbook")

```

---

### Scilab code Exa 39.19 Value of resistance to be connected in motor circuit

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
      MOTORS
8
9 // EXAMPLE : 1.19 :
10 // Page number 701
11 clear ; clc ; close ; // Clear the work space and
      console
12
13 // Given data
14 T = 312.5           // Load torque (N-m)
15 N = 500.0           // Speed limit (rpm)
16 R_total = 1.0        // Total resistance of armature &
      field (ohm)
17
18 // Calculations
19 input_load = 2*pi*N*T/60          // Input from
      load (W)

```

```

20 E = 345.0 // Voltage from
    magnetization curve(V). From Fig E1.5 page no 701
21 I = 47.5 // Current from
    magnetization curve(A). From Fig E1.5 page no 701
22 R = E/I // Resistance(ohm
)
23 R_add = R-R_total // Additional
    resistance required(ohm)
24
25 // Results
26 disp("PART IV - EXAMPLE : 1.19 : SOLUTION :-")
27 printf("\nValue of resistance to be connected in
    motor circuit = %.2f ohm", R_add)

```

---

### Scilab code Exa 39.20 Current drawn by the motor from supply and Resistance required

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
MOTORS
8
9 // EXAMPLE : 1.20 :
10 // Page number 702
11 clear ; clc ; close ; // Clear the work space and
    console
12 funcprot(0)
13
14 // Given data
15 V = 500.0 // Shunt motor voltage(V)
16 load = 400.0 // Hoist load(kg)
17 speed = 2.5 // Hoist raised speed(m/sec )

```

```

18 n_motor = 0.85      // Efficiency of motor
19 n_hoist = 0.75       // Efficiency of hoist
20
21 // Calculations
22 P_output = load*speed*9.81           //
   Power output from motor(W)
23 P_input = P_output/(n_motor*n_hoist)    //
   Motor input(W)
24 I = P_input/V                         //
   Current drawn from supply(A)
25 output_G = load*speed*9.81*n_motor*n_hoist  //
   Generator output(W)
26 R = V**2/output_G                     //
   Resistance required in the armature circuit for
   rheostatic braking(ohm)
27
28 // Results
29 disp("PART IV – EXAMPLE : 1.20 : SOLUTION :-")
30 printf("\nCurrent drawn by the motor from supply = %
   .1f A", I)
31 printf("\nResistance required in the armature
   circuit for rheostatic braking , R = %.f ohm" , R)

```

---

### Scilab code Exa 39.21 One hour rating of motor

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
   MOTORS
8
9 // EXAMPLE : 1.21 :

```

```

10 // Page number 705
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 t = 1.0           // Time( hour )
15 hp = 15.0         // Motor rating (hp)
16 T = 2.0           // Time constant( hour )
17 theta_f = 40.0    // Temperature rise( C )
18
19 // Calculations
20 P = (1.0/(1-exp(-t/T)))**0.5*hp      // One-hour
   rating of motor(hp)
21
22 // Results
23 disp("PART IV – EXAMPLE : 1.21 : SOLUTION :-")
24 printf("\nOne-hour rating of motor , P = %.f hp\n", P
   )
25 printf("\nNOTE: Changes in the obtained answer from
   that of textbook is due to more approximation in
   the textbook solution")

```

---

### Scilab code Exa 39.22 Final temperature rise and Thermal time constant of the motor

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
   MOTORS
8
9 // EXAMPLE : 1.22 :
10 // Page number 706

```

```

11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 hp = 10.0                      // Motor rating(hp)
15 d = 0.7                         // Diameter of cylinder(
    m)
16 l = 1.0                          // Length of cylinder(m)
17 w = 380.0                        // Weight of motor(kgm)
18 heat_specific = 700.0           // Specific heat(J/kg/1
    C)
19 heat_dissipation = 15.0          // Outer surface heat
    dissipation rate(W/sq.cm/ C )
20 n = 0.88                         // Efficiency
21
22 // Calculations
23 output = hp*735.5
                                // Output
    of motor(W)
24 loss = (1-n)/n*output
                                // Losses(W)
25 area_cooling = %pi*d*l
                                // Cooling
    surface area(sq.m)
26 theta_m = loss/(area_cooling*heat_dissipation)
                                // Final temperature rise( C )
27 T_sec = w*heat_specific/(area_cooling*
    heat_dissipation) // Thermal time constant(sec)
28 T_hour = T_sec/3600
                                // Thermal
    time constant(hours)
29
30 // Results
31 disp("PART IV – EXAMPLE : 1.22 : SOLUTION :-")
32 printf("\nFinal temperature rise , _m = %.1f C" ,
    theta_m)
33 printf("\nThermal time constant of the motor = %.2f
    hours\n" , T_hour)

```

```
34 printf("\nNOTE: ERROR: Mistake in calculating  
thermal time constant in the textbook solution")
```

---

### Scilab code Exa 39.23 Half hour rating of motor

```
1 // A Texbook on POWER SYSTEM ENGINEERING  
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  
3 // DHANPAT RAI & Co.  
4 // SECOND EDITION  
5  
6 // PART IV : UTILIZATION AND TRACTION  
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC  
MOTORS  
8  
9 // EXAMPLE : 1.23 :  
10 // Page number 706  
11 clear ; clc ; close ; // Clear the work space and  
console  
12  
13 // Given data  
14 hp = 25.0          // Motor rating (hp)  
15 T = 100.0/60       // Heating time constant (hour)  
16 theta = 40.0        // Temperature rise ( C )  
17 t = 0.5            // Time (hour)  
18 n = 0.85           // Motor maximum efficiency  
19  
20 // Calculations  
21 output = hp*735.5/1000          //  
    Output of motor (kW)  
22 output_max = output*n          //  
    Power at maximum efficiency (kW)  
23 theta_f2 = theta/(1-exp(-t/T)) //  
    f2 ( C )  
24 loss = 1+(output/output_max)**2 //  
    Losses at 18.4 kW output in terms of W
```

```

25 P = ((theta_f2/theta*loss)-1)**0.5*output_max //  

     Half-hour rating of motor(kW)  

26 P_hp = P*1000/735.5 //  

     Half-hour rating of motor(hp)  

27  

28 // Results  

29 disp("PART IV - EXAMPLE : 1.23 : SOLUTION :-")  

30 printf("\nHalf-hour rating of motor, P = %.f kW = %  

     .1f hp (metric)\n", P,P_hp)  

31 printf("\nNOTE: ERROR: Calculation mistake from  

     final temperature rise onwards in textbook")

```

---

### Scilab code Exa 39.24 Time for which the motor can run at twice the continuously r

```

1 // A Texbook on POWER SYSTEM ENGINEERING  

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar  

3 // DHANPAT RAI & Co.  

4 // SECOND EDITION  

5  

6 // PART IV : UTILIZATION AND TRACTION  

7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC  

     MOTORS  

8  

9 // EXAMPLE : 1.24 :  

10 // Page number 706  

11 clear ; clc ; close ; // Clear the work space and  

     console  

12  

13 // Given data  

14 theta_f1 = 40.0 // Temperature rise( C )  

15 T = 100.0 // Heating time constant(min)  

16 rated_2 = 2.0 // Motor at twice the  

     continuously rating  

17  

18 // Calculations

```

```

19 loss_cu = 2.0**2 // Copper
    loss at twice full load in terms of W
20 loss_total = loss_cu+1 // Total
    losses at full load in terms of W
21 theta_f2 = theta_f1*loss_total/rated_2 // f2 (
    C )
22 t = log(1-(theta_f1/theta_f2))*(-T) // Time for
    which motor can run at twice the continuously
    rated output without overheating(min)
23
24 // Results
25 disp("PART IV - EXAMPLE : 1.24 : SOLUTION :-")
26 printf("\nMotor can run at twice the continuously
    rated output without overheating for time , t = %.
    f min", t)

```

---

**Scilab code Exa 39.25 Maximum overload that can be carried by the motor**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
    MOTORS
8
9 // EXAMPLE : 1.25 :
10 // Page number 706-707
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 kW = 20.0 // Motor output (kW)
15 theta_1 = 50.0 // Temperature rise not to be

```

```

        exceeded on overload( C )
16 t_1 = 1.0           // Time on overload( hour )
17 theta_2 = 30.0      // Temperature rise on full-load(
    C )
18 t_2 = 1.0           // Time on full-load( hour )
19 theta_3 = 40.0      // Temperature rise on full-load(
    C )
20 t_3 = 2.0           // Time on full-load( hour )
21
22 // Calculations
23 e_lambda = 1.0/3          // Obtained
    directly from textbook
24 theta_f = theta_2/(1-e_lambda)   // f ( C )
25 theta_f1 = theta_1/(1-e_lambda)  // 'f( C )
26 P = (theta_f1/theta_f)**0.5*kW // Maximum overload
    that can be carried by the motor(kW)
27
28 // Results
29 disp("PART IV – EXAMPLE : 1.25 : SOLUTION :-")
30 printf("\nMaximum overload that can be carried by
    the motor , P = %.1f kW", P)

```

---

### Scilab code Exa 39.26 Required size of continuously rated motor

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
    MOTORS
8
9 // EXAMPLE : 1.26 :
10 // Page number 707–708

```

```

11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 hp_1 = 100.0          // Motor load (hp)
15 t_1 = 10.0            // Time of operation (min)
16 hp_2 = 0               // Motor load (hp)
17 t_2 = 5.0              // Time of operation (min)
18 hp_3 = 60.0            // Motor load (hp)
19 t_3 = 8.0              // Time of operation (min)
20 hp_4 = 0               // Motor load (hp)
21 t_4 = 4.0              // Time of operation (min)
22
23 // Calculations
24 t_total = t_1+t_2+t_3+t_4
                           //
                           Total time of operation (min)
25 rms = ((hp_1**2*t_1+hp_2**2*t_2+hp_3**2*t_3+hp_4**2*
           t_4)/t_total)**0.5 // rms horsepower
26
27 // Results
28 disp("PART IV – EXAMPLE : 1.26 : SOLUTION :-")
29 printf("\nRequired size of continuously rated motor
           = %.f H.P\n", rms)
30 printf("\nNOTE: ERROR: Calculation mistake in the
           textbook")
31 printf("\n      Actual value is written here instead
           of standard values")

```

---

### Scilab code Exa 39.27 Suitable size of the motor

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION

```

```

5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
    MOTORS
8
9 // EXAMPLE : 1.27 :
10 // Page number 708
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 hp_1 = 200.0          // Motor load (hp)
15 t_1 = 5.0              // Time of operation (min)
16 hp_2 = 100.0           // Motor load (hp)
17 t_2 = 10.0             // Time of operation (min)
18 hp_3 = 0                // Motor load (hp)
19 t_3 = 3.0              // Time of operation (min)
20
21 // Calculations
22 m = hp_1/t_1
                                //
                                Slope of uniform rise power
23 t_total = t_1+t_2+t_3
                                // Total time of
                                operation (min)
24 ans = integrate(' (m*x)**2 , 'x' , 0 , t_1)
                                // Integarted uniform area upto 5
                                min
25 rms = ((ans+hp_2**2*t_2+hp_3**2*t_3)/t_total)**0.5
                                // rms horsepower
26
27 // Results
28 disp("PART IV – EXAMPLE : 1.27 : SOLUTION :-")
29 printf("\nrms horsepower = %.1f HP. Therefore , a
    motor of %.f H.P should be selected" , rms , rms+4)

```

---

**Scilab code Exa 39.28** Time taken to accelerate the motor to rated speed against fu

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
    MOTORS
8
9 // EXAMPLE : 1.28 :
10 // Page number 710
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 440.0          // DC shunt motor voltage(V)
15 hp = 50.0          // Motor rating(hp)
16 N = 600.0          // Speed(rpm)
17 I = 80.0           // Current at full-load(A)
18 I_1 = 1.1           // Lower current limit in terms of
    full current
19 I_2 = 1.5           // Upper current limit in terms of
    full current
20 J = 20.0            // Moment of inertia(kg-m^2)
21
22 // Calculations
23 T = hp*746*60/(2*pi*N)        // Full load torque of
    motor(N-m)
24 T_avg_start = (I_1+I_2)/2*T   // Average starting
    torque(N-m)
25 T_g = ((I_1+I_2)/2-1)*T       // Torque available
    for acceleration(N-m)
```

```

26 alpha = T_g/J           // Angular
      acceleration (rad/sec^2)
27 t = 2*pi*N/(60*alpha)   // Time taken to
      accelerate the motor (sec)
28
29 // Results
30 disp("PART IV – EXAMPLE : 1.28 : SOLUTION :-")
31 printf("\nTime taken to accelerate the motor to
      rated speed against full load torque , t = %.2f
      sec\n", t)
32 printf("\nNOTE: ERROR: Calculation mistake in the
      textbook solution")

```

---

**Scilab code Exa 39.29 Time taken to accelerate the motor to rated speed**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
      MOTORS
8
9 // EXAMPLE : 1.29 :
10 // Page number 710
11 clear ; clc ; close ; // Clear the work space and
      console
12
13 // Given data
14 hp = 50.0           // Motor rating (hp)
15 N = 600.0           // Speed(rpm)
16 energy = 276.0      // Stored energy (kg-m/hp)
17
18 // Calculations

```

```

19 g = 9.81
20 T = hp*746*60/(2*pi*N*g) // Full load
   torque of motor (kg-m)
21 J = hp*energy*2*g/(2*pi*N/60)**2 // Moment of
   inertia (kg-m^2)
22 alpha = T*g/J // Angular
   acceleration (rad/sec^2)
23 t = 2*pi*N/(60*alpha) // Time taken to
   accelerate the motor to rated speed (sec)
24
25 // Results
26 disp("PART IV - EXAMPLE : 1.29 : SOLUTION :-")
27 printf("\nTime taken to accelerate the motor to
   rated speed , t = %.2f sec", t)

```

---

### Scilab code Exa 39.30 Time taken to accelerate a fly wheel

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
   MOTORS
8
9 // EXAMPLE : 1.30 :
10 // Page number 710
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 J = 1270.0 // Moment of inertia of fly-wheel (kg-
   m^2)
15 N = 500.0 // Speed (rpm)

```

```

16 hp = 50.0           // Motor rating (hp)
17
18 // Calculations
19 g = 9.81
20 T = hp*746*60/(2*pi*N*g)          // Full load
   torque of motor (kg-m)
21 T_m = 2*T                      // Accelerating
   torque (kg-m)
22 alpha = T_m*g/J                // Angular
   acceleration (rad/sec^2)
23 t = 2*pi*N/(60*alpha)          // Time taken to
   accelerate a fly-wheel (sec)
24
25 // Results
26 disp("PART IV - EXAMPLE : 1.30 : SOLUTION :-")
27 printf("\nTime taken to accelerate a fly-wheel , t =
   %.1f sec", t)

```

---

**Scilab code Exa 39.31 Time taken for dc shunt motor to fall in speed with constant**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
   MOTORS
8
9 // EXAMPLE : 1.31 :
10 // Page number 710-711
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data

```

```

14 N_1 = 1000.0 // Speed of dc shunt motor(rpm)
15 N_2 = 400.0 // Speed of dc shunt motor(rpm)
16 R = 14.0 // Resistance connected across
    armature(ohm)
17 E_1 = 210.0 // EMF induced in armature at 1000
    rpm(V)
18 J = 17.0 // Moment of inertia(kg-m^2)
19 T_F = 1.0 // Frictional torque(kg-m)
20
21 // Calculations
22 g = 9.81
23 output = E_1**2/R // Motor
    output(W)
24 T_E = output*60/(2*%pi*N_1*g) // Electric
    braking torque(kg-m)
25 w_1 = 2*%pi*N_1/60 // -1 (rad
    /sec)
26 k = T_E/w_1
27 t = J/(g*k)*log(N_1/N_2) // Time
    taken for dc shunt motor to fall in speed with
    constant excitation(sec)
28 kw = T_E*N_2/N_1 // k
29 t_F = J/(g*k)*log((1+T_E)/(1+kw)) // Time for
    the same fall if frictional torque exists(sec)
30
31 // Results
32 disp("PART IV - EXAMPLE : 1.31 : SOLUTION :-")
33 printf("\nTime taken for dc shunt motor to fall in
    speed with constant excitation , t = %.1f sec", t)
34 printf("\nTime for the same fall if frictional
    torque exists , t = %.1f sec", t_F)

```

---

Scilab code Exa 39.32 Time taken and Number of revolutions made to come to standstill

1 // A Texbook on POWER SYSTEM ENGINEERING

```

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
MOTORS
8
9 // EXAMPLE : 1.32 :
10 // Page number 711
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 V = 400.0          // Voltage of synchronous motor(V)
15 p = 8.0            // Number of poles
16 J = 630.0          // Moment of inertia(kg-m^2)
17 T_E = 165.0         // Braking torque(kg-m)
18 kw_1 = 690.0        // Electric braking torque(kg-m)
19 T_F = 1.4           // Frictional torque(kg-m)
20 f = 50.0            // Frequency(Hz). Assumed normal
supply frequency
21
22 // Calculations
23 g = 9.81
24 // Case(a) Plugging
25 T_B = T_E+T_F

// Torque(kg-m)
26 beta = T_B*g/J

// Retardation(rad/sec ^2)
27 N_s = 120*f/p

// Synchronous speed(rad/sec)
28 w = 2*pi*N_s/60

// (rad/sec)

```

```

29 t_a = integrate( '-1.0/beta' , 'w' , w , 0)
                           // Time taken to
                           stop the motor(sec)
30 n_a = integrate( '-w/(2*pi*beta)' , 'w' , w , 0)
                           // Number of revolutions
31 // Case(b) Rheostatic braking
32 k = kw_1/w
33 t_b = J/(g*k)*log((T_F+kw_1)/T_F)
                           // Time taken
                           to stop the motor(sec)
34 n_b = 1.0/(2*pi*k)*(J/(g*k)*(T_F+kw_1)*(1-exp(-k*g*
t_b/J))-T_F*t_b) // Number of revolutions
35
36 // Results
37 disp("PART IV - EXAMPLE : 1.32 : SOLUTION :-")
38 printf("\nCase(a): Time taken to come to standstill
by plugging , t = %.1f sec" , t_a)
39 printf("\n      Number of revolutions made to
come to standstill by plugging , n = %.f
revolutions" , n_a)
40 printf("\nCase(b): Time taken to come to standstill
by rheostatic braking , t = %.1f sec" , t_b)
41 printf("\n      Number of revolutions made to
come to standstill by rheostatic braking , n = %.f
revolutions\n" , n_b)
42 printf("\nNOTE: ERROR: Calculation mistake in
finding number of revolution in case(a) in
textbook solution")

```

---

### Scilab code Exa 39.33 Inertia of flywheel required

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION

```

```

5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
    MOTORS
8
9 // EXAMPLE : 1.33 :
10 // Page number 712–713
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 hp = 500.0          // Rating of IM(hp)
15 N_nl = 40.0          // No-load speed (rpm)
16 S_f1 = 0.12          // Slip at full-load
17 T_l = 41500.0        // Load torque (kg-m)
18 t = 10.0             // Duration of each rolling period (
    sec)
19
20 // Calculations
21 g = 9.81
22 T_f1 = hp*746*60/(2*pi*N_nl*g*(1-S_f1))           //
    Torque at full-load (kg-m)
23 T_m = 2.0*T_f1                                         //
    Motor torque at any instant (kg-m)
24 slip = S_f1*N_nl                                       //
    (rpm)
25 slip_rad = slip*2*pi/60                                //
    (rad/sec)
26 k = slip_rad/T_f1
27 J = -g*t/(k*log(1-(T_m/T_l)))                         //
    Inertia of flywheel (kg-m^2)
28
29 // Results
30 disp("PART IV – EXAMPLE : 1.33 : SOLUTION :-")
31 printf("\nInertia of flywheel required , J = %.3e kg-
    m^2\n", J)
32 printf("\nNOTE: ERROR : J = 2.93*10^6 kg-m^2 and not
    2.93*10^5 as mentioned in the textbook solution")

```

)

---

### Scilab code Exa 39.34 Moment of inertia of the flywheel

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 1: INDUSTRIAL APPLICATIONS OF ELECTRIC
MOTORS
8
9 // EXAMPLE : 1.34 :
10 // Page number 713
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 T_l = 150.0      // Load torque(kg-m)
15 t = 15.0        // Duration of load torque(sec)
16 T_m = 85.0      // Motor torque(kg-m)
17 N = 500.0        // Speed(rpm)
18 s_f1 = 0.1       // Full-load slip
19
20 // Calculations
21 g = 9.81
22 slip = N*s_f1*2*pi/60           // Slip(rad/
sec)
23 k = slip/T_m
24 T_0 = 0                  // No-load
torque(kg-m)
25 J = -g*t/(k*log((T_l-T_m)/(T_l-T_0))) // Moment of
inertia of flywheel(kg-m^2)
26
```

```
27 // Results
28 disp("PART IV – EXAMPLE : 1.34 : SOLUTION :-")
29 printf("\nInertia of flywheel required , J = %.f kg-m
^2\n", J)
30 printf("\nNOTE: ERROR : Calculation mistake in the
textbook solution")
```

---

# Chapter 40

## HEATING AND WELDING

Scilab code Exa 40.1 Diameter Length and Temperature of the wire

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 2: HEATING AND WELDING
8
9 // EXAMPLE : 2.1 :
10 // Page number 724–725
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 P = 15.0*10**3          // Power supplied(W)
15 V = 220.0                // Voltage(V)
16 T_w = 1000.0             // Temperature of wire( C )
17 T_c = 600.0               // Temperature of charges( C )
18 k = 0.6                  // Radiatting efficiency
19 e = 0.9                  // Emissivity
20
```

```

21 // Calculations
22 rho = 1.016/10**6

    // Specific resistance (ohm-m)
23 d_square = 4*rho*P/(%pi*V**2)                                // d^2 in
    terms of 1
24 T_1 = T_w+273

    // Absolute temperature( C )
25 T_2 = T_c+273

    // Absolute temperature( C )
26 H = 5.72*10**4*k*e*((T_1/1000)**4-(T_2/1000)**4)           // Heat produced(watts/sq.m)
27 dl = P/(%pi*H)
28 l = (dl**2/d_square)**(1.0/3)                                  // Length of
    wire(m)
29 d = dl/l

    // Diameter of wire(m)
30 T_2_cold = 20.0+273                                         //

    Absolute temperature at the 20 C normal
    temperature( C )
31 T_1_cold = (H/(5.72*10**4*k*e)+(T_2_cold/1000)**4)        // Absolute temperature when
    *(1.0/4)*1000          // charge is cold( C )
32 T_1_c = T_1_cold-273                                         //

    Temperature when charge is cold( C )
33
34 // Results
35 disp("PART IV - EXAMPLE : 2.1 : SOLUTION :-")
36 printf("\nDiameter of the wire , d = %.3f cm" , d*100)
37 printf("\nLength of the wire , l = %.2f m" , l)
38 printf("\nTemperature of the wire when charge is"

```

```

    cold , T_1 = %.f C absolute = %.f C \n" ,
T_1_cold,T_1_c)
39 printf("\nNOTE: Slight changes in the obtained
      answer from that of textbook is due to more
      precision here")

```

---

### Scilab code Exa 40.2 Width and Length of nickel chrome strip

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 2: HEATING AND WELDING
8
9 // EXAMPLE : 2.2 :
10 // Page number 725
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 P = 15.0*10**3           // Power supplied (W)
15 V = 220.0                 // Voltage(V)
16 T_w = 1000.0              // Temperature of wire( C )
17 T_c = 600.0                // Temperature of charges( C
18 )                         // Radiatting efficiency
19 e = 0.9                   // Emissivity
20 thick = 0.25/1000         // Thickness of nickel-chrome
   strip (m)
21
22 // Calculations
23 rho = 1.016/10**6          // Specific

```

```

        resistance (ohm-m)
24 R = V**2/P
                                //
        Resistance (ohm)
25 l_w = R*thick/rho
                                // Length of
        strip in terms of w
26 T_1 = T_w+273
                                //
        Absolute temperature( C )
27 T_2 = T_c+273
                                //
        Absolute temperature( C )
28 H = 5.72*10**4*k*e*((T_1/1000)**4-(T_2/1000)**4)
        // Heat produced(watts/sq.m)
29 wl = P/(2*H)
30 w = (wl/l_w)**0.5
                                // Width of
        nickel-chrome strip (m)
31 l = w*l_w
                                //
        Length of nickel-chrome strip (m)
32
33 // Results
34 disp("PART IV - EXAMPLE : 2.2 : SOLUTION :-")
35 printf("\nWidth of nickel-chrome strip , w = %.3f cm"
       , w*100)
36 printf("\nLength of nickel-chrome strip , l = %.1f m"
       , l)

```

---

### Scilab code Exa 40.3 Power drawn under various connections

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.

```

```

4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 2: HEATING AND WELDING
8
9 // EXAMPLE : 2.3 :
10 // Page number 726-727
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 R = 50.0      // Resistance of each resistor in oven(
    ohm)
15 n = 6.0       // Number of resistance
16 V = 400.0     // Supply voltage(V)
17 tap = 50.0    // Auto-transformer tapping(%)
18
19 // Calculations
20 // Case(a)(i)
21 P_a_i = n*V**2/R*10**-3                         //
    Power consumption for 6 elements in parallel(kW)
22 // Case(a)(ii)
23 P_each_a_ii = V**2/(R+R)*10**-3                 //
    Power consumption in each group of 2 resistances
    in series(kW)
24 P_a_ii = n/2*P_each_a_ii                          //
    Power consumption for 3 groups(kW)
25 // Case(b)(i)
26 V_b_i = V/3**0.5                                  //
    Supply voltage against each resistance(V)
27 P_each_b_i = 2*V_b_i**2/R*10**-3                 //
    Power consumption in each branch(kW)
28 P_b_i = n/2*P_each_b_i                           //
    Power consumption for 2 elements in parallel in
    each phase(kW)
29 // Case(b)(ii)
30 V_b_ii = V/3**0.5                                //
    Supply voltage to any branch(V)

```

```

31 P_each_b_ii = V_b_ii**2/(R+R)*10**-3 //  

    Power consumption in each branch (kW)  

32 P_b_ii = n/2*P_each_b_ii //  

    Power consumption for 2 elements in series in  

    each phase (kW)  

33 // Case(c)(i)  

34 P_each_c_i = V**2/(R+R)*10**-3 //  

    Power consumption by each branch (kW)  

35 P_c_i = n/2*P_each_c_i //  

    Power consumption for 2 elements in series in  

    each branch (kW)  

36 // Case(c)(ii)  

37 P_each_c_ii = 2*V**2/R*10**-3 //  

    Power consumption by each branch (kW)  

38 P_c_ii = n/2*P_each_c_ii //  

    Power consumption for 2 elements in parallel in  

    each branch (kW)  

39 // Case(d)  

40 V_d = V*tap/100 //  

    Voltage under tapping (V)  

41 ratio_V = V_d/V //  

    Ratio of normal voltage to tapped voltage  

42 loss = ratio_V**2 //  

    Power loss in terms of normal power  

43  

44 // Results  

45 disp("PART IV – EXAMPLE : 2.3 : SOLUTION :-")  

46 printf("\nCase(a): AC Single phase 400 V supply")  

47 printf("\n      Case(i) : Power consumption for  

        6 elements in parallel = %.1f kW", P_a_i)  

48 printf("\n      Case(ii) : Power consumption for  

        3 groups in parallel with 2 element in series = %  

        .1f kW", P_a_ii)  

49 printf("\nCase(b): AC Three phase 400 V supply with  

      star combination")  

50 printf("\n      Case(i) : Power consumption for  

        2 elements in parallel in each phase = %.1f kW",  

        P_b_i)

```

```

51 printf("\n      Case(ii): Power consumption for
      2 elements in series in each phase = %.1f kW",
      P_b_ii)
52 printf("\nCase(c): AC Three phase 400 V supply with
      delta combination")
53 printf("\n      Case(i) : Power consumption for
      2 elements in series in each branch = %.1f kW",
      P_c_i)
54 printf("\n      Case(ii): Power consumption for
      2 elements in parallel in each branch = %.1f kW",
      P_c_ii)
55 printf("\nCase(d): Power loss will be %.2f of the
      values obtained as above with auto-transformer
      tapping", loss)

```

---

#### Scilab code Exa 40.4 Amount of energy required to melt brass

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 2: HEATING AND WELDING
8
9 // EXAMPLE : 2.4 :
10 // Page number 728
11 clear ; clc ; close ; // Clear the work space and
      console
12
13 // Given data
14 w_brass = 1000.0          // Weight of brass(kg)
15 time = 1.0                // Time(hour)
16 heat_sp = 0.094            // Specific heat
17 fusion = 40.0              // Latent heat of fusion(

```

```

        kcal/kg)
18 T_initial = 24.0           // Initial temperature( C )
19 melt_point = 920.0          // Melting point of brass(
                                C )
20 n = 0.65                   // Efficiency
21
22 // Calculations
23 heat_req = w_brass*heat_sp*(melt_point-T_initial)
                // Heat required to raise the temperature(
                kcal)
24 heat_mel = w_brass*fusion
                    // Heat required for
                    melting( kcal)
25 heat_total = heat_req+heat_mel
                    // Total heat required(
                    kcal)
26 energy = heat_total*1000*4.18/(10**3*3600*n)
                    // Energy input(kWh)
27 power = energy/time
                    // Power(kW)
28
29 // Results
30 disp("PART IV – EXAMPLE : 2.4 : SOLUTION :-")
31 printf("\nAmount of energy required to melt brass =
%.f kWh", energy)

```

---

**Scilab code Exa 40.5 Height up to which the crucible should be filled to obtain ma**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 2: HEATING AND WELDING

```

```

8
9 // EXAMPLE : 2.5 :
10 // Page number 728–729
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V_2 = 12.0          // Secondary voltage (V)
15 P = 30.0*10**3      // Power (W)
16 PF = 0.5            // Power factor
17
18 // Calculations
19 I_2 = P/(V_2*PF)    // Secondary current (A)
20 Z_2 = V_2/I_2        // Secondary impedance (
    ohm)
21 R_2 = Z_2*PF         // Secondary resistance (
    ohm)
22 sin_phi = (1-PF**2)**0.5
23 X_2 = Z_2*sin_phi    // Secondary reactance (
    ohm)
24 h = R_2/X_2
25 H_m = h              // Height up to which
    the crucible should be filled to obtain maximum
    heating effect in terms of H_c
26
27 // Results
28 disp("PART IV – EXAMPLE : 2.5 : SOLUTION :-")
29 printf("\nHeight up to which the crucible should be
    filled to obtain maximum heating effect , H_m = %
    .3f*H_c \n", H_m)
30 printf("\nNOTE: ERROR: Calculation mistake in
    textbook solution and P is 30 kW not 300 kW")

```

---

**Scilab code Exa 40.6 Voltage necessary for heating and Current flowing in the mate**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 2: HEATING AND WELDING
8
9 // EXAMPLE : 2.6 :
10 // Page number 732
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 l = 10.0           // Length of material(cm)
15 b = 10.0           // Breadth of material(cm)
16 t = 3.0            // Thickness of material(cm)
17 f = 20.0*10**6    // Frequency(Hz)
18 P = 400.0          // Power absorbed(W)
19 e_r = 5.0          // Relative permittivity
20 PF = 0.05          // Power factor
21
22 // Calculations
23 e_0 = 8.854*10**-12      // Absolute
   permittivity
24 A = l*b*10**-4        // Area(Sq.m)
25 C = e_0*e_r*A/(t/100) // Capacitace of
   parallel plate condenser(F)
26 X_c = 1.0/(2*pi*f*C) // Reactance of
   condenser(ohm)
27 phi = acosd(PF)       // ( )
28 R = X_c*tand(phi)    // Resistance of
   condenser(ohm)
29 V = (P*R)**0.5        // Voltage necessary
   for heating(V)
30 I_c = V/X_c          // Current flowing in
   the material(A)

```

31

```

32 // Results
33 disp("PART IV – EXAMPLE : 2.6 : SOLUTION :-")
34 printf("\nVoltage necessary for heating , V = %.f V",
       V)
35 printf("\nCurrent flowing in the material , I_c = %.2
       f A\n", I_c)
36 printf("\nNOTE: Changes in the obtained answer from
       that of textbook is due to more precision here &
       approximation in textbook")

```

---

### Scilab code Exa 40.7 Voltage applied across electrodes and Current through the mat

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 2: HEATING AND WELDING
8
9 // EXAMPLE : 2.7 :
10 // Page number 732–733
11 clear ; clc ; close ; // Clear the work space and
                           console
12
13 // Given data
14 l = 4.0           // Length of material(cm)
15 b = 2.0           // Breadth of material(cm)
16 t = 1.0           // Thickness of material(cm)
17 l_e = 20.0         // Length of area(cm)
18 b_e = 2.0           // Breadth of area(cm)
19 dis = 1.6          // Distance of separation of
                           electrode(cm)
20 f = 20.0*10**6    // Frequency(Hz)
21 P = 80.0           // Power absorbed(W)

```

```

22 e_r1 = 5.0           // Relative permittivity
23 e_r2 = 1.0           // Relative permittivity of air
24 PF = 0.05            // Power factor
25
26 // Calculations
27 e_0 = 8.854*10**-12 // Absolute
                           permittivity
28 A_1 = (l_e-1)*b_e*10**-4 // Area of one
                               electrode (sq.m)
29 A_2 = 1*b*10**-4       // Area of
                           material under electrode (sq.m)
30 d = dis*10**-2         //
                           Distance of separation of electrode (m)
31 d_1 = t*10**-2         // (m)
32 d_2 = (d-d_1)          // (m)
33 C = e_0*((A_1*e_r2/d)+(A_2/((d_1/e_r1)+(d_2/e_r2)))) // Capacitance(F)
34 X_c = 1.0/(2*pi*f*C) // Reactance(ohm
                           )
35 phi = acosd(PF)        // ( )
36 R = X_c*tand(phi)      //
                           Resistance(ohm)
37 V = (P*R)**0.5         //
                           Voltage applied across electrodes(V)
38 I_c = V/X_c            //
                           Current through the material(A)
39

```

```

40 // Results
41 disp("PART IV – EXAMPLE : 2.7 : SOLUTION :-")
42 printf("\nVoltage applied across electrodes , V = %.f
        V" , V)
43 printf("\nCurrent through the material , I_c = %.1f A
        \n" , I_c)
44 printf("\nNOTE: ERROR: Calculation mistake in the
        textbook solution")

```

---

**Scilab code Exa 40.8 Time taken to melt Power factor and Electrical efficiency of**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 2: HEATING AND WELDING
8
9 // EXAMPLE : 2.8 :
10 // Page number 736–737
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 weight = 3000.0          // Weight of steel(kg)
15 I = 5000.0               // Current(A)
16 V_arc = 60.0             // Arc voltage(V)
17 R_t = 0.003              // Resistance of transformer(
    ohm)
18 X_t = 0.005              // Reactance of transformer(
    ohm)
19 heat_sp = 0.12            // Specific heat of steel
20 heat_latent = 8.89         // Latent heat of steel(kilo-
    cal/kg)

```

```

21 t_2 = 1370.0 // Melting point of steel( C )
22 t_1 = 18.0 // Initial temperature of
               steel( C )
23 n = 0.6 // Overall efficiency
24
25 // Calculations
26 R_arc_phase = V_arc/I // Arc resistance per phase(ohm)
27 IR_t = I*R_t // Voltage drop across resistance(V)
28 IX_t = I*X_t // Voltage drop across reactance(V)
29 V = ((V_arc+IR_t)**2+IX_t**2)**0.5 // Voltage(V)
30 PF = (V_arc+IR_t)/V // Power factor
31 heat_kg = (t_2-t_1)*heat_sp+heat_latent // Amount of heat required per kg of steel(kcal)
32 heat_total = weight*heat_kg // Heat for 3 tonnes(kcal)
33 heat_actual_kcal = heat_total/n // Actual heat required(kcal)
34 heat_actual = heat_actual_kcal*1.162*10**-3 // Actual heat required(kWh)
35 P_input = 3*V*I*PF*10**-3 // Power input(kW)
36 time = heat_actual/P_input*60 // Time required(min)
37 n_select = 3*V_arc*I/(P_input*1000)*100 // Electrical efficiency(%)
38
39 // Results
40 disp("PART IV – EXAMPLE : 2.8 : SOLUTION :-")
41 printf("\nTime taken to melt 3 metric tonnes of
           steel = %.f minutes", time)
42 printf("\nPower factor of the furnace = %.2f ", PF)
43 printf("\nElectrical efficiency of the furnace = %.f
           percent\n", n_select)

```

44 **printf**("\nNOTE: ERROR: Calculation and substitution  
mistake in the textbook solution")

---

# Chapter 41

## ELECTROLYTIC AND ELECTRO METALLURGICAL PROCESSES

Scilab code Exa 41.1 Quantity of electricity and Time taken for the process

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 3: ELECTROLYTIC AND ELECTRO-METALLURGICAL
    PROCESSES
8
9 // EXAMPLE : 3.1 :
10 // Page number 747–748
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
```

```

14 l = 20.0          // Length of shaft (cm)
15 d = 10.0          // Diameter of shaft (cm)
16 thick = 1.5       // Layer of nickel (mm)
17 J = 195.0         // Current density (A/sq.m)
18 n_I = 0.92        // Current efficiency
19 g = 8.9           // Specific gravity of nickel
20
21 // Calculations
22 Wt = %pi*l*d*thick/10*g*10**-3           // Weight of
   nickel to be deposited (kg)
23 ece_nickel = 1.0954                         // Electro-
   chemical equivalent of nickel (kg/1000 Ah)
24 Q_I = Wt*1000/(ece_nickel*n_I)              // Quantity of
   electricity required (Ah)
25 time = Q_I/(%pi*l*d*10**-4*J)              // Time taken (
   hours)
26
27 // Results
28 disp("PART IV – EXAMPLE : 3.1 : SOLUTION :-")
29 printf("\nQuantity of electricity = %.f Ah", Q_I)
30 printf("\nTime taken for the process = %.f hours",
   time)

```

---

**Scilab code Exa 41.2 Annual output of refined copper and Energy consumption**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 3: ELECTROLYTIC AND ELECTRO-METALLURGICAL
   PROCESSES
8
9 // EXAMPLE : 3.2 :

```

```

10 // Page number 748
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 no_cells = 600.0      // Number of cells employed for
   copper refining
15 I = 4000.0            // Current(A)
16 V = 0.3                // Voltage per cell(V)
17 hour = 90.0           // Time of plant operation(hours
   )
18 ece_cu = 1.1844       // Electro-chemical equivalent
   of copper(kg/1000 Ah)
19
20 // Calculations
21 Ah_week = I*hour          // Ah
   per week per cell
22 Ah_year = Ah_week*52        // Ah
   per year per cell
23 Wt = no_cells*ece_cu*Ah_year/(1000*10**3)    //
   Weight of copper refined per year(tonnes)
24 energy = V*I*no_cells*hour*52/1000          //
   Energy consumed(kWh)
25 consumption = energy/Wt                  //
   Consumption(kWh/tonne)
26
27 // Results
28 disp("PART IV – EXAMPLE : 3.2 : SOLUTION :-")
29 printf("\nAnnual output of refined copper = %.f
   tonnes", Wt)
30 printf("\nEnergy consumption = %.1f kWh/tonne\n",
   consumption)
31 printf("\nNOTE: ERROR: Substitution & calculation
   mistake in the textbook solution")

```

---

### Scilab code Exa 41.3 Weight of aluminium produced from aluminium oxide

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 3: ELECTROLYTIC AND ELECTRO-METALLURGICAL
    PROCESSES
8
9 // EXAMPLE : 3.3 :
10 // Page number 748
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 hour = 24.0           // Time(hour)
15 I = 3500.0            // Average current(A)
16 n = 0.9               // Current efficiency
17 valency = 3.0          // Aluminium valency
18 w = 27.0              // Atomic weight of aluminium
19 ece_Ag = 107.98        // Electro-chemical equivalent
    of silver
20 Wt_dep = 0.00111      // Silver deposition by one
    coulomb(gm)
21
22 // Calculations
23 chemical_eq_Al = w/valency           //
    Chemical equivalent of aluminium
24 eme_Al = Wt_dep/ece_Ag*chemical_eq_Al   //
    Electro-chemical equivalent of aluminium(gm/
    coulomb)
25 Wt_Al_liberated = I*hour*3600*n*eme_Al/1000   //
    Weight of aluminium liberated(Kg)
26
27 // Results
28 disp("PART IV – EXAMPLE : 3.3 : SOLUTION :-")
```

```
29 printf("\nWeight of aluminium produced from  
aluminium oxide = %.1f kg", Wt_Al_liberated)
```

---

# Chapter 42

## ILLUMINATION

Scilab code Exa 42.2 mscp of lamp Illumination on the surface when it is normal In

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 4: ILLUMINATION
8
9 // EXAMPLE : 4.2 :
10 // Page number 753
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 lumens = 800.0          // Flux emitted by a lamp(
    lumens)
15 cp = 100.0              // cp of a lamp
16 d = 2.0                 // Distance b/w plane surface
    & lamp(m)
17 theta_ii = 45.0         // Inclined surface( )
18 theta_iii = 90.0         // Parallel rays( )
```

```

19
20 // Calculations
21 // Case(a)
22 mscp = lumens/(4.0*pi)           // mscp of lamp
23 // Case(b)
24 I_i = cp/d**2                   // Illumination
   on the surface when it is normal(lux)
25 I_ii = cp/d**2*cosd(theta_ii)    // Illumination
   on the surface when it is inclined to 45 (lux)
26 I_iii = cp/d**2*cosd(theta_iii)  // Illumination
   on the surface when it is parallel to rays(lux)
27
28 // Results
29 disp("PART IV – EXAMPLE : 4.2 : SOLUTION :-")
30 printf("\nCase(a): mscp of the lamp, mscp = %.f ", 
       mscp)
31 printf("\nCase(b): Case(i) : Illumination on the
   surface when it is normal, I = %.f lux", I_i)
32 printf("\n          Case(ii) : Illumination on the
   surface when it is inclined to 45 , I = %.3f lux
   ", I_ii)
33 printf("\n          Case(iii): Illumination on the
   surface when it is parallel to rays, I = %.f lux\
   n", abs(I_iii))
34 printf("\nNOTE: ERROR: Calculation mistake in case(a)
   ) in textbook solution")

```

---

### Scilab code Exa 42.3 Illumination at the centre Edge of surface with and Without r

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION

```

```

7 // CHAPTER 4: ILLUMINATION
8
9 // EXAMPLE : 4.3 :
10 // Page number 753–754
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 cp = 200.0          // cp of a lamp
15 per = 0.6           // Reflector directing light
16 D = 10.0            // Diameter(m)
17 h = 6.0             // Height at which lamp is hung(m)
18
19 // Calculations
20 flux = cp*4*%pi                // Flux(
    lumens)
21 I_i = cp/h**2                  //
    Illumination at the centre without reflector(lux)
22 d = (h**2+(D/2)**2)**0.5        // (m)
23 I_without = (cp/h**2)*(h/d)     //
    Illumination at the edge without reflector(lux)
24 I_with = cp*4*%pi*per/(25*%pi)   //
    Illumination at the edge with reflector(lux)
25 theta = acosd(h/d)              //
    (
)
26 w = 2.0*%pi*(1-cosd(theta/2))   //
    (steradian)
27 phi = cp*w                     //
    (lumens)
28 I_avg = phi/(25*%pi)            //
    Average illumination over the area without
    reflector(lux)
29
30 // Results
31 disp("PART IV – EXAMPLE : 4.3 : SOLUTION :-")
32 printf("\nCase(i) : Illumination at the centre
    without reflector = %.2f lux", I_i)
33 printf("\n        Illumination at the centre with

```

```

        reflector = %.1f lux", I_with)
34 printf("\nCase(ii): Illumination at the edge of the
           surface without reflector = %.2f lux", I_without)
35 printf("\n           Illumination at the edge of the
           surface with reflector = %.1f lux", I_with)
36 printf("\nAverage illumination over the area without
           the reflector , I = %.3f lux\n", I_avg)
37 printf("\nNOTE: ERROR: Slight calculation mistake &
           more approximation in textbook solution")

```

---

**Scilab code Exa 42.5 cp of the globe and Percentage of light emitted by lamp that**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 4: ILLUMINATION
8
9 // EXAMPLE : 4.5 :
10 // Page number 754
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 flux = 900.0          // Lamp emitting light(lumens)
15 D = 30.5              // Diameter of globe(cm)
16 B = 250.0*10**-3     // Uniform brightness(Ambert)
17
18 // Calculations
19 cp = %pi/4*D**2*(B/%pi)          // Candle power
20 flux_emit = cp*4*%pi               // Flux emitted
   by globe(lumens)
21 flux_abs = flux-flux_emit         // Flux absorbed

```

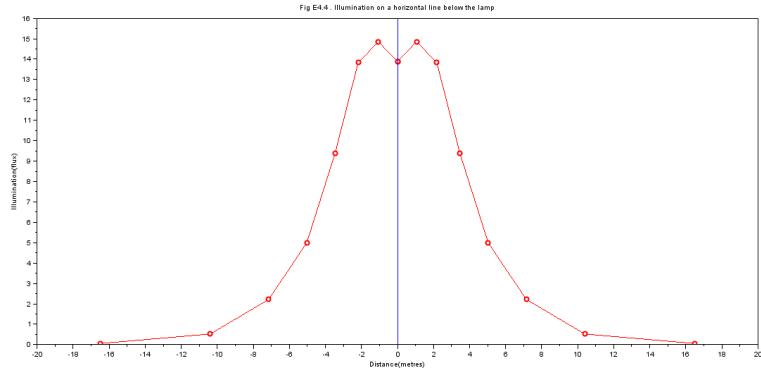


Figure 42.1: Curve showing illumination on a horizontal line below lamp

```

    by globe(lumens)
22 light_abs_per = flux_abs/flux*100 // Light absorbed
    (%)
23
24 // Results
25 disp("PART IV - EXAMPLE : 4.5 : SOLUTION :-")
26 printf("\ncp of the globe = %.f ", cp)
27 printf("\nPercentage of light emitted by lamp that
        is absorbed by the globe = %.1f percent\n",
        light_abs_per)
28 printf("\nNOTE: Changes in the obtained answer from
        that of textbook is due to more precision here &
        approximation in textbook solution")

```

---

#### Scilab code Exa 42.6 Curve showing illumination on a horizontal line below lamp

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.

```

```

4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 4: ILLUMINATION
8
9 // EXAMPLE : 4.6 :
10 // Page number 754–755
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 cp_0 = 500.0          // Candle power
15 theta_0 = 0.0          // ( )
16 cp_1 = 560.0          // Candle power
17 theta_1 = 10.0         // ( )
18 cp_2 = 600.0          // Candle power
19 theta_2 = 20.0         // ( )
20 cp_3 = 520.0          // Candle power
21 theta_3 = 30.0         // ( )
22 cp_4 = 400.0          // Candle power
23 theta_4 = 40.0         // ( )
24 cp_5 = 300.0          // Candle power
25 theta_5 = 50.0         // ( )
26 cp_6 = 150.0          // Candle power
27 theta_6 = 60.0         // ( )
28 cp_7 = 50.0           // Candle power
29 theta_7 = 70.0         // ( )
30 h = 6.0               // Height of lamp(m)
31
32 // Calculations
33 I_0 = cp_0/h**2*(cosd(theta_0))**3      // Illumination (lux)
34 l_0 = h*tand(theta_0)                      // Distance (m)
35 I_1 = cp_1/h**2*(cosd(theta_1))**3      // Illumination (lux)
36 l_1 = h*tand(theta_1)                      // Distance (m)
37 I_2 = cp_2/h**2*(cosd(theta_2))**3      // Illumination (lux)

```

```

38 l_2 = h*tand(theta_2) // Distance (m)
39 I_3 = cp_3/h**2*(cosd(theta_3))**3 //
   Illumination (lux)
40 l_3 = h*tand(theta_3) // Distance (m)
41 I_4 = cp_4/h**2*(cosd(theta_4))**3 //
   Illumination (lux)
42 l_4 = h*tand(theta_4) // Distance (m)
43 I_5 = cp_5/h**2*(cosd(theta_5))**3 //
   Illumination (lux)
44 l_5 = h*tand(theta_5) // Distance (m)
45 I_6 = cp_6/h**2*(cosd(theta_6))**3 //
   Illumination (lux)
46 l_6 = h*tand(theta_6) // Distance (m)
47 I_7 = cp_7/h**2*(cosd(theta_7))**3 //
   Illumination (lux)
48 l_7 = h*tand(theta_7) // Distance (m)
49 l = [-l_7,-l_6,-l_5,-l_4,-l_3,-l_2,-l_1,l_0,l_0,l_1,
       l_2,l_3,l_4,l_5,l_6,l_7]
50 I = [I_7,I_6,I_5,I_4,I_3,I_2,I_1,I_0,I_0,I_1,I_2,I_3
       ,I_4,I_5,I_6,I_7]
51 a = gca() ;
52 a.thickness = 2
               // sets
      thickness of plot
53 plot(l,I,'ro-') // Plot of
   illumination curve
54 x = [0,0,0,0,0,0]
55 y = [0,5,10,11,14,16]
56 plot(x,y)
               //
      Plot of straight line
57 a.x_label.text = 'Distance (metres)'
                   // labels x-axis
58 a.y_label.text = 'Illumination (flux)'
                   // labels y-axis
59 xtitle("Fig E4.4 . Illumination on a horizontal line
           below the lamp")

```

```

60 xset('thickness',2) // sets
    thickness of axes
61
62 // Results
63 disp("PART IV - EXAMPLE : 4.6 : SOLUTION :-")
64 printf("\nThe curve showing illumination on a
        horizontal line below lamp is represented in
        Figure E4.4")

```

---

**Scilab code Exa 42.7 Maximum and Minimum illumination on the floor along the centri**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 4: ILLUMINATION
8
9 // EXAMPLE : 4.7 :
10 // Page number 755
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 d = 9.15 // Lamp space(m)
15 h = 4.575 // Height(m)
16 P = 100.0 // Power(candle)
17
18 // Calculations
19 theta_3_max = 0

    // ( )
20 cos_theta_3_max_cubic = cosd(theta_3_max)**3

```

```

21 theta_4_max = atand(2)

    // ( )
22 cos_theta_4_max_cubic = cosd(theta_4_max)**3
23 theta_5_max = atand(4)

    // ( )
24 cos_theta_5_max_cubic = cosd(theta_5_max)**3
25 theta_6_max = atand(6)

    // ( )
26 cos_theta_6_max_cubic = cosd(theta_6_max)**3
27 I_max = P/h**2*(cos_theta_3_max_cubic+2*
    cos_theta_4_max_cubic+2*cos_theta_5_max_cubic+2*
    cos_theta_6_max_cubic) // Max illumination (lux)
28 theta_4_min = atand(1)

    // ( )
29 cos_theta_4_min_cubic = cosd(theta_4_min)**3
30 theta_5_min = atand(3)

    // ( )
31 cos_theta_5_min_cubic = cosd(theta_5_min)**3
32 theta_6_min = atand(5)

    // ( )
33 cos_theta_6_min_cubic = cosd(theta_6_min)**3
34 I_min = P/h**2*2*(cos_theta_4_min_cubic+
    cos_theta_5_min_cubic+cos_theta_6_min_cubic)
    // Minimum illumination (lux)
35
36 // Results
37 disp("PART IV - EXAMPLE : 4.7 : SOLUTION :-")
38 printf("\nMaximum illumination on the floor along
    the centre line = %.2f lux", I_max)
39 printf("\nMinimum illumination on the floor along
    the centre line = %.2f lux", I_min)

```

---

### Scilab code Exa 42.8 Illumination on the working plane

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 4: ILLUMINATION
8
9 // EXAMPLE : 4.8 :
10 // Page number 758
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 b = 15.25      // Breadth of workshop(m)
15 l = 36.6       // Length of workshop(m)
16 no = 20.0       // Number of lamps
17 P = 500.0       // Power of each lamp(W)
18 n = 15.0        // Luminous efficiency of each lamp(
    lumens/watt)
19 df = 0.7        // Depreciation factor
20 cou = 0.5        // Co-efficient of utilization
21
22 // Calculations
23 lumen_lamp = no*P*n                      // Lamp lumens
24 lumen_plane = lumen_lamp*df*cou           // Lumens on the
    working plane
25 I = lumen_plane/(l*b)                     // Illumination (
    lm/sq.m)
26
27 // Results
28 disp("PART IV – EXAMPLE : 4.8 : SOLUTION :-")
```

```

29 printf("\nIllumination on the working plane = %.1f
      lm per sq.m\n", I)
30 printf("\nNOTE: ERROR: The breadth should be 15.25m
      but mentioned as 5.25m in textbook statement")

```

---

### Scilab code Exa 42.9 Suitable scheme of illumination and Saving in power consumption

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 4: ILLUMINATION
8
9 // EXAMPLE : 4.9 :
10 // Page number 758–759
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 b = 27.45      // Breadth of hall(m)
15 l = 45.75      // Length of hall(m)
16 I_avg = 108.0  // Average illumination (lumens/sq .m)
17 h = 0.75       // Height (m)
18 cou = 0.35     // Co-efficient of utilization
19 pf = 0.9        // Precioation factor
20 P_f1 = 80.0    // Fluorescent lamp power(W)
21 n_100 = 13.4   // Luminous efficiency for 100W
   filament lamp(lumens/watt)
22 n_200 = 14.4   // Luminous efficiency for 200W
   filament lamp(lumens/watt)
23 n_80 = 30.0    // Luminous efficiency for 80W
   fluorescent lamp(lumens/watt)
24

```

```

25 // Calculations
26 area = b*l                                // Area
      to be illuminated (Sq.m)
27 I_total = area*I_avg                       //
      Total illumination on working plane(lumens)
28 gross_lumen = I_total/(cou*pf)              //
      Gross lumens required
29 P_required = gross_lumen/n_200              //
      Power required for illumination(W)
30 P_required_kW = P_required/1000             //
      Power required for illumination(kW)
31 no_lamp = P_required/200                   //
      Number of lamps
32 P_required_new = gross_lumen/n_80           //
      Power required when fluorescent lamp used(W)
33 P_required_new_kW = P_required_new/1000       //
      Power required when fluorescent lamp used(kW)
34 P_saving = P_required_kW-P_required_new_kW  //
      Saving in power(kW)
35
36 // Results
37 disp("PART IV – EXAMPLE : 4.9 : SOLUTION :-")
38 printf("\nSuitable scheme: Whole area divided into %
      .f rectangles & 200-watt fitting is suspended at
      centre of each rectangle", no_lamp)
39 printf("\nSaving in power consumption = %.1f kW",
      P_saving)

```

---

## Chapter 43

# ELECTRIC TRACTION SPEED TIME CURVES AND MECHANICS OF TRAIN MOVEMENT

Scilab code Exa 43.1 Maximum speed over the run

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 5: ELECTRIC TRACTION-SPEED TIME CURVES
    AND MECHANICS OF TRAIN MOVEMENT
8
9 // EXAMPLE : 5.1 :
10 // Page number 778
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
```

```

14 speed = 45.0          // Scheduled speed (kmph)
15 D = 1.5                // Distance between 2 stops (km)
16 t = 20.0                // Time of stop (sec)
17 alpha = 2.4             // Acceleration (km phps)
18 beta = 3.2              // Retardation (km phps)
19
20 // Calculations
21 t_total = D*3600/speed           // Total
   time(sec)
22 T = t_total-t                  // Actual
   time for run(sec)
23 k = (alpha+beta)/(alpha*beta)    // Constant
24 V_m = (T/k)-((T/k)**2-(7200*D/k))**0.5 // Maximum
   speed over the run (kmph)
25
26 // Results
27 disp("PART IV - EXAMPLE : 5.1 : SOLUTION :-")
28 printf("\nMaximum speed over the run , V_m = %.f kmph
   ", V_m)

```

---

### Scilab code Exa 43.2 Value of retardation

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 5: ELECTRIC TRACTION-SPEED TIME CURVES
   AND MECHANICS OF TRAIN MOVEMENT
8
9 // EXAMPLE : 5.2 :
10 // Page number 778
11 clear ; clc ; close ; // Clear the work space and
   console

```

```

12
13 // Given data
14 V_m = 65.0          // Maximum speed (kmph)
15 t = 30.0            // Time of stop (sec)
16 speed = 43.5        // Scheduled speed (kmph)
17 alpha = 1.3          // Acceleration (km phps)
18 D = 3.0              // Distance between 2 stops (km)
19
20 // Calculations
21 t_total = D*3600/speed
                           // Total time of
                           run including stop (sec)
22 T = t_total-t
                           //
                           Actual time for run (sec)
23 V_a = D/T*3600
                           //
                           Average speed (kmph)
24 beta = 1/((7200.0*D/V_m**2*((V_m/V_a)-1))-(1/alpha))
                           // Value of retardation (km phps)
25
26 // Results
27 disp("PART IV – EXAMPLE : 5.2 : SOLUTION :-")
28 printf("\nValue of retardation ,      = %.3f km phps\n"
       , beta)
29 printf("\nNOTE: Changes in the obtained answer from
           that of textbook is due to more precision here")
30 printf("\n      ERROR:      unit is km phps & not km
           phps as mentioned in textbook solution")

```

---

### Scilab code Exa 43.3 Rate of acceleration required to operate service

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.

```

```

4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 5: ELECTRIC TRACTION—SPEED TIME CURVES
    AND MECHANICS OF TRAIN MOVEMENT
8
9 // EXAMPLE : 5.3 :
10 // Page number 778–779
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 speed = 25.0          // Scheduled speed (kmph)
15 D = 800.0/1000        // Distance between 2 stations (km
    )
16 t = 20.0              // Time of stop (sec)
17 V_m_per = 20.0         // Maximum speed higher than (%)
18 beta = 3.0             // Retardation (km phps)
19
20 // Calculations
21 t_total = D*3600/speed
                           // Total time of
                           run including stop (sec)
22 T = t_total-t
                           //
                           Actual time for run (sec)
23 V_a = D/T*3600
                           //
                           Average speed (kmph)
24 V_m = (100+V_m_per)*V_a/100
                           // Maximum speed (kmph)
25 alpha = 1/((7200.0*D/V_m)**2*((V_m/V_a)-1))-(1/beta)
                           // Value of acceleration (km phps)
26
27 // Results
28 disp("PART IV – EXAMPLE : 5.3 : SOLUTION :-")
29 printf("\nRate of acceleration required to operate
    this service ,      = %.2f km phps", alpha)

```

---

### Scilab code Exa 43.4 Duration of acceleration Coasting and Braking periods

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 5: ELECTRIC TRACTION-SPEED TIME CURVES
     AND MECHANICS OF TRAIN MOVEMENT
8
9 // EXAMPLE : 5.4 :
10 // Page number 779
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 D = 2.0          // Distance between 2 stations (km)
15 V_a = 40.0       // Average speed (kmph)
16 V_1 = 60.0       // Maximum speed limitation (kph)
17 alpha = 2.0       // Acceleration (km phps)
18 beta_c = 0.15     // Coasting retardation (km phps)
19 beta = 3.0        // Braking retardation (km phps)
20
21 // Calculations
22 t_1 = V_1/alpha
                           // Time
    for acceleration(sec)
23 T = 3600*D/V_a
                           // Actual
    time of run(sec)
24 V_2 = (T-t_1-(V_1/beta_c))*beta*beta_c/(beta_c-beta)
                           // Speed at the end of coasting period (kmph)
25 t_2 = (V_1-V_2)/beta_c
```

```

                // Coasting
        period(sec)
26 t_3 = V_2/beta
                //
        Braking period(sec)
27
28 // Results
29 disp("PART IV - EXAMPLE : 5.4 : SOLUTION :-")
30 printf("\nDuration of acceleration , t_1 = %.f sec" ,
        t_1)
31 printf("\nDuration of coasting , t_2 = %.f sec" , t_2)
32 printf("\nDuration of braking , t_3 = %.f sec" , t_3)

```

---

### Scilab code Exa 43.5 Tractive resistance

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 5: ELECTRIC TRACTION-SPEED TIME CURVES
    AND MECHANICS OF TRAIN MOVEMENT
8
9 // EXAMPLE : 5.5 :
10 // Page number 781-782
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 r = 1.0      // Tractive resistance(N/tonne)
15
16 // Calculations
17 tractive_res_i = 0.278*r      // Tractive resistance(
    N/tonne) = Energy consumption(Wh/tonne-km)

```

```

18 beta = 1/277.8           // Tractive resistance (
    N/tonne) = Retardation (km kmmps/tonne)
19 energy = 98.1*1000/3600   // 1% gradient = energy
    (Wh per tonne km)

20
21 // Results
22 disp("PART IV - EXAMPLE : 5.5 : SOLUTION :-")
23 printf("\nCase(i) : Tractive resistance of 1 N per
    tonne = %.3f Wh per tonne-km", tractive_res_i)
24 printf("\nCase(ii) : Tractive resistance of 1 N per
    tonne = %.5f km phps per tonne", beta)
25 printf("\nCase(iii) : 1 percent gradient = %.2f Wh
    per tonne km\n", energy)
26 printf("\nNOTE: Slight change in the obtained answer
    from that of textbook is due to more precision
    here")

```

---

### Scilab code Exa 43.6 Torque developed by each motor

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 5: ELECTRIC TRACTION-SPEED TIME CURVES
    AND MECHANICS OF TRAIN MOVEMENT
8
9 // EXAMPLE : 5.6 :
10 // Page number 782
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 W = 254.0      // Weight of motor-coach train (tonne)

```

```

15 no = 4.0           // Number of motor
16 t_1 = 20.0          // Time(sec)
17 V_m = 40.25         // Maximum speed(kmph)
18 G = 1.0             // Gradient(%)
19 gamma = 3.5          // Gear ratio
20 n = 0.95            // Gear efficiency
21 D = 91.5/100        // Wheel diameter(m)
22 r = 44.0             // Train resistance(N/tonne)
23 I = 10.0              // Rotational inertia(%)
24
25 // Calculations
26 W_e = W*(100+I)/100           // Accelerating
      weight of train(tonne)
27 alpha = V_m/t_1                // Acceleration
      (km phps)
28 F_t = 277.8*W_e*alpha+W*r+98.1*W*G // Tractive
      effort(N)
29 T = F_t*D/(2*n*gamma)          // Torque
      developed(N-m)
30 T_each = T/no                 // Torque
      developed by each motor(N-m)
31
32 // Results
33 disp("PART IV - EXAMPLE : 5.6 : SOLUTION :-")
34 printf("\nTorque developed by each motor = %.f N-m\n",
      T_each)
35 printf("\nNOTE: Changes in the obtained answer from
      that of textbook is due to more precision here &
      more approximation in textbook")
36 printf("\n      ERROR: W = 254 tonne, not 256 tonne
      as mentioned in textbook problem statement")

```

---

**Scilab code Exa 43.7 Time taken by train to attain speed**

```
1 // A Texbook on POWER SYSTEM ENGINEERING
```

```

2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 5: ELECTRIC TRACTION—SPEED TIME CURVES
    AND MECHANICS OF TRAIN MOVEMENT
8
9 // EXAMPLE : 5.7 :
10 // Page number 782
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 W = 203.0      // Weight of motor-coach train (tonne)
15 no = 4.0       // Number of motors
16 T = 5130.0     // Shaft torque (N-m)
17 V_m = 42.0     // Maximum speed (kmph)
18 G = 100.0/250  // Gradient
19 gamma = 3.5    // Gear ratio
20 n = 0.93       // Gear efficiency
21 D = 91.5/100   // Wheel diameter (m)
22 r = 45.0        // Train resistance (N/tonne)
23 I = 10.0        // Rotational inertia (%)
24
25 // Calculations
26 W_e = W*(100+I)/100           //
    Accelerating weight of train (tonne)
27 F_t = n*4*T*2*gamma/D         //
    effort (N)
28 alpha = (F_t-W*r-98.1*W*G)/(277.8*W_e) //
    Acceleration (km phps)
29 t_1 = V_m/alpha                //
    taken by train to attain speed (sec)
30
31 // Results
32 disp("PART IV – EXAMPLE : 5.7 : SOLUTION :-")
33 printf("\nTime taken by train to attain speed , t_1 =

```

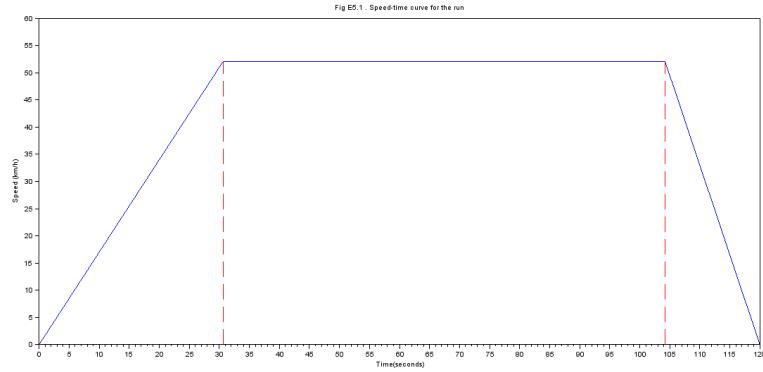


Figure 43.1: Speed Time curve for the run and Energy consumption at the axles of train

---

%.1 f sec" , t\_1)

**Scilab code Exa 43.8 Speed Time curve for the run and Energy consumption at the axles of train**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 5: ELECTRIC TRACTION-SPEED TIME CURVES
     AND MECHANICS OF TRAIN MOVEMENT
8
9 // EXAMPLE : 5.8 :
10 // Page number 782-783
11 clear ; clc ; close ; // Clear the work space and
      console
12

```

```

13 // Given data
14 V_a = 42.0 // Average speed of train (kmph)
15 D = 1400.0/1000 // Distance(km)
16 alpha = 1.7 // Acceleration(km phps)
17 beta = 3.3 // Retardation(km phps)
18 r = 50.0 // Tractive resistance(N/tonne)
19 I = 10.0 // Rotational inertia(%)
20
21 // Calculations
22 T = D*3600/V_a // Time for
run(sec)
23 k = (alpha+beta)/(alpha*beta) // Constant
24 V_m = (T/k)-((T/k)**2-(7200*D/k))**0.5 // Maximum speed over the run (kmph)
25 t_1 = V_m/alpha // Time of
acceleration (sec)
26 t_3 = V_m/beta // Time(sec
)
27 t_2 = T-(t_1+t_3) // Time(sec)
28 D_1 = D-(V_a*t_1/(2*3600)) // Distance(km)
29 We_W = (100+I)/100 // W_e/W
30 energy = (0.0107*V_m**2*We_W/D)+(0.278*r*D_1/D) // Energy consumption(Wh per tonne-km)
31 a = gca(); a.thickness = 2 // sets
thickness of plot
33 plot([0,t_1,t_1,(t_1+t_2),(t_1+t_2),(t_1+t_2+t_3)], [0,V_m,V_m,V_m,V_m,0]) // Plotting speed-
time curve
34 plot([t_1,t_1],[0,V_m], 'r--')

```

```

35 plot([t_1+t_2,t_1+t_2],[0,V_m],'r--')
36 a.x_label.text = 'Time(seconds)'
37 a.y_label.text = 'Speed(km/h)'
38 xtitle("Fig E5.1 . Speed-time curve for the run")
39 xset('thickness',2)
40 // sets
41 // thickness of axes
42 // Results
43 disp("PART IV - EXAMPLE : 5.8 : SOLUTION :-")
44 printf("\nSpeed-time curve for the run is shown in
Figure E5.1")
45 printf("\nEnergy consumption at the axles of train =
%.1f Wh per tonne-km", energy)

```

---

### Scilab code Exa 43.9 Acceleration Coasting retardation and Scheduled speed

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 5: ELECTRIC TRACTION-SPEED TIME CURVES
AND MECHANICS OF TRAIN MOVEMENT
8
9 // EXAMPLE : 5.9 :
10 // Page number 783
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 V_A = 48.0           // Speed(kmph)

```

```

15 t_1 = 24.0          // Time taken to accelerate from
   rest to speed(sec)
16 t_2 = 69.0          // Coasting time(sec)
17 r = 58.0            // Constant resistance(N/tonne)
18 beta = 3.3           // Retardation(km phps)
19 t_3 = 11.0           // Retardation time(sec)
20 t_iii_a = 20.0       // Station stop time(sec)
21 t_iii_b = 15.0       // Station stop time(sec)
22 I = 10.0             // Rotational inertia(%)
23
24 // Calculations
25 alpha = V_A/t_1

               // Acceleration(km phps)
26 V_B = beta*t_3

               // Speed at B(km phps)
27 beta_c = (V_A-V_B)/t_2
                           //
                           Retardation during coasting(km phps)
28 distance_acc = 1.0/2*t_1*V_A/3600
                           // Distance
                           covered during acceleration(km)
29 distance_coasting = (V_A**2-V_B**2)/(2*beta_c*3600)
                           // Distance covered during coasting
                           (km)
30 distance_braking = t_3*V_B/(3600*2)
                           // Distance covered
                           during braking(km)
31 distance_total = distance_acc+distance_coasting+
   distance_braking // Total distance(km)
32 speed_iii_a = distance_total*3600/(t_1+t_2+t_3+
   t_iii_a)           // Scheduled speed with a stop
   of 20 sec(kmph)
33 speed_iii_b = distance_total*3600/(t_1+t_2+t_3+
   t_iii_b)           // Scheduled speed with a stop
   of 15 sec(kmph)

```

34

```

35 // Results
36 disp("PART IV - EXAMPLE : 5.9 : SOLUTION :-")
37 printf("\nCase(i) : Acceleration,   = %.f km phps"
       , alpha)
38 printf("\nCase(ii) : Coasting retardation, -c = %.
        .2 f km phps", beta_c)
39 printf("\nCase(iii): Scheduled speed with a stop of
           20 seconds = %.2 f kmph", speed_iii_a)
40 printf("\n           Scheduled speed with a stop of
           15 seconds = %.2 f kmph\n", speed_iii_b)
41 printf("\nNOTE: ERROR: Calculation mistakes in the
           textbook solution")

```

---

### Scilab code Exa 43.10 Minimum adhesive weight of the locomotive

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 5: ELECTRIC TRACTION-SPEED TIME CURVES
     AND MECHANICS OF TRAIN MOVEMENT
8
9 // EXAMPLE : 5.10 :
10 // Page number 784
11 clear ; clc ; close ; // Clear the work space and
      console
12
13 // Given data
14 W = 350.0      // Weight of train(tonne)
15 G = 1.0        // Gradient
16 alpha = 0.8    // Acceleration(km phps)
17 u = 0.25       // Co-efficient of adhesion
18 r = 44.5       // Train resistance(N/tonne)

```

```

19 I = 10.0           // Rotational inertia(%)
20
21 // Calculations
22 W_e = W*(100+I)/100          // Accelerating
   weight of train(tonne)
23 F_t = 277.8*W_e*alpha+W*r+98.1*W*G // Tractive
   effort(N)
24 adhesive_weight = F_t/(u*9.81*1000) // Adhesive
   weight(tonnes)
25
26 // Results
27 disp("PART IV – EXAMPLE : 5.10 : SOLUTION :-")
28 printf("\nMinimum adhesive weight of the locomotive
   = %.1f tonnes\n", adhesive_weight)
29 printf("\nNOTE: ERROR: Train resistance is 44.5 N
   per tonne & not 45 N per tonne as mentioned in
   textbook problem statement")

```

---

### Scilab code Exa 43.11 Energy usefully employed in attaining speed and Specific energy

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 5: ELECTRIC TRACTION–SPEED TIME CURVES
   AND MECHANICS OF TRAIN MOVEMENT
8
9 // EXAMPLE : 5.11 :
10 // Page number 784
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data

```

```

14 W = 400.0          // Weight of train(tonne)
15 G = 100.0/75       // Gradient
16 alpha = 1.6         // Acceleration(km phps)
17 r = 66.75           // Train resistance(N/tonne)
18 I = 10.0            // Rotational inertia(%)
19 V = 48.0             // Speed(kmph)
20 n = 0.7              // Overall efficiency of equipment
21
22 // Calculations
23 W_e = W*(100+I)/100           // Accelerating
   weight of train(tonne)
24 F_t = 277.8*W_e*alpha+W*r+98.1*W*G // Tractive
   effort(N)
25 t = V/alpha                  // Time(sec)
26 energy_a = F_t*V*t/(2*3600**2)    // Energy
   usefully employed(kWh)
27 G_r = 98.1*G+r                // Force(N)
28 work_tonne_km = G_r*1000        // Work done
   per tonne per km(Nw-m)
29 energy_b = work_tonne_km/(n*3600) // Energy
   consumption(Wh per tonne-km)
30
31 // Results
32 disp("PART IV - EXAMPLE : 5.11 : SOLUTION :-")
33 printf("\nCase(a): Energy usefully employed in
   attaining speed = %.2f kWh", energy_a)
34 printf("\nCase(b): Specific energy consumption at
   steady state speed = %.1f Wh per tonne-km",
   energy_b)

```

---

### Scilab code Exa 43.12 Minimum adhesive weight of a locomotive

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.

```

```

4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 5: ELECTRIC TRACTION—SPEED TIME CURVES
    AND MECHANICS OF TRAIN MOVEMENT
8
9 // EXAMPLE : 5.12 :
10 // Page number 784–785
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 W = 200.0          // Trailing weight (tonne)
15 G = 1.0            // Gradient (%)
16 alpha = 1.0         // Acceleration (km phps)
17 u = 0.2             // Co-efficient of adhesion
18 r = 50.0            // Train resistance (N/tonne)
19 I = 10.0            // Rotational inertia (%)
20
21 // Calculations
22 W_L = ((277.8*(100+I)/100*alpha)+98.1*G+r)*W/(u
    *9.81*1000 - ((277.8*(100+I)/100*alpha)+98.1*G+r))
    // Weight of locomotive (tonnes)
23
24 // Results
25 disp("PART IV – EXAMPLE : 5.12 : SOLUTION :-")
26 printf("\nMinimum adhesive weight of a locomotive ,
    W_L = %.1f tonnes\n", W_L)
27 printf("\nNOTE: ERROR: Calculation mistake in
    textbook solution in calculating W_L")

```

---

## Chapter 44

# MOTORS FOR ELECTRIC TRACTION

Scilab code Exa 44.1 Speed current of the motor

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 6: MOTORS FOR ELECTRIC TRACTION
8
9 // EXAMPLE : 6.1 :
10 // Page number 788
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 I_1 = 10.0    // Current(A)
15 T_1 = 54.0    // Torque(N-m)
16 I_2 = 20.0    // Current(A)
17 T_2 = 142.0   // Torque(N-m)
18 I_3 = 30.0    // Current(A)
```

```

19 T_3 = 250.0 // Torque (N-m)
20 I_4 = 40.0 // Current (A)
21 T_4 = 365.0 // Torque (N-m)
22 I_5 = 50.0 // Current (A)
23 T_5 = 480.0 // Torque (N-m)
24 I_6 = 60.0 // Current (A)
25 T_6 = 620.0 // Torque (N-m)
26 I_7 = 70.0 // Current (A)
27 T_7 = 810.0 // Torque (N-m)
28 E = 500.0 // Operating voltage (V)
29 R_a = 0.6 // Armature resistance (ohm)
30
31 // Calculations
32 N_1 = 9.55*(E-I_1*R_a)*I_1/T_1 // Speed (rpm)
33 N_2 = 9.55*(E-I_2*R_a)*I_2/T_2 // Speed (rpm)
34 N_3 = 9.55*(E-I_3*R_a)*I_3/T_3 // Speed (rpm)
35 N_4 = 9.55*(E-I_4*R_a)*I_4/T_4 // Speed (rpm)
36 N_5 = 9.55*(E-I_5*R_a)*I_5/T_5 // Speed (rpm)
37 N_6 = 9.55*(E-I_6*R_a)*I_6/T_6 // Speed (rpm)
38 N_7 = 9.55*(E-I_7*R_a)*I_7/T_7 // Speed (rpm)
39
40 // Results
41 disp("PART IV – EXAMPLE : 6.1 : SOLUTION :-")
42 printf("\nSpeed-current of the motor")
43 printf("\n-----")
44 printf("\n Current(A) : Speed(rpm) ")
45 printf("\n-----")
46 printf("\n %.f : %.f , I_1 , N_1)
47 printf("\n %.f : %.f , I_2 , N_2)
48 printf("\n %.f : %.f , I_3 , N_3)
49 printf("\n %.f : %.f , I_4 , N_4)
50 printf("\n %.f : %.f , I_5 , N_5)
51 printf("\n %.f : %.f , I_6 ,

```

```

        N_6)
52 printf("\n    %.f :           %.f ", I_7,
        N_7)
53 printf("\n-----\n")
54 printf("\nNOTE: ERROR: Calculation mistakes in the
        textbook solution")

```

---

### Scilab code Exa 44.2 Speed torque for motor

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 6: MOTORS FOR ELECTRIC TRACTION
8
9 // EXAMPLE : 6.2 :
10 // Page number 788–789
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 N_1 = 500.0      // Speed(rpm)
15 I_1 = 50.0       // Current(A)
16 E_1 = 220.0      // Armature voltage(V)
17 I_2 = 100.0      // Current(A)
18 E_2 = 350.0      // Armature voltage(V)
19 I_3 = 150.0      // Current(A)
20 E_3 = 440.0      // Armature voltage(V)
21 I_4 = 200.0      // Current(A)
22 E_4 = 500.0      // Armature voltage(V)
23 I_5 = 250.0      // Current(A)
24 E_5 = 540.0      // Armature voltage(V)

```

```

25 I_6 = 300.0      // Current (A)
26 E_6 = 570.0      // Armature voltage (V)
27 R_wb = 0.08       // Armature and brush resistance (ohm)
28 R_f = 0.05        // Resistance of series field (ohm)
29 V = 600.0         // Operating voltage (V)
30
31 // Calculations
32 R_a = R_wb+R_f           // Armature resistance (
   ohm)
33 N_11 = N_1/E_1*(V-I_1*R_a) // Speed (rpm)
34 T_1 = 9.55*E_1*I_1/N_1     // Torque (N-m)
35 N_2 = N_1/E_2*(V-I_2*R_a) // Speed (rpm)
36 T_2 = 9.55*E_2*I_2/N_1     // Torque (N-m)
37 N_3 = N_1/E_3*(V-I_3*R_a) // Speed (rpm)
38 T_3 = 9.55*E_3*I_3/N_1     // Torque (N-m)
39 N_4 = N_1/E_4*(V-I_4*R_a) // Speed (rpm)
40 T_4 = 9.55*E_4*I_4/N_1     // Torque (N-m)
41 N_5 = N_1/E_5*(V-I_5*R_a) // Speed (rpm)
42 T_5 = 9.55*E_5*I_5/N_1     // Torque (N-m)
43 N_6 = N_1/E_6*(V-I_6*R_a) // Speed (rpm)
44 T_6 = 9.55*E_6*I_6/N_1     // Torque (N-m)
45
46 // Results
47 disp("PART IV - EXAMPLE : 6.2 : SOLUTION :-")
48 printf("\nSpeed-torque curve for motor")
49 printf("\n-----")
50 printf("\n Speed (rpm) : Torque (N-m) ")
51 printf("\n-----")
52 printf("\n %.f : %.f ", N_11,
   T_1)
53 printf("\n %.f : %.f ", N_2,
   T_2)
54 printf("\n %.f : %.f ", N_3,
   T_3)
55 printf("\n %.f : %.f ", N_4,
   T_4)
56 printf("\n %.f : %.f ", N_5,
   T_5)

```

```

57 printf("\n    %.f : %.f ", N_6,
      T_6)
58 printf("\n-----\n")
59 printf("\nNOTE: ERROR: Calculation mistakes in the
      textbook solution")

```

---

### Scilab code Exa 44.3 Speed of motors when connected in series

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 6: MOTORS FOR ELECTRIC TRACTION
8
9 // EXAMPLE : 6.3 :
10 // Page number 790
11 clear ; clc ; close ; // Clear the work space and
      console
12
13 // Given data
14 V = 650.0      // Voltage supply (V)
15 r_A = 45.0      // Radius of driving wheel (cm)
16 r_B = 43.0      // Radius of driving wheel (cm)
17 N_A = 400.0     // Speed (rpm)
18 drop = 10.0     // Voltage drop (%)
19
20 // Calculations
21 rho = r_B/r_A
22 IR = drop*V/100           // Voltage drop (V)
23 V_A = (rho*(V-IR)+IR)/(1+rho) // Voltage (V)
24 V_B = V-V_A               // Voltage (V)
25 N_A_A = N_A*(V_A-IR)/(V-IR) // N'_A (rpm)

```

```

26 N_B_B = N_A_A*r_A/r_B           // N_B (rpm)
27
28 // Results
29 disp("PART IV - EXAMPLE : 6.3 : SOLUTION :-")
30 printf("\nSpeed of first motor when connected in
series , N_A = %.f rpm", N_A_A)
31 printf("\nSpeed of second motor when connected in
series , N_B = %.f rpm\n", N_B_B)
32 printf("\nNOTE: Changes in the obtained answer from
that of textbook is due to more precision here")

```

---

### Scilab code Exa 44.4 HP delivered by the locomotive when dc series motor and Induc

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 6: MOTORS FOR ELECTRIC TRACTION
8
9 // EXAMPLE : 6.4 :
10 // Page number 791
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 F_t = 33800.0    // Tractive effort (N)
15 V = 48.3         // Velocity (kmph)
16 T = 53400.0      // Tractive effort (N)
17
18 // Calculations
19 HP = F_t*V*1000/(60*60*746)      // HP on level track(
hp)
20 HP_i = HP*(T/F_t)**0.5            // hp delivered by

```

```

    locomotive for dc series motor(hp)
21 HP_ii = HP*T/F_t           // hp delivered by
    locomotive for induction motor(hp)
22
23 // Results
24 disp("PART IV - EXAMPLE : 6.4 : SOLUTION :-")
25 printf("\nhp delivered by the locomotive when dc
    series motor is used = %.f HP", HP_i)
26 printf("\nhp delivered by the locomotive when
    induction motor is used = %.f HP", HP_ii)

```

---

### Scilab code Exa 44.5 New characteristics of motor

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 6: MOTORS FOR ELECTRIC TRACTION
8
9 // EXAMPLE : 6.5 :
10 // Page number 792-793
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 I_1 = 100.0           // Current (A)
15 N_1 = 71.0            // Speed(kmph)
16 F_t1 = 2225.0         // Tractive effort (N)
17 I_2 = 150.0            // Current (A)
18 N_2 = 57.0             // Speed(kmph)
19 F_t2 = 6675.0          // Tractive effort (N)
20 I_3 = 200.0            // Current (A)
21 N_3 = 50.0             // Speed(kmph)

```

```

22 F_t3 = 11600.0          // Tractive effort (N)
23 I_4 = 250.0             // Current (A)
24 N_4 = 45.0              // Speed (kmph)
25 F_t4 = 17350.0          // Tractive effort (N)
26 I_5 = 300.0             // Current (A)
27 N_5 = 42.0              // Speed (kmph)
28 F_t5 = 23200.0          // Tractive effort (N)
29 D_A = 101.6             // Size of wheels (cm)
30 ratio_gear = 72.0/23    // Gear ratio
31 D_B = 106.7             // Size of wheels (cm)
32 ratio_gear_new = 75.0/20 // Gear ratio
33
34 // Calculations
35 N_B = ratio_gear*D_B/(ratio_gear_new*D_A)      //
   Speed in terms of V(kmph)
36 F_tB = D_A*ratio_gear_new/(ratio_gear*D_B)      //
   Tractive effort in terms of F_tA(N)
37 N_B1 = N_B*N_1                         //
   Speed(kmph)
38 F_tB1 = F_tB*F_t1                      //
   Tractive effort (N)
39 N_B2 = N_B*N_2                         //
   Speed(kmph)
40 F_tB2 = F_tB*F_t2                      //
   Tractive effort (N)
41 N_B3 = N_B*N_3                         //
   Speed(kmph)
42 F_tB3 = F_tB*F_t3                      //
   Tractive effort (N)
43 N_B4 = N_B*N_4                         //
   Speed(kmph)
44 F_tB4 = F_tB*F_t4                      //
   Tractive effort (N)
45 N_B5 = N_B*N_5                         //
   Speed(kmph)
46 F_tB5 = F_tB*F_t5                      //
   Tractive effort (N)
47

```

```

48 // Results
49 disp("PART IV – EXAMPLE : 6.5 : SOLUTION :-")
50 printf("\nNew characteristics of motor")
51 printf("\n-----")
52 printf("\n Current(A) : Speed(kmph) : F_t(N)")
53 printf("\n-----")
54 printf("\n %.f : %.1f : %.f",
      I_1,N_B1,F_tb1)
55 printf("\n %.f : %.1f : %.f",
      I_2,N_B2,F_tb2)
56 printf("\n %.f : %.1f : %.f",
      I_3,N_B3,F_tb3)
57 printf("\n %.f : %.1f : %.f",
      I_4,N_B4,F_tb4)
58 printf("\n %.f : %.1f : %.f",
      I_5,N_B5,F_tb5)
59 printf("\n-----\n")
60 printf("\nNOTE: Changes in the obtained answer from
      that of textbook is due to more precision here")

```

---

# Chapter 45

## CONTROL OF MOTORS

Scilab code Exa 45.1 Approximate loss of energy in starting rheostats

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 7: CONTROL OF MOTORS
8
9 // EXAMPLE : 7.1 :
10 // Page number 798
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 no = 2.0          // Number of motors
15 V_m = 48.0        // Uniform speed (kmph)
16 t = 30.0          // Time(sec)
17 F_t_m = 13350.0  // Average tractive effort per
    motor(N)
18
19 // Calculations
```

```

20 F_t = no*F_t_m           // Average tractive
   effort (N)
21 energy = t*F_t*V_m/(2*3600**2) // Useful energy for
   acceleration (kWh)
22 energy_loss = energy/no      // Approximate loss
   of energy in starting rheostats (kWh)
23
24 // Results
25 disp("PART IV - EXAMPLE : 7.1 : SOLUTION :-")
26 printf("\nApproximate loss of energy in starting
   rheostats = %.3f kWh", energy_loss)

```

---

**Scilab code Exa 45.2 Energy supplied during the starting period Energy lost in the**

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 7: CONTROL OF MOTORS
8
9 // EXAMPLE : 7.2 :
10 // Page number 798
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 W = 175.0           // Weight of multiple unit train (
   tonnes)
15 no = 6.0            // Number of motors
16 F_t = 69000.0       // Total tractive effort (N)
17 V = 600.0           // Line voltage (V)
18 I = 200.0           // Average current (A)
19 V_m = 38.6          // Speed (kmph)

```

```

20 R = 0.15           // Resistance of each motor(ohm)
21
22 // Calculations
23 alpha = F_t/(277.8*W)          //
   Acceleration(km phps)
24 T = V_m/alpha                //
   Time for acceleration(sec)
25 t_s = (V-2*I*R)*T/(2*(V-I*R))    //
   Duration of starting period(sec)
26 t_p = T-t_s                  //
   (sec)
27 energy_total_series = no/2*V*I*t_s    //
   Total energy supplied in series position(watt-
   sec)
28 energy_total_parallel = no*V*I*t_p      //
   Total energy supplied in parallel position(watt-
   sec)
29 total_energy = (energy_total_series+
   energy_total_parallel)/(1000*3600)        //
   Energy supplied during starting period(kWh)
30 energy_waste_series = (no/2)/2*(V-2*I*R)*I*t_s    //
   Energy wasted in starting resistance in series
   position(watt-sec)
31 energy_waste_parallel = no*(V/2)/2*I*t_p      //
   Energy wasted in starting resistance in parallel
   position(watt-sec)
32 total_energy_waste = (energy_waste_series+
   energy_waste_parallel)/(1000*3600) // Total
   energy wasted in starting resistance(kWh)
33 energy_lost = (no*I**2*R*T)/(1000*3600)      //
   Energy lost in motor resistance(kWh)
34 useful_energy = T*F_t*V_m/(2*3600**2)        //
   Useful energy supplied to train(kWh)
35
36 // Results
37 disp("PART IV - EXAMPLE : 7.2 : SOLUTION :-")
38 printf("\nEnergy supplied during the starting period
   = %.2f kWh", total_energy)

```

```

39 printf("\nEnergy lost in the starting resistance = %f kWh", total_energy_waste)
40 printf("\nUseful energy supplied to the train = %.1f kWh", useful_energy)

```

---

### Scilab code Exa 45.3 Duration of starting period Speed of train at transition Rheo

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 7: CONTROL OF MOTORS
8
9 // EXAMPLE : 7.3 :
10 // Page number 799
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 W = 132.0           // Weight of electric train(
    tonnes)
15 no = 4.0            // Number of motors
16 V = 600.0           // Voltage of motor(V)
17 I = 400.0           // Current per motor(A)
18 F_t_m = 19270.0    // Tractive effort per motor at
    400A & 600V(N)
19 V_m = 39.0          // Train speed (kmph)
20 G = 1.0             // Gradient
21 r = 44.5            // Resistance to traction (N/tonne
    )
22 inertia = 10.0      // Rotational inertia(%)
23 R = 0.1              // Resistance of each motor(ohm)
24

```

```

25 // Calculations
26 W_e = W*(100+inertia)/100
                                // Accelerating
                                weight of train(tonne)
27 F_t = F_t_m*no
                                // Total
                                tractive effort at 400A & 600V(N)
28 alpha = (F_t-W*r-98.1*W*G)/(277.8*W_e)
                                // Acceleration(km phps)
29 T = V_m/alpha
                                // Time
                                for acceleration(sec)
30 t_s = (V-2*I*R)*T/(2*(V-I*R))
                                // Duration of starting
                                period(sec)
31 V_transition = alpha*t_s
                                // Speed at
                                transition(km phps)
32 t_p = T-t_s
                                // (
                                sec)
33 loss_series = (no/2*((V-2*I*R)/2)*I*t_s)/(1000*3600)
                                // Energy lost during series period(kWh)
34 loss_parallel = (no*(V/2)/2*I*t_p)/(1000*3600)
                                // Energy lost during parallel period(kWh)
)
35
36 // Results
37 disp("PART IV – EXAMPLE : 7.3 : SOLUTION :-")
38 printf("\nCase(i) : Duration of starting period ,
        t_s = %.1f sec", t_s)
39 printf("\nCase(ii) : Speed of train at transition ,
        t = %.1f sec", V_transition)
40 printf("\nCase(iii): Case(a): Rheostatic losses
        during series starting = %.2f kWh", loss_series)
41 printf("\n                 Case(b): Rheostatic losses
        during parallel starting = %.2f kWh\n",
        loss_parallel)

```

42 **printf**(”\nNOTE: ERROR: Calculation mistakes in the  
textbook solution”)

---

# Chapter 46

## BRAKING

Scilab code Exa 46.1 Braking torque

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 8: BRAKING
8
9 // EXAMPLE : 8.1 :
10 // Page number 806
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 525.0      // Voltage of motor(V)
15 I_1 = 50.0     // Current(A)
16 T_1 = 216.0    // Torque(N-m)
17 I_2 = 70.0     // Current(A)
18 T_2 = 344.0    // Torque(N-m)
19 I_3 = 80.0     // Current(A)
20 T_3 = 422.0    // Torque(N-m)
```

```

21 I_4 = 90.0 // Current (A)
22 T_4 = 500.0 // Torque (N-m)
23 V_m = 26.0 // Speed (kmph)
24 R_b = 5.5 // Resistance of braking rheostat (ohm)
25 R_m = 0.5 // Resistance of motor (ohm)
26
27 // Calculations
28 I = 75.0 // Current drawn at 26 kmph(A)
29 back_emf = V-I*R_m // Back emf of the motor (V)
30 R_t = R_b+R_m // Total resistance (ohm)
31 I_del = back_emf/R_t // Current delivered (A)
32 T_b = T_3*I_del/I_3 // Braking torque (N-m)
33
34 // Results
35 disp("PART IV - EXAMPLE : 8.1 : SOLUTION :-")
36 printf("\nBraking torque = %.f N-m", T_b)

```

---

### Scilab code Exa 46.2 Current delivered when motor works as generator

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 8: BRAKING
8
9 // EXAMPLE : 8.2 :
10 // Page number 806
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 V = 525.0 // Voltage of motor (V)

```

```

15 I_1 = 50.0      // Current (A)
16 N_1 = 1200.0    // Speed (rpm)
17 I_2 = 100.0     // Current (A)
18 N_2 = 950.0     // Speed (rpm)
19 I_3 = 150.0     // Current (A)
20 N_3 = 840.0     // Speed (rpm)
21 I_4 = 200.0     // Current (A)
22 N_4 = 745.0     // Speed (rpm)
23 N = 1000.0      // Speed operating (rpm)
24 R = 3.0          // Resistance (ohm)
25 R_m = 0.5        // Resistance of motor (ohm)
26
27 // Calculations
28 I = 85.0          // Current drawn at 1000 rpm
(A)
29 back_emf = V-I*R_m // Back emf of the motor (V)
30 R_t = R+R_m        // Total resistance (ohm)
31 I_del = back_emf/R_t // Current delivered (A)
32
33 // Results
34 disp("PART IV - EXAMPLE : 8.2 : SOLUTION :-")
35 printf("\nCurrent delivered when motor works as
generator = %.f A", I_del)

```

---

### Scilab code Exa 46.3 Energy returned to lines

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 8: BRAKING
8
9 // EXAMPLE : 8.3 :

```

```

10 // Page number 810
11 clear ; clc ; close ; // Clear the work space and
   console
12
13 // Given data
14 W = 400.0          // Weight of train(tonne)
15 G = 100.0/70       // Gradient(%)
16 t = 120.0          // Time(sec)
17 V_1 = 80.0          // Speed(km/hr)
18 V_2 = 50.0          // Speed(km/hr)
19 r_kg = 5.0          // Tractive resistance(kg/tonne)
20 I = 7.5             // Rotational inertia(%)
21 n = 0.75            // Overall efficiency
22
23 // Calculations
24 W_e = W*(100+I)/100
                           //
                           Accelerating weight of train(tonne)
25 r = r_kg*9.81
                           //
                           Tractive resistance(N-m/tonne)
26 energy_recuperation = 0.01072*W_e*(V_1**2-V_2**2)
   /1000 // Energy available for recuperation(kWh)
27 F_t = W*(r-98.1*G)
                           //
                           Tractive
                           effort during retardation(N)
28 distance = (V_1+V_2)*1000*t/(2*3600)
                           //
                           Distance travelled by
                           train during retardation period(m)
29 energy_train = abs(F_t)*distance/(3600*1000)
                           //
                           Energy available during train
                           movement(kWh)
30 net_energy = n*(energy_recuperation+energy_train)
                           //
                           Net energy returned to supply system(
                           kWh)
31
32 // Results
33 disp("PART IV – EXAMPLE : 8.3 : SOLUTION :-")

```

```

34 printf("\nEnergy returned to lines = %.2f kWh\n",
35     net_energy)
35 printf("\nNOTE: ERROR: Calculation mistakes & more
approximation in textbook solution")

```

---

### Scilab code Exa 46.4 Energy returned to the line

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 8: BRAKING
8
9 // EXAMPLE : 8.4 :
10 // Page number 810
11 clear ; clc ; close ; // Clear the work space and
console
12
13 // Given data
14 W = 355.0           // Weight of train(tonne)
15 V_1 = 80.5          // Speed(km/hr)
16 V_2 = 48.3          // Speed(km/hr)
17 D = 1.525           // Distance(km)
18 G = 100.0/90        // Gradient(%)
19 I = 10.0            // Rotational inertia(%)
20 r = 53.0             // Tractive resistance(N/tonne)
21 n = 0.8              // Overall efficiency
22
23 // Calculations
24 beta = (V_1**2-V_2**2)/(2*D*3600)    // Braking
retardation(km phps)
25 W_e = W*(100+I)/100                  // Accelerating
weight of train(tonne)

```

```

26 F_t = 277.8*W_e*beta+98.1*W*G-W*r // Tractive
    effort (N)
27 work_done = F_t*D*1000 // Work done by
    this effort (N-m)
28 energy = work_done*n/(1000*3600) // Energy
    returned to line (kWh)
29
30 // Results
31 disp("PART IV - EXAMPLE : 8.4 : SOLUTION :-")
32 printf("\nEnergy returned to the line = %.1f kWh",
    energy)

```

---

### Scilab code Exa 46.5 Braking effect and Rate of retardation produced by this braking

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 8: BRAKING
8
9 // EXAMPLE : 8.5 :
10 // Page number 811–812
11 clear ; clc ; close ; // Clear the work space and
    console
12 funcprot(0)
13
14 // Given data
15 area = 16.13 // Area of brakes (sq.cm/pole face
    )
16 phi = 2.5*10**-3 // Flux (Wb)
17 u = 0.2 // Co-efficient of friction
18 W = 10.0 // Weight of car (tonnes)
19

```

```

20 // Calculations
21 a = area*10**-4                                // Area of brakes(
      sq.m/pole face)
22 F = phi**2/(2*pi*10**-7*a)                  // Force(N)
23 force = F*u                                    // Braking effect
      considering flux and coefficient of friction(N)
24 beta = u*F/(W*1000)*100                      // Rate of
      retardation produced by braking effect(cm/sec^2)
25
26 // Results
27 disp("PART IV - EXAMPLE : 8.5 : SOLUTION :-")
28 printf("\nBraking effect , F = %.f N", force)
29 printf("\nRate of retardation produced by this
      braking effect ,      = %.2 f cm/sec^2", beta)

```

---

## Chapter 47

# ELECTRIC TRACTION SYSTEMS AND POWER SUPPLY

Scilab code Exa 47.1 Maximum potential difference between any two points of the rail

```
1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 9: ELECTRIC TRACTION SYSTEMS AND POWER
    SUPPLY
8
9 // EXAMPLE : 9.1 :
10 // Page number 817–818
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 L = 3.0           // Length of section ACB of rail (
    km)
```

```

15 L_B_A = 2.0          // Distance of B from A(km)
16 I_load = 350.0       // Loading(A/km)
17 r_rail = 0.035        // Resistance of rail(ohm/km)
18 r_feed = 0.03         // Resistance of negative feeder(
                           ohm/km)
19
20 // Calculations
21 x_val = integrate('I_load*(L-x)', 'x', 0, L_B_A)           //
22 I = x_val/(L_B_A-0)                                         // Current in negative feeder(A)
23 x = L-(I/I_load)                                           //
24 C = integrate('r_rail*I_load*x', 'x', 0, x)                //
25 V = r_feed*L_B_A*I                                         // Voltage produced by negative booster(V)
26 rating = V*I/1000                                           //
27                                                 // Rating of the booster(kW)
28 // Results
29 disp("PART IV – EXAMPLE : 9.1 : SOLUTION :-")
30 printf("\nMaximum potential difference between any
           two points of the rails , C = %.2f V", C)
31 printf("\nRating of the booster = %.1f kW", rating)

```

---

### Scilab code Exa 47.2 Maximum sag and Length of wire required

```

1 // A Texbook on POWER SYSTEM ENGINEERING
2 // A. Chakrabarti , M.L. Soni , P.V. Gupta , U.S. Bhatnagar
3 // DHANPAT RAI & Co.
4 // SECOND EDITION
5
6 // PART IV : UTILIZATION AND TRACTION
7 // CHAPTER 9: ELECTRIC TRACTION SYSTEMS AND POWER
   SUPPLY
8

```

```

9 // EXAMPLE : 9.2 :
10 // Page number 820
11 clear ; clc ; close ; // Clear the work space and
    console
12
13 // Given data
14 D = 50.0      // Distance between poles (m)
15 w = 0.5       // Weight of trolley wire per metre (kg)
16 T = 520.0     // Maximum tension (kg)
17
18 // Calculations
19 l = D/2          // Half
    distance b/w poles (m)
20 d = w*l**2/(2*T)           // Sag (m)
21 wire_length = 2*(l+(2*d**2/(3*l))) // Length of
    wire required (m)
22
23 // Results
24 disp("PART IV – EXAMPLE : 9.2 : SOLUTION :-")
25 printf("\nMaximum sag , d = %.4f metres", d)
26 printf("\nLength of wire required = %.f metres",
    wire_length)

```

---