

Scilab Textbook Companion for
Numerical Methods: Principles, Analysis, And
Algorithms
by S. Pal¹

Created by
Saurav Suman
B.Tech
Others
NIT Jamshedpur
College Teacher
NA
Cross-Checked by

July 31, 2019

¹Funded 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: Numerical Methods: Principles, Analysis, And Algorithms

Author: S. Pal

Publisher: Oxford University Press

Edition: 1

Year: 2009

ISBN: 9780195693751

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
1 Background to Numerical Methods	5
2 Scope of Numerical and Mathematical Methods	36
3 Errors and Their Propagation	38
4 Programming Tools and Techniques	42
5 Solutions of Algebraic and Transcendental Equations	46
6 Numerical Methods of Linear Equations Direct Methods	87
7 Numerical Solutions for Matrix Inversion	98
8 Numerical Solutions of Linear Systems of Equations Iterative Methods	108
9 Linear Least Squares Problem	125
10 Numerical Solutions of System of Non Linear Equations	136
11 Eigenvalues and Eigenvectors	144

12 Interpolation and Extrapolation	163
13 Numerical Differentiation	175
14 Numerical Integration	184
15 Numerical Solutions of Ordinary Differential Equations Initial Value Problem	196
16 Numerical Solutions of Ordinary Differential Equations Boundary Value Problems	217
18 Numerical Solutions of Parabolic Partial Differential Equations	226
19 Numerical Solutions of Hyperbolic Partial Differential Equations	236
20 Numerical Solutions of Elliptical Partial Differential Equations	255
21 Advances in Numerical Methods Using Parallel Computing Paradigm	266
22 Numerical Methods Using Neural Networks	275

List of Scilab Codes

Exa 1.1	Conversion to Decimal System	5
Exa 1.2	Conversion Using Shortcut Method	6
Exa 1.3	Conversion to Base B from Decimal System	7
Exa 1.4	Conversion to Binary System	7
Exa 1.5	Conversion to Binary System	8
Exa 1.6	Conversion to Decimal Number	9
Exa 1.7	Conversion to Decimal Number	9
Exa 1.8	Conversion to Base B from Binary System	10
Exa 1.9	Conversion to Binary System	11
Exa 1.10	Conversion to Binary System and to Base N	11
Exa 1.13	1s compliment and 2s compliment	12
Exa 1.14	1s compliment	14
Exa 1.15	Addition and Subtraction	15
Exa 1.16	Addition	17
Exa 1.17	Addition	19
Exa 1.18	Addition	20
Exa 1.19	Addition	22
Exa 1.20	Subtraction	23
Exa 1.23	Multiplication	25
Exa 1.24	Multiplication	26
Exa 1.25	Division	27
Exa 1.26	Multiplication	29
Exa 1.29	Normalized Floating Point Representation	31
Exa 1.30	Add	32
Exa 1.31	Add	32
Exa 1.32	Add	33
Exa 1.33	Add	33
Exa 1.34	Subtraction	34

Exa 1.35	Multiplication	34
Exa 1.36	Division	35
Exa 2.4	Solving Simultaneous Linear Equation	36
Exa 2.6	Integration	37
Exa 3.1	Limiting Error	38
Exa 3.2	Known Error	38
Exa 3.3	Absolute Relative and Percentage Errors	39
Exa 3.4	Absolute Relative and Percentage Errors	39
Exa 3.5	Absolute Relative and Percentage Errors	40
Exa 4.1	Quadratic Equation	42
Exa 4.2	Database Management	42
Exa 5.1	Bisection Method	46
Exa 5.2	Bisection Method	47
Exa 5.3	Regula Falsi Method	48
Exa 5.4	Ridders Method	49
Exa 5.5	General Iterative Method	50
Exa 5.6	Linear Iterative Method	50
Exa 5.7	Aitkens Method	51
Exa 5.8	Newton Raphson Method	52
Exa 5.9	Modified Newton Raphson Method	53
Exa 5.10	Newton Raphson Method	53
Exa 5.11	Newton Raphson Method	54
Exa 5.12	Newton Raphson Method	55
Exa 5.13	Secant Method	56
Exa 5.14	Kizner Method	57
Exa 5.15	Brent Method	58
Exa 5.19	Horner Method	58
Exa 5.20	Laguerre Method	59
Exa 5.21	Mullers Method	60
Exa 5.22	Mullers Method	63
Exa 5.23	Bairstow Hitchcock Method	66
Exa 5.24	Bernoulli Method	68
Exa 5.25	Graeffe Method	68
Exa 5.26	QD Method	70
Exa 5.27	Linear Iteration Method	71
Exa 5.28	Aitkens Method	72
Exa 5.29	Newton Raphson Method	72
Exa 5.31	Secant Method	73

Exa 5.32	Regula Falsi Newton Raphson and Mullers Method	74
Exa 5.33	Newton Raphson and Mullers Method	78
Exa 5.34	QD Method	80
Exa 5.35	Newton Raphson Method	81
Exa 5.36	Secant Method	82
Exa 5.37	Newton Raphson Method	83
Exa 5.38	Newton Raphson Method	84
Exa 5.39	Newton Raphson Method	85
Exa 5.40	Newton Raphson Method	86
Exa 6.1	Gaussian Elimination Method	87
Exa 6.2	Gaussian Elimination Method for TriDiagonal System	88
Exa 6.3	Gauss Jordan Method	89
Exa 6.4	Gaussian Elimination Method without Pivoting	90
Exa 6.5	Dolittle Factorization Method	91
Exa 6.6	Trangularization Method	93
Exa 6.7	Wilkinson Method	94
Exa 6.8	Choleskys Factorization	95
Exa 6.9	Complex System of Linear Equation	96
Exa 6.10	Solving Matrices	96
Exa 7.1	Gauss Jordan Two Array Method	98
Exa 7.2	Inverse in Place without Pivoting	102
Exa 7.3	Inverse in Place with Pivoting	103
Exa 7.4	Inverse of Triangular Matrices	105
Exa 7.5	Inverse of Complex Matrices	106
Exa 7.6	Iterative Procedure	107
Exa 8.1	Jacobi Method	108
Exa 8.2	Gauss Seidel Method	109
Exa 8.3	SOR Method	110
Exa 8.4	Gauss Seidel Point Iterative Method	111
Exa 8.5	Gauss Seidel Point Iterative Method	113
Exa 8.6	Block Jacobi Method	116
Exa 8.7	Block Gauss Seidel Method	118
Exa 8.8	Block SOR Method	121
Exa 9.1	Moore Penrose Generalized Inverse	125
Exa 9.2	Curve Fitting	125

Exa 9.3	Gram Schmidt Orthogonalization or Orthonormalization Process	128
Exa 9.4	QR Decomposition	129
Exa 9.5	Vector Computation	131
Exa 9.6	House Holder Transformation	132
Exa 9.7	Givens QR Method	133
Exa 9.8	Recursive Least Square Method	134
Exa 10.1	System of Non Linear Equations	136
Exa 10.2	Contraction Method and Seidel Method	137
Exa 10.3	Non Linear System of Equation	137
Exa 10.4	Newton Method	138
Exa 10.5	Newton Raphshon Method	139
Exa 10.6	Newton Method	140
Exa 10.7	Iterative Method	142
Exa 10.8	Steepest Descent	142
Exa 11.1	Eigenvalues and Eigenvectors	144
Exa 11.2	Leverriers Method	145
Exa 11.3	Danilevsky Method	146
Exa 11.4	Power Method	149
Exa 11.5	Inverse Power Method	150
Exa 11.6	Rayleigh Quotient	151
Exa 11.7	Jacobi Method	152
Exa 11.8	Recursive Formula	154
Exa 11.9	QR Method	154
Exa 11.10	LU Method	157
Exa 11.11	Generalized Eigenvalue Problem	159
Exa 12.1	Linear Interpolation Technique	163
Exa 12.2	Lagarangian Method	164
Exa 12.3	Aitken Nevilles Method	165
Exa 12.4	Newtons Divided Difference Interpolation	166
Exa 12.5	Interpolation Methods	167
Exa 12.6	Chebyshev Interpolating Polynomial	170
Exa 12.7	Double Interpolation	172
Exa 12.8	Spline Interpolation	173
Exa 13.1	Differentiation	175
Exa 13.2	Calculation of x coordinate of Minimum Point	175
Exa 13.3	Newton Forward Difference Formula	177
Exa 13.4	Newton Backward Difference Formula	178

Exa 13.5	Stirlings Central Difference Derivatives . . .	180
Exa 13.6	Extrapolation	181
Exa 13.7	Richardson Extrapolation	181
Exa 13.8	Application	182
Exa 14.2	Simpsons 1 3rd Rule	184
Exa 14.3	Trapezoidal Rule and Simpsons Rule	185
Exa 14.5	Romberg Method	187
Exa 14.7	Gaussian Quadrature Formula	188
Exa 14.8	Gauss Legendre Two Point Rule	188
Exa 14.9	Gauss Legendre Three Point Rule	189
Exa 14.10	Spline Integration Method	189
Exa 14.11	Trapezoidal Rule	190
Exa 14.14	Trapezoidal and Simpsons Rule	191
Exa 14.15	Trapezoidal and Simpsons Rule	192
Exa 14.16	Multiple Integration with Variable Limits .	194
Exa 14.18	Integration	195
Exa 14.19	Integration	195
Exa 15.1	Ordinary Differential Equation	196
Exa 15.6	Taylor Method	196
Exa 15.7	Picard Method	197
Exa 15.8	Euler Method	198
Exa 15.9	Trapezium Method	198
Exa 15.10	Heun Method	199
Exa 15.11	Midpoint Method	200
Exa 15.12	Modified Midpoint Method	200
Exa 15.13	Single Step Method	201
Exa 15.14	Second Order Runge Kutta Method	202
Exa 15.15	Third Order Runge Kutta Method	203
Exa 15.16	Fourth Order Runge Kutta Method	203
Exa 15.17	New Variant of Runge Kutta Method	204
Exa 15.18	Runge Kutta Merson Method	205
Exa 15.19	Runge Kutta Fehlberg Method	205
Exa 15.20	Carp Karp Runge Kutta Method	206
Exa 15.21	Implicit Runge Kutta Method	207
Exa 15.22	Linear Multi Step Method	208
Exa 15.23	Milne Simpson Predictor Corrector Method	209
Exa 15.24	Improved Milne Simpson Predictor Corrector Method	209

Exa 15.25	Hamming Predictor Corrector Method	211
Exa 15.26	Multi Valued Method	212
Exa 15.27	First order ODE	214
Exa 15.28	Differential Equation	215
Exa 16.1	Outline of Linear Shooting Method	217
Exa 16.2	Linear Shooting Method	218
Exa 16.3	Multiple Shooting Method	220
Exa 16.4	Finite Difference Method	222
Exa 16.5	Non Linear Problem	223
Exa 16.6	Collocation Method	224
Exa 18.4	Forward Difference Method	226
Exa 18.5	Bender Schmidt Method	227
Exa 18.6	Crank Nicolson Method	228
Exa 18.7	Gauss Seidel Method	230
Exa 18.8	ADI Method	231
Exa 19.3	Simple Explicit Method	236
Exa 19.4	Simple Implicit Method	237
Exa 19.5	Lax Wendroff Method	238
Exa 19.6	Wendroff Method	239
Exa 19.7	Leapfrog Method	240
Exa 19.8	Variable Coefficients	242
Exa 19.9	Inhomogeneous 1st Order Hyperbolic Differential Equation	245
Exa 19.10	Non Linear 1st Order Hyperbolic Differential Equation	248
Exa 19.11	Finite Difference Method	249
Exa 19.12	Hyperbolic Partial Differential Equations . .	251
Exa 19.13	Hyperbolic Differential Equations in 2D or 3D	251
Exa 20.1	Direct Method	255
Exa 20.2	Five Point Formula	256
Exa 20.3	Finite Difference Method	257
Exa 20.4	Seven Point Formula	258
Exa 20.5	Nine Point Formula	259
Exa 20.6	Five Point Formula	261
Exa 20.7	Laplace Distribution	262
Exa 20.8	Spherical Coordinate System	263
Exa 21.1	Parallel Bisection Method	266
Exa 21.2	Lagrange Interpolation in Parallel Computing	268

Exa 21.3	Trapezoidal Rule and Simpsons Rule in Parallel Computing	269
Exa 21.4	Parallel Gauss Seidel Method	270
Exa 21.5	Poissons Partial Differential Equation	271
Exa 22.1	MLP Algorithm	275
Exa 22.2	MLP	276
Exa 22.3	Bisection Method	277
Exa 22.4	Hopfield Neural Network	278
Exa 22.5	RBF Network	279
Exa 22.7	First Order ODE	280

Chapter 1

Background to Numerical Methods

Scilab code Exa 1.1 Conversion to Decimal System

```
1 //Example 1.1
2 //Conversion to Decimal System
3 //Page no. 4
4 clc;close;clear;
5 function [s]=bas2dec(x,b)
6     xi=int(x)
7     xd=x-int(x)
8     s=0
9     for i=1:10
10        xi=xi/10
11        s=s+(10*(xi-fix(xi))*b^(i-1))
12        xi=int(xi)
13        if (xi==0)
14            break
15        end
16    end
17    for i=1:1
```

```

18     xd=xd*10;
19     s=s+(ceil(xd)/b^(i))
20     xd=xd-fix(xd)
21     if(xd==0)
22         break
23     end
24 end
25 endfunction
26
27 //conversion from hexadecimal to decimal system
28 disp(hex2dec('1A2C'), '1A2C=');           //inbuilt function
29
30 //conversion from hexadecimal to decimal system
31 disp(bas2dec(428.5,8), '428.5=');        //inline function
32
33 //conversion from hexadecimal to decimal system
34 disp(bas2dec(120.1,3), '120.1=');        //inline
function

```

Scilab code Exa 1.2 Conversion Using Shortcut Method

```

1 //Example 1.2
2 //Conversion Using Shortcut Method
3 //Page no. 4
4 clc;close;clear;
5 A=10;C=12;
6 d=((1)*16+A)*16+2)*16+C;
7 disp(d, 'Decimal form of 1A2C is =');

```

Scilab code Exa 1.3 Conversion to Base B from Decimal System

```
1 //Example 1.3
2 //Conversion to Base B from Decimal System
3 //Page no. 5
4 clc;close;clear;
5 //conversion from binary to octal
6 disp(dec2oct(bin2dec('10101101110')), 'Octal form of
    10101101110 is ='); //inbuilt function
7
8 //conversion from binary to hexadecimal
9 disp(dec2hex(bin2dec('10101101110')), 'Hexadecimal
    form of 10101101110 is ='); //inbuilt function
10
11 //conversion from binary to octal
12 s=dec2oct(bin2dec('1011'));
13 s1=dec2oct(bin2dec('110011010100'));//inbuilt
    function
14 printf('\n Octal form of 1011.1100110101 is = \n\n
    %s.%s',s,s1)
15
16 //conversion from binary to hexadecimal
17 s=dec2hex(bin2dec('1011'));
18 s1=dec2hex(bin2dec('110011010100'));//inbuilt
    function
19 printf('\n\n Hexadecimal form of 1011.1100110101 is
    = \n\n %s.%s',s,s1)
```

Scilab code Exa 1.4 Conversion to Binary System

```
1 //Example 1.4
2 //Conversion to Binary System
3 //Page no. 6
```

```

4 clc;close;clear;
5 //conversion from octal to binary
6 disp(dec2bin(oct2dec('1753')), 'Binary form of 1753
    is =');      //inbuilt function
7
8 //conversion from octal to binary
9 disp(dec2bin(hex2dec('A478')), 'Binary form of A478
    is =');      //inbuilt function
10
11 //conversion from octal to binary
12 s=dec2bin(oct2dec('3'));
13 s1=dec2bin(oct2dec('154'));      //inbuilt function
14 printf('\n Octal form of 3.154 is = \n\n %s.00%s',s,
    s1)

```

Scilab code Exa 1.5 Conversion to Binary System

```

1 //Example 1.5
2 //Conversion to Binary System
3 //Page no. 6
4 clc;close;clear;
5 //conversion from octal to binary
6 b=dec2bin(oct2dec('1753'))
7 disp(b, 'Binary form of 1753 is =');      //inbuilt
    function
8 b=dec2hex(oct2dec('1753'))
9 disp(b, 'Hexadecimal form of 1753 is =');      //
    inbuilt function
10 //conversion from octal to binary
11 b=dec2bin(hex2dec('A478'))
12 disp(b, 'Binary form of A478 is =');      //inbuilt
    function
13 b=dec2oct(hex2dec('A478'))

```

```
14 disp(b,'Octal form of A478 is ='); //inbuilt  
    function  
15 //conversion from octal to binary  
16 s=dec2bin(oct2dec('3'));  
17 s1=dec2bin(oct2dec('154'));//inbuilt function  
18 printf('\n Octal form of 3.154 is = \n\n %s.00%s',s,  
    s1)  
19 s=dec2hex(oct2dec('3'));  
20 s1=dec2hex(oct2dec('154'));//inbuilt function  
21 printf('\n\n Hexadecimal form of 3.154 is = \n\n %s.  
    %s',s,s1)
```

Scilab code Exa 1.6 Conversion to Decimal Number

```
1 //Example 1.6  
2 //Conversion to Decimal Number  
3 //Page no. 7  
4 clc;close;clear;  
5  
6 disp(dec2bin(182),'Binary of 182='); //inbuilt  
    function
```

Scilab code Exa 1.7 Conversion to Decimal Number

```
1 //Example 1.7  
2 //Conversion to Decimal Number  
3 //Page no. 7  
4 clc;close;clear;  
5
```

```
6 disp(dec2oct(467), 'Octal of 467=') //  
    inbuilt function
```

Scilab code Exa 1.8 Conversion to Base B from Binary System

```
1 //Example 1.8  
2 //Conversion to Base B from Binary System  
3 //Page no. 8  
4 clc; close; clear;  
5 //conversion from binary to octal  
6 disp(dec2oct(bin2dec('10101101110')), 'Octal form of  
    10101101110 is ='); //inbuilt function  
7  
8 //conversion from binary to hexadecimal  
9 disp(dec2hex(bin2dec('10101101110')), 'Hexadecimal  
    form of 10101101110 is ='); //inbuilt function  
10  
11 //conversion from binary to octal  
12 s=dec2oct(bin2dec('1011'));  
13 s1=dec2oct(bin2dec('110011010100')); //inbuilt  
    function  
14 printf('\n Octal form of 1011.1100110101 is = \n\n  
    %s.%s', s, s1)  
15  
16 //conversion from binary to hexadecimal  
17 s=dec2hex(bin2dec('1011'));  
18 s1=dec2hex(bin2dec('110011010100')); //inbuilt  
    function  
19 printf('\n\n Hexadecimal form of 1011.1100110101 is  
    = \n\n %s.%s', s, s1)
```

Scilab code Exa 1.9 Conversion to Binary System

```
1 //Example 1.9
2 //Conversion to Binary System
3 //Page no. 8
4 clc;close;clear;
5 //conversion from octal to binary
6 disp(dec2bin(oct2dec('1753')), 'Binary form of 1753
    is =');      //inbuilt function
7
8 //conversion from octal to binary
9 disp(dec2bin(hex2dec('A478')), 'Binary form of A478
    is =');      //inbuilt function
10
11 //conversion from octal to binary
12 s=dec2bin(oct2dec('3'));
13 s1=dec2bin(oct2dec('154'));//inbuilt function
14 printf('\n Octal form of 3.154 is = \n\n %s.00%s',s,
    s1)
```

Scilab code Exa 1.10 Conversion to Binary System and to Base N

```
1 //Example 1.10
2 //Conversion to Binary System and to Base N
3 //Page no. 9
4 clc;close;clear;
5
6 b=dec2bin(oct2dec('1753'))
```

```

7 disp(b,'Binary form of 1753 is ='); //inbuilt
    function
8 b=dec2hex(oct2dec('1753'))
9 disp(b,'Hexadecimal form of 1753 is ='); //inbuilt
    function
10 //conversion from octal to binary
11 b=dec2bin(hex2dec('A478'))
12 disp(b,'Binary form of A478 is ='); //inbuilt
    function
13 b=dec2oct(hex2dec('A478'))
14 disp(b,'Octal form of A478 is ='); //inbuilt
    function
15 //conversion from octal to binary
16 s=dec2bin(oct2dec('3'));
17 s1=dec2bin(oct2dec('154'));//inbuilt function
18 printf('\n Octal form of 3.154 is = \n\n %s.00%s',s,
    s1)
19 s=dec2hex(oct2dec('3'));
20 s1=dec2hex(oct2dec('154'));//inbuilt function
21 printf('\n\n Hexadecimal form of 3.154 is = \n\n %s.
    %s',s,s1)

```

Scilab code Exa 1.13 1s compliment and 2s compliment

```

1 //Example 1.13
2 //1s compliment and 2s compliment
3 //Page no. 11
4 clc;close;clear;
5
6 function [x1]=com1(x) //function for 1s
    compliment
7 for i=8:-1:1
8     x=x/10;

```

```

9         xd=x-fix(x)
10        if(floor((xd*10)+0.1)==1)
11            x1(1,i)=0;
12        else
13            x1(1,i)=1;
14        end
15        x=x-xd;
16    end
17 endfunction
18 function [x1]=com2(x)           // function for 2s
19     compliment()
20     for i=8:-1:1
21         x=x/10;
22         xd=x-fix(x)
23         if(int((xd*10)+0.1)==1)
24             x1(1,i)=0;
25         else
26             x1(1,i)=1;
27         end
28     end
29     for i=8:-1:1
30         if (x1(1,i)==0) then
31             x1(1,i)=1;
32             break;
33         else
34             x1(1,i)=0;
35         end
36     end
37 endfunction
38 a
=[00010011,01110110,11101101,10000001,10000000,00000000];
39 for i=1:6
40     printf('1s Compliment of %.8i=' ,a(i));
41     disp(com1(a(i)))
42     printf('2s Compliment of %.8i=' ,a(i));
43     disp(com2(a(i)))

```

```
44     printf ('\n\n')
45 end
```

Scilab code Exa 1.14 1s compliment

```
1 //Example 1.14
2 //1s compliment
3 //Page no. 12
4 clc;close;clear;
5
6 function [x1]=com1(x)          //function for 1s
    compliment
7     for i=8:-1:1
8         x=x/10;
9         xd=x-fix(x)
10        if(floor((xd*10)+0.1)==1)
11            x1(1,i)=0;
12        else
13            x1(1,i)=1;
14        end
15        x=x-xd;
16    end
17 endfunction
18 a
=[00010011,01110110,11101101,10000001,10000000,00000000];
19 for i=1:6
20     printf('1s Compliment of %.8i=',a(i));
21     disp(com1(a(i)))
22     printf('\n\n')
23 end
```

Scilab code Exa 1.15 Addition and Subtraction

```
1 //Example 1.15
2 //Addition and Subtraction
3 //Page no. 13
4 clc;clear;close;
5 function [x1]=add(x,y)                                //function
    for addition of binaries
6     c=0;
7     for i=1:10
8         x1(1,i)=0
9     end
10    for i=10:-1:1
11        x=x/10;
12        xd=x-fix(x)
13        x=x-xd;
14        y=y/10;
15        yd=y-fix(y)
16        y=y-yd;
17        if c==1 then
18            if floor((xd*10)+0.1)==1 & floor((yd*10)
+0.1)==1 then
19                x1(1,i)=1;c=1;
20            elseif floor((xd*10)+0.1)==0 & floor((yd
*10)+0.1)==0
21                x1(1,i)=1;c=0;
22            else
23                x1(1,i)=0;c=0;
24            end
25        else
26            if floor((xd*10)+0.1)==1 & floor((yd*10)
+0.1)==1 then
```

```

27           x1(1,i)=0;c=1;
28       elseif floor((xd*10)+0.1)==0 & floor((yd
29           *10)+0.1)==0
30           x1(1,i)=0;c=0;
31       else
32           x1(1,i)=1;c=0;
33       end
34   end
35   disp(x1,'Addition of 173 and 141=')
36 endfunction
37 function [x1]=sub(x,y) // function for
38     subtraction of binaries
39     c=0;
40     for i=1:10
41         x1(1,i)=0
42     end
43     for i=10:-1:1
44         x=x/10;
45         xd=x-fix(x)
46         x=x-xd;
47         y=y/10;
48         yd=y-fix(y)
49         y=y-yd;
50         if c==1 then
51             if floor((xd*10)+0.1)==0 & floor((yd
52                 *10)+0.1)==1 then
53                 x1(1,i)=0;c=1;
54             elseif floor((xd*10)+0.1)==0 & floor
55                 ((yd*10)+0.1)==0
56                 x1(1,i)=1;c=0;
57             elseif floor((xd*10)+0.1)==1 & floor
58                 ((yd*10)+0.1)==1
59                 x1(1,i)=1;c=1;
60             elseif floor((xd*10)+0.1)==1 & floor
61                 ((yd*10)+0.1)==0
62                 x1(1,i)=0;c=0;
63         end

```

```

59         else
60             if floor((xd*10)+0.1)==1 & floor((yd
61                 *10)+0.1)==1 then
62                 x1(1,i)=1;c=1;
63             elseif floor((xd*10)+0.1)==0 & floor
64                 ((yd*10)+0.1)==0
65                 x1(1,i)=0;c=0;
66             elseif floor((xd*10)+0.1)==1 & floor
67                 ((yd*10)+0.1)==0
68                 x1(1,i)=1;c=0;
69             elseif floor((xd*10)+0.1)==0 & floor
70                 ((yd*10)+0.1)==1
71                 x1(1,i)=1;c=1;
72         end
73     end
74     disp(x1,'Subtraction of 45 from 228=')
75 endfunction
76 add(10101101,10001101)
77 sub(11100100,00101101)

```

Scilab code Exa 1.16 Addition

```

1 //Example 1.16
2 //Addition
3 //Page no. 14
4 clc;close;clear;
5
6 function [x1]=add(x,y)                      // function
7     for addition of binaries
8     c=0;
9     printf('Addition of %.4i and %.4i= ',x,y)
10    for i=1:4

```

```

10          x1(1,i)=0
11      end
12  for i=4:-1:1
13      x=x/10;
14      xd=x-fix(x)
15      x=x-xd;
16      y=y/10;
17      yd=y-fix(y)
18      y=y-yd;
19      if c==1 then
20          if floor((xd*10)+0.1)==1 & floor((yd*10)
21              +0.1)==1 then
22              x1(1,i)=1;c=1;
23          elseif floor((xd*10)+0.1)==0 & floor((yd
24              *10)+0.1)==0
25              x1(1,i)=1;c=0;
26          else
27              x1(1,i)=0;c=1;
28          end
29      else
30          if floor((xd*10)+0.1)==1 & floor((yd*10)
31              +0.1)==1 then
32              x1(1,i)=0;c=1;
33          elseif floor((xd*10)+0.1)==0 & floor((yd
34              *10)+0.1)==0
35              x1(1,i)=0;c=0;
36          else
37              x1(1,i)=1;c=0;
38          end
39      end
40      disp(x1)
41  endfunction
42 add(0010,0101);
43 add(1110,1011);
44 add(1110,0101);
45 add(0010,1011);

```

```
44 add(1110,0010);  
45 add(0000,0000);
```

Scilab code Exa 1.17 Addition

```
1 //Example 1.17  
2 //Addition  
3 //Page no. 14  
4 clc;close;clear;  
5  
6 function [x1]=add(x,y) //function  
    for addition of binaries  
7     c=0;  
8     printf('Addition of %.4i and %.4i= ',x,y)  
9     for i=1:5  
10        x1(1,i)=0  
11    end  
12    for i=5:-1:1  
13        x=x/10;  
14        xd=x-fix(x)  
15        x=x-xd;  
16        y=y/10;  
17        yd=y-fix(y)  
18        y=y-yd;  
19        if c==1 then  
20            if floor((xd*10)+0.1)==1 & floor((yd*10)  
+0.1)==1 then  
21                x1(1,i)=1;c=1;  
22            elseif floor((xd*10)+0.1)==0 & floor((yd  
*10)+0.1)==0  
23                x1(1,i)=1;c=0;  
24            else  
25                x1(1,i)=0;c=1;
```

```

26         end
27     else
28         if floor((xd*10)+0.1)==1 & floor((yd*10)
29             +0.1)==1 then
30             x1(1,i)=0;c=1;
31         elseif floor((xd*10)+0.1)==0 & floor((yd
32             *10)+0.1)==0
33             x1(1,i)=0;c=0;
34         else
35             x1(1,i)=1;c=0;
36         end
37     end
38     disp(x1)
39 endfunction
40
41 add(0010,0101);
42 add(1101,1010);
43 add(1101,0101);
44 add(0010,1010);
45 add(1101,0010);
46 add(1111,0000);

```

Scilab code Exa 1.18 Addition

```

1 //Example 1.18
2 //Addition
3 //Page no. 15
4 clc;close;clear;
5
6 function [x1]=add(x,y)           //function
    for addition of binaries

```

```

7   c=0;
8   printf('Addition of %.4f and %.4f = ',x,y)
9   for i=1:5
10    x1(1,i)=0
11   end
12   for i=5:-1:1
13    x=x/10;
14    xd=x-fix(x)
15    x=x-xd;
16    y=y/10;
17    yd=y-fix(y)
18    y=y-yd;
19    if c==1 then
20      if floor((xd*10)+0.1)==1 & floor((yd*10)
21          +0.1)==1 then
22        x1(1,i)=1;c=1;
23      elseif floor((xd*10)+0.1)==0 & floor((yd
24          *10)+0.1)==0
25        x1(1,i)=1;c=0;
26      else
27        x1(1,i)=0;c=1;
28      end
29    else
30      if floor((xd*10)+0.1)==1 & floor((yd*10)
31          +0.1)==1 then
32        x1(1,i)=0;c=1;
33      elseif floor((xd*10)+0.1)==0 & floor((yd
34          *10)+0.1)==0
35        x1(1,i)=0;c=0;
36      else
37        x1(1,i)=1;c=0;
38      end
39    end
40    disp(x1)
41 endfunction
42

```

```
41 add(0100,0101);  
42 add(1100,1011);  
43 add(1000,1000);
```

Scilab code Exa 1.19 Addition

```
1 //Example 1.19  
2 //Addition  
3 //Page no. 15  
4 clc;close;clear;  
5  
6 function [x1]=add(x,y) // function  
    for addition of binaries  
7     c=0;  
8     printf('Addition of %.4i and %.4i= ',x,y)  
9     for i=1:5  
10        x1(1,i)=0  
11    end  
12    for i=5:-1:1  
13        x=x/10;  
14        xd=x-fix(x)  
15        x=x-xd;  
16        y=y/10;  
17        yd=y-fix(y)  
18        y=y-yd;  
19        if c==1 then  
20            if floor((xd*10)+0.1)==1 & floor((yd*10)  
+0.1)==1 then  
21                x1(1,i)=1;c=1;  
22            elseif floor((xd*10)+0.1)==0 & floor((yd  
*10)+0.1)==0  
23                x1(1,i)=1;c=0;  
24            else
```

```

25           x1(1,i)=0;c=1;
26       end
27   else
28       if floor((xd*10)+0.1)==1 & floor((yd*10)
29           +0.1)==1 then
30           x1(1,i)=0;c=1;
31       elseif floor((xd*10)+0.1)==0 & floor((yd
32           *10)+0.1)==0
33           x1(1,i)=0;c=0;
34       else
35           x1(1,i)=1;c=0;
36       end
37   end
38   disp(x1)
39 endfunction
40
41 add(0010,0101);
42 add(11110,11011);
43 add(1000,0101);
44 add(00010,11011);
45 add(11110,00010);
46 add(11111,0000);

```

Scilab code Exa 1.20 Subtraction

```

1 //Example 1.20
2 //Subtraction
3 //Page no. 16
4 clc;close;clear;
5 function [x1]=add(x,y)          //function
    for addition of binaries

```

```

6      c=0;
7      for i=1:5
8          x1(1,i)=0
9      end
10     for i=5:-1:1
11         x=x/10;
12         xd=x-fix(x)
13         x=x-xd;
14         y=y/10;
15         yd=y-fix(y)
16         y=y-yd;
17         if c==1 then
18             if floor((xd*10)+0.1)==1 & floor((yd*10)
19                         +0.1)==1 then
20                 x1(1,i)=1;c=1;
21             elseif floor((xd*10)+0.1)==0 & floor((yd
22                         *10)+0.1)==0
23                 x1(1,i)=1;c=0;
24             else
25                 x1(1,i)=0;c=0;
26             end
27         else
28             if floor((xd*10)+0.1)==1 & floor((yd*10)
29                         +0.1)==1 then
30                 x1(1,i)=0;c=1;
31             elseif floor((xd*10)+0.1)==0 & floor((yd
32                         *10)+0.1)==0
33                 x1(1,i)=0;c=0;
34             else
35                 x1(1,i)=1;c=0;
36             end
37         end
38         disp(x1,'Addition of 173 and 141=')
39     endfunction

```

```
40 add(1000,1000);
```

Scilab code Exa 1.23 Multiplication

```
1 //Example 1.23
2 //Multiplication
3 //Page no. 18
4 clc;clear;close;
5
6 function [x1]=mul(x,y)
7   for i=1:8
8     x1(1,i)=0
9   end
10  printf('Multiplication of %.4i and %.4i = ',x,y)
11  x=x*y;
12  c=0;
13  for i=8:-1:1
14    x=x/10;
15    xd=floor((x-fix(x))*10+0.1)
16    if c==1 then
17      if xd==0 then
18        x1(1,i)=1;c=0
19      elseif xd==1
20        x1(1,i)=0;
21        c=1;
22      elseif xd==2
23        x1(1,i)=1;c=1;
24      end
25    else
26      if xd==0 | xd==1 then
27        x1(1,i)=xd;c=0
28      elseif xd==2
29        x1(1,i)=0;
```

```

30           i=i-1;c=1;
31       end
32   end
33 end
34 disp(x1)
35 endfunction
36 mul(1110,1011);

```

Scilab code Exa 1.24 Multiplication

```

1 //Example 1.24
2 //Multiplication
3 //Page no. 18
4 clc;clear;close;
5
6 function [x1]=mul(x,y)
7   for i=1:8
8     x1(1,i)=0
9   end
10  printf(' Multiplication of %.4i and %.4i = ',x,y)
11  x=x*y;
12  c=0;
13  for i=8:-1:1
14    x=x/10;
15    xd=floor((x-fix(x))*10+0.1)
16    if c==1 then
17      if xd==0 then
18        x1(1,i)=1;c=0
19      elseif xd==1
20        x1(1,i)=0;
21        c=1;
22      elseif xd==2
23        x1(1,i)=1;c=1;

```

```

24         end
25     else
26         if xd==0 | xd==1 then
27             x1(1,i)=xd;c=0
28         elseif xd==2
29             x1(1,i)=0;
30             i=i-1;c=1;
31         end
32     end
33 end
34 disp(x1)
35 endfunction
36 mul(1110,1011);

```

Scilab code Exa 1.25 Division

```

1 //Example 1.25
2 //Division
3 //Page no. 19
4 clc;close;clear;
5 function [co]=com(x,y)
6     co=1;
7     for i=1:length(x)
8         if x(i)>y(i) then
9             break
10        elseif x(i)==y(i)
11            continue
12        else
13            co=0;break
14        end
15    end
16 endfunction
17 function [x1]=sub(x,y)           // function for

```

```

    subtraction of binaries
18      c=0; m=0;
19      for i=1:5
20          x1(1,i)=0
21      end
22      for i=5:-1:1
23          if c==1 then
24              if x(i)==0 & y(i)==1 then
25                  x1(1,i)=0; c=1;
26              elseif x(i)==0 & y(i)==0
27                  x1(1,i)=1; c=0;
28              elseif x(i)==1 & y(i)==1
29                  x1(1,i)=1; c=1;
30              elseif x(i)==1 & y(i)==0
31                  x1(1,i)=0; c=0;
32              end
33          else
34              if x(i)==1 & y(i)==1 then
35                  x1(1,i)=0; c=0;
36              elseif x(i)==0 & y(i)==0
37                  x1(1,i)=0; c=0;
38              elseif x(i)==1 & y(i)==0
39                  x1(1,i)=1; c=0;
40              elseif x(i)==0 & y(i)==1
41                  x1(1,i)=1; c=1;
42              end
43          end
44      end
45      disp(x1, 'Remainder = ')
46 endfunction
47 d1=11011001; d2=01011; d22=[0,0,0,0,0]
48 for i=8:-1:1
49     d3=d1/10;
50     div(1,i)=int(10*(d3-int(d3)))
51     d1=d1/10
52 end
53 for i=5:-1:1
54     d3=d2/10;

```

```

55      d21(1,i)=int(10*(d3-int(d3))+0.5)
56      d2=d2/10
57  end
58  div1(1,1)=0
59  for j=1:4
60      div1(1,j+1)=div(1,j)
61  end
62  for i1=1:5
63      printf('After Step %i : \n',i1)
64      if com(div1,d21)==1 then
65          dis(1,i1)=1
66          n=sub(div1,d21)
67      else
68          dis(1,i1)=0
69          n=sub(div1,d22)
70  end
71  disp(dis,'Divisor = ')
72  if i1==5 then
73      break
74  end
75      for j=1:5
76          if j<5 then
77              div1(1,j)=n(j+1)
78          else
79              div1(1,j)=div(1,i1+4)
80          end
81      end
82
83  printf('\n\n\n')
84 end

```

Scilab code Exa 1.26 Multiplication

```

1 //Example 1.26
2 //Multiplication
3 //Page no. 19
4 clc;clear;close;
5
6
7 function [x1]=mul(x,y)
8     for i=1:8
9         x1(1,i)=0
10    end
11    printf(' Multiplication of %.4i and %.4i = ',x,y)
12    x=x*y;
13    c=0;
14    for i=10:-1:1
15        x=x/10;
16        xd=floor((x-fix(x))*10+0.1)
17        if c==1 then
18            if xd==0 then
19                x1(1,i)=1;c=0
20            elseif xd==1
21                x1(1,i)=0;
22                c=1;
23            elseif xd==2
24                x1(1,i)=1;c=1;
25            end
26        else
27            if xd==0 | xd==1 then
28                x1(1,i)=xd;c=0
29            elseif xd==2
30                x1(1,i)=0;
31                i=i-1;c=1;
32            end
33        end
34    end
35    for i=1:10
36        if x1(1,i)==1 then
37            x1(1,i-1)=1;
38            break

```

```
39         end
40     end
41     disp(x1)
42 endfunction
43 mul(1110,1011);
```

Scilab code Exa 1.29 Normalized Floating Point Representation

```
1 //Example 1.29
2 //Normalized Floating Point Representation
3 //Page no. 23
4 clc;clear;close;
5
6 function []=fp(x)
7     x1=x;
8     if x>0 then
9         for i=1:10
10            x=x/10
11            if int(x)==0 then
12                break
13            end
14        end
15        printf('\'\n
16
17 %i\nNormalized Floating Point
18 Representation of %g = %.4f x 10 ',i,x1,x
19 )
20
21 else
22     for i=1:10
23     x=x*10
24     if ceil(x)~=0 then
25         break
26     end
```

```

22     end
23     x=x/10; i=i-1;
24     printf( '\n
25             -%i\nNormalized Floating Point
26             Representation of %g = %.4f x 10 ^ ,i,x1,x
27         )
28     end
29 endfunction
30
31 x=[25.12,-0.00287,87000];
32 for i=1:3
33     fp(x(i))
34 end

```

Scilab code Exa 1.30 Add

```

1 //Example 1.30
2 //Add
3 //Page no. 26
4 clc;clear;close;
5 a=0.4532e7;b=0.5427e7;
6 c=a+b
7 printf('Addition of %.6g and %.6g = %.6g',a,b,c)

```

Scilab code Exa 1.31 Add

```

1 //Example 1.31

```

```
2 //Add
3 //Page no. 26
4 clc;clear;close;
5 a=0.4532e5;b=0.5427e7;
6 c=a+b
7 printf('Addition of %.4g and %.6g = %.6g',a,b,c)
```

Scilab code Exa 1.32 Add

```
1 //Example 1.32
2 //Add
3 //Page no. 26
4 clc;clear;close;
5 a=0.4532e3;b=0.5427e7;
6 c=a+b
7 printf('Addition of %.2g and %.6g = %.4g',a,b,c)
```

Scilab code Exa 1.33 Add

```
1 //Example 1.33
2 //Add
3 //Page no. 27
4 clc;clear;close;
5 a=[0.4632e3,0.4632e99];b=[0.5427e3,0.5427e99];
6
7 for i=1:2
8     c(i)=a(i)+b(i)
9     printf('\nAddition of %.2g and %.2g = %.5g\n',a(
    i),b(i),c(i))
```

```
10 end
```

Scilab code Exa 1.34 Subtraction

```
1 //Example 1.34
2 //Subtraction
3 //Page no. 27
4 clc;clear;close;
5 a=[0.5427e-3,0.9627e4,0.9627e-99];b=[0.9632e
   -4,0.9622e4,0.9622e-99];
6 for i=1:3
7   c(i)=a(i)-b(i)
8   printf('\nSubtraction of %.2g from %.3g = %.6g\n',
   ,a(i),b(i),c(i))
9 end
```

Scilab code Exa 1.35 Multiplication

```
1 //Example 1.35
2 //Multiplication
3 //Page no. 28
4 clc;clear;close;
5 a=[0.9632e12,0.1132e12,0.1132e52,0.1132e-52];b
   =[0.5427e-15,0.1027e15,0.1027e50,0.1027e-50];
6 for i=1:4
7   c(i)=a(i)*b(i)
8   printf('\nMultiplication of %.3g and %.2g = %.6g
   \n',a(i),b(i),c(i))
9 end
```

Scilab code Exa 1.36 Division

```
1 //Example 1.36
2 //Division
3 //Page no. 28
4 clc;clear;close;
5 a=[0.1132e1,0.1132e-6,0.1132e6];b=[0.1000e-99,0.1000
   e99,0.1000e3];
6 for i=1:3
7     c(i)=a(i)/b(i)
8     printf ('\nDivision of %.2g by %.3g= %.3g\n',a(i)
   ,b(i),c(i))
9 end
```

Chapter 2

Scope of Numerical and Mathematical Methods

Scilab code Exa 2.4 Solving Simultaneous Linear Equation

```
1 //Example 2.4
2 //Solving Simultaneous Linear Equation
3 //Page 36
4 clc;close;clear;
5 //eq1= 5x-331y=3.5
6 //eq2= 6x-397y=5.2
7
8 A=[5 , -331;6 , -397];
9 B=[3 .5;5 .2];
10 C=inv(A)*B;           //finding value by multiplying
    inverse with values
11 disp(C(1,1) , 'Value of x=');
12 disp(C(2,1) , 'Value of y=');
```

Scilab code Exa 2.6 Integration

```
1 //Example 2.6
2 //Integration
3 //Page no. 36
4 clc;clear;close;
5 disp(integrate('1/x','x',exp(-4),1),'Integration
    Value=');      //performing integration with
    respect to dx
```

Chapter 3

Errors and Their Propagation

Scilab code Exa 3.1 Limiting Error

```
1 //Example 3.1
2 //Limiting Error
3 //Page no. 45
4 clc;clear;close;
5 R=1000;
6 e=0.1*1000;           //limiting error calculation
7 printf('Magnitude of the Resistor resistance (R) =\
n%i <= R <= %i',R-e,R+e)
```

Scilab code Exa 3.2 Known Error

```
1 //Example 3.2
2 //Known Error
3 //Page no. 46
4 clc;clear;close;
```

```

5 l=28;d=5;
6 v=%pi*l*(d/2)^2;
7 printf ('\nVolume of Cylinder= %f cu. cm',v);
8 re_d=0.1;re_l=-0.5;
9 re_v=2*re_d+re_l;           // relative error
    computation
10 printf ('\n\nRelative error in volume= %f %%',re_v);

```

Scilab code Exa 3.3 Absolute Relative and Percentage Errors

```

1 //Example 3.3
2 //Absolute , Relative and Percetage Errors
3 //Page no. 48
4 clc;clear;close;
5 x=0.00006;x1=0.00005;
6 ex=x-x1;           //absolute error
7 Ex=ex/x1;          //relative error
8 px=100*Ex;         //percentage error
9 printf ('\nAbsolute Error= %f\nRelative Error= %f\
    nPercentage Error= %f %%',ex,Ex,px);

```

Scilab code Exa 3.4 Absolute Relative and Percentage Errors

```

1 //Example 3.4
2 //Absolute , Relative and Percetage Errors
3 //Page no. 48
4 clc;clear;close;
5 x=100500;x1=100000;
6 ex=x-x1;           //absolute error

```

```

7 Ex=ex/x1;           //relative error
8 px=100*Ex;          //percentage error
9 printf ('\nAbsolute Error= %f\nRelative Error= %f\
           nPercentage Error= %f %%', ex, Ex, px);

```

Scilab code Exa 3.5 Absolute Relative and Percentage Errors

```

1 //Example 3.5
2 //Absolute , Relative and Percentage Errors
3 //Page no. 52
4 clc;clear;close;
5 x=9.12345;y=7.654321;
6 x1=9.1234;y1=7.6543;           //on a 5 decimal computer
7 ex=x-x1;                      //absolute error of x
8 ey=y-y1;                      //absolute error of y
9 z1=x1+y1;
10 printf ('\nAbsolute Error in x= %f',ex);
11 printf ('\nAbsolute Error in y= %f',ey);
12 printf ('\nAddition on a 5 decimal computer yields= %
           .5g',z1);
13 z2=16.777;
14 printf ('\nAbsolute Total Error= %f',x+y-z2);
15 printf ('\nAbsolute Propagated Error= %f',x+y-z1);
16 printf ('\nAbsolute Round-off Error= %.4g',z1-z2);
17 printf ('\nRelative Total Error= %.4g',(x+y-z2)/(x+y)
           );
18 printf ('\nRelative Propagated Error= %.2g',(x+y-z1)
           /(x+y));
19 printf ('\nRelative Round-off Error= %.3g',(z1-z2)/(x
           +y));
20 printf ('\nBound on the propagated relative error= %f
           ',2*10^-4);
21 printf ('\nBound on the total relative error= %f'

```

```
,3*10^-4);  
22 printf ('\nAs we can see that both the propagated and  
total relative error are less than their bound  
values')
```

Chapter 4

Programming Tools and Techniques

Scilab code Exa 4.1 Quadratic Equation

```
1 //Example 4.1
2 //Quadratic Equation
3 //Page no. 96
4 clc;clear;close;
5 a=input("Enter value of a= ");
6 b=input("Enter vlaue of b= ");
7 c=input("Enter value of c= ");
8 x1=(-1*b+sqrt((b^2)-4*a*c))/(2*a);      //1st root
9 x2=(-1*b-sqrt((b^2)-4*a*c))/(2*a);      //2nd root
10 printf ('\n1st Root= %f', x1);
11 printf ('\n2nd Root= %f', x2);
```

Scilab code Exa 4.2 Database Management

Reg. No.	Name of Students	Test 1	Test 2	Test 3	Final	Composite Score
CS/01	C. V. Rajan	12	25	21	35	81
CS/02	B. X. Roy	25	07	23	29	77
CS/03	P. C. Sasikumar	10	27	07	36	73
CS/04	B. D. Box	26	26	26	35	87
CS/05	K. K. Mukherjee	29	0	23	30	82

Topper is:

Reg. No.	Name of Students	Composite Score
CS/04	B. D. Box	87

-->|

Figure 4.1: Database Management

```

1 //Example 4.2
2 //Database Management
3 //Page no. 112
4 clc;clear;close;
5 M
    =[12,25,21,35;25,7,23,29;10,27,7,36;26,26,26,35;29,0,23,30];
        //marks
6
7 // calculation of composite score
8 for i=1:5,
9     j=1;k=0;
10    max1=M(i,j);
11    if(max1<M(i,j+1))
12        max1=M(i,j+1)
13    else
14        k=1;
15    end,
16
17    if(M(i,j+2)>M(i,j+k))
18        max2=M(i,j+2);
19    else
20        max2=M(i,j);
21    end,
22    CS(i,1)=max1+max2+M(i,4);
23 end
24

```

```

25 I=[ 'Reg. No. ', 'Name of Students ', 'Test 1 ', 'Test 2 ', '
      Test 3 ', 'Final ';
26 'CS/01 ', 'C.V. Rajan ', '12 ', '25 ', '21 ', '35 ';
27 'CS/02 ', 'B.X. Roy ', '25 ', '07 ', '23 ', '29 ';
28 'CS/03 ', 'P.C. Sasikumar ', '10 ', '27 ', '07 ', '36 ';
29 'CS/04 ', 'B.D. Box ', '26 ', '26 ', '26 ', '35 ';
30 'CS/05 ', 'K.K. Mukherjee ', '29 ', '0 ', '23 ', '30 '];
31 printf ('\n ')
32 for i=1:6
33     for j=1:6
34         if(j>2)
35             printf ('\t ')
36         end
37
38         printf ('%s    ',I(i,j));
39         if(i~=1)
40             if(j>2)
41                 printf ('\t ')
42             end
43             printf ('      ')
44
45         end
46         if(i==1 & j==6)
47             printf ('Composite Score\n')
48         end
49
50     end
51
52     if(i~=1)
53         printf ('%i\n ',CS(i-1,1));
54     end
55
56 end
57 //disp(CS,'Composite Score ',I);
58 max1=CS(1,1);j=1;
59 for i=2:5
60     if(max1<CS(i,1))
61         max1=CS(i,1);j=i;

```

```
62     end,
63 end
64 printf( '\n\nTopper is : %s %s %s , I(1,1) , I(1,2) , '
65     Composite Score')
66 printf( '\nCS/0%i %t %s %t %t %i , j , I(j+1,2) , CS(j,1))
```

Chapter 5

Solutions of Algebraic and Transcendental Equations

Scilab code Exa 5.1 Bisection Method

```
1 //Example 5.1
2 //Bisection Method
3 //Page no. 145
4 clc;clear;close;
5 deff( 'y=f(x)' , 'y=2^x-3*x' )
6 x1=0;x2=2;e=0.001;i=0;
7 printf('Iteration\tx1\t\tx2\t\tz\t\tf(z)\n')
8 printf(


---


n')
9 while abs(x1-x2)>e
10     z=(x1+x2)/2
11     printf('      %i\t\t%t\t%t\t%t\t%t\n',i,x1,x2,z,f
(z))
12     if f(z)*f(x1)>0
13         x1=z
14     else
```

```

15         x2=z
16     end
17     i=i+1
18 end
19 printf ('\n\nThe solution of this equation is %g
           after %i Iterations ',z,i-1)

```

Scilab code Exa 5.2 Bisection Method

```

1 //Example 5.2
2 //Bisection Method
3 //Page no. 147
4 clc;clear;close;
5 deff ('y=f(x)', 'y=x^x-2*x+2')
6 x1=0;x2=2;e=0.001;i=0;
7 printf ('Iteration\tx1\t\tx2\t\tz\t\tf(z)\n')
8 printf (

```

```

    n')
9 while abs(x1-x2)>e
10     z=(x1+x2)/2
11     printf ('      %i\t\t%f\t\t%f\t\t%f\n',i,x1,x2,z,f
               (z))
12     if f(z)*f(x1)>0
13         x1=z
14     else
15         x2=z
16     end
17     i=i+1
18 end
19 printf ('\n\nThe solution of this equation is %g
           after %i Iterations ',z,i-1)
20

```

```
21 printf('\n\n\nNote : There are computational errors  
in the answer given by the book for this example'  
)
```

Scilab code Exa 5.3 Regula Falsi Method

```
1 //Example 5.3  
2 //Regula Falsi Method  
3 //Page no. 149  
4 clc;clear;close;  
5 deff( 'y=f(x) ', 'y=x^3-3*x-5 ')  
6 x1=2;x2=3;e=0.00001  
7 printf( '\n\tx1\t\tf(x1)\t\tx2\t\tf(x2)\t\tx3\t\tf(x3)  
          ')  
8 printf( '\n  
          ')  
9 for i=0:19  
10     x3=x2*f(x1)/(f(x1)-f(x2))+x1*f(x2)/(f(x2)-f(x1))  
11     printf(' %i\t%f\t%f\t%f\t%f\t%f\n',i,x1,f(x1)  
          ),x2,f(x2),x3,f(x3))  
12     if f(x1)*f(x3)>0 then  
13         x1=x3  
14     else  
15         x2=x3  
16     end  
17     if abs(f(x3))<e then  
18         break  
19     end  
20 end  
21 printf( '\n\nTherefore the solution is %.10g',x3)
```

Scilab code Exa 5.4 Ridders Method

answers given by the book in this example\n\n(value of x1 is used instead of x2)')

Scilab code Exa 5.5 General Iterative Method

```
1 //Example 5.5
2 //General Iterative Method
3 //Page no. 154
4 clc;clear;close;
5 deff('x=f(x)', 'x=sqrt(3+5/x)')
6 printf('n\tx\t\tx(x)\n')
7 printf('-----\n')
8 x=2;
9 for i=1:8
10     printf(' %i\t%.10f\t%.10f\n', i, x, f(x))
11     x=f(x);
12 end
13 printf('\n\nThe solution of this equation after %i
Iterations is %.10f', i, x)
```

Scilab code Exa 5.6 Linear Iterative Method

```
1 //Example 5.6
2 //Linear Iterative Method
3 //Page no. 159
4 clc;clear;close;
5 deff('x=f(x)', 'x=1+sin(x)/10')
6 printf('n\tx\t\tx(x)\n')
7 printf('-----\n')
```

```

8 x=0;
9 for i=1:7
10    printf( ' %i\t%.10f\t%.10f\n' ,i,x,f(x))
11    x=f(x);
12 end
13 printf( '\n\nThe solution of this equation after %i
Iterations is %.10f' ,i,x)

```

Scilab code Exa 5.7 Aitkens Method

```

1 //Example 5.7
2 //Aitkens Method
3 //Page no. 161
4 clc;clear;close;
5 deff( 'x=f(x) ','x=exp(-x)')
6 printf( 'n\tx0\t\tx1\t\tx2\t\tx3\t\ty\t\tdx0\n')
7 printf(

```

```

     n')
8 x0=0.5;e=0.0001
9 for i=1:3
10    x1=f(x0);x2=f(x1);x3=f(x2);
11    y=x3-((x3-x2)^2)/(x3-2*x2+x1)
12    dx0=y-x0;
13
14    printf( ' %i\t%.10f\t%.10f\t%.10f\t%.10f\t%.10f\
t%.10f\n' ,i,x0,x1,x2,x3,y,dx0)
15    x0=y;
16    if abs(x0)<e then
17        break;
18    end
19 end
20 printf( '\n\nThe solution of this equation after %i

```

Iterations is %.10f ', i , y)

Scilab code Exa 5.8 Newton Raphson Method

```
1 //Example 5.8
2 //Newton Raphson Method
3 //Page no. 163
4 clc;clear;close;
5 deff( 'x=f(x) ', 'x=x-exp(-x) ')
6 deff( 'x=f1(x) ', 'x=1+exp(-x) ')
7 printf('n\txn\t\tf(xn)\t\tf1(xn)\t\tXn+1\t\tError\n
')
8 printf(
    n')
9 x0=0.5;e=0.00001
10 for i=1:4
11     x1=x0-f(x0)/f1(x0)
12     e1=abs(x0-x1)
13     printf(' %i\t%.10f\t%.10f\t%.10f\t%.10f\t%.10f\n
        ',i-1,x0,f(x0),f1(x0),x1,e1)
14     x0=x1;
15     if abs(x0)<e then
16         break;
17     end
18 end
19 printf('\n\nThe solution of this equation after %i
Iterations is %.10f ',i,x1)
```

Scilab code Exa 5.9 Modified Newton Raphson Method

```
1 //Example 5.9
2 //Modified Newton Raphson Method
3 //Page no. 165
4 clc;clear;close;
5 deff( 'x=f(x) ', 'x=exp(x)-3*x-sin(x)')
6 deff( 'x=f1(x) ', 'x=exp(x)-3-cos(x)')
7 printf( 'n\xtn\t\tf(xn)\t\tf1(xn)\t\tXn+1\t\tError\n'
    ')
8 printf( '
    n')
9 x0=0;e=0.00001
10 for i=1:4
11     x1=x0-f(x0)/f1(x0)
12     e1=abs(x0-x1)
13     printf( ' %i\t%.10f\t%.10f\t%.10f\t%.10f\t%.10f\n'
        ,i-1,x0,f(x0),f1(x0),x1,e1)
14     x0=x1;
15     if abs(x0)<e then
16         break;
17     end
18 end
19 printf( '\n\nTherefore, the root is %.10f',x1)
```

Scilab code Exa 5.10 Newton Raphson Method

```
1 //Example 5.10
2 //Newton Raphson Method
3 //Page no. 167
4 clc;clear;close;
5 deff( 'x=f(x) ', 'x=x*exp(-x)')
```

```

6 deff( 'x=f1(x)', 'x=exp(-x)-x*exp(-x)')
7 printf('n\txn\t\tf(xn)\t\tf1(xn)\t\tXn+1\t\tError\n')
8 printf(


---


9 n')
10 x0=2; e=0.00001
11 for i=1:11
12     x1=x0-f(x0)/f1(x0)
13     e1=abs(x0-x1)
14     printf(' %i\t%.10f\t%.10f\t%.10f\t%.10f\t%.10f\n',
15             i-1, x0, f(x0), f1(x0), x1, e1)
16     x0=x1;
17     if abs(x0)<e then
18         break;
19     end
20 end
21 printf('\n\nTherefore, this is not convergent (i.e.)\n'
22       'divergent')

```

Scilab code Exa 5.11 Newton Raphson Method

```
1 //Example 5.11
2 //Newton Raphson Method
3 //Page no. 167
4 clc; clear; close;
5 deff( 'x=f(x)' , 'x=x^3-x-3' )
6 deff( 'x=f1(x)' , 'x=3*x^2-1' )
7 printf( 'n\xtn\t\tf(xn)\t\tf1(xn)\t\tXn+1\tError\n'
        ')
8 printf( '


---


n')
```

```

9 x0=0;e=0.00001
10 for i=1:11
11     x1=x0-f(x0)/f1(x0)
12     e1=abs(x0-x1)
13     printf( '%i\t%.10f\t%.10f\t%.10f\t%.10f\t%.10f\n'
14         ,i-1,x0,f(x0),f1(x0),x1,e1)
15     x0=x1;
16     if abs(x0)<e then
17         break;
18     end
19 printf('\n\nTherefore , it is cyclic in nature')

```

Scilab code Exa 5.12 Newton Raphson Method

```

1 //Example 5.12
2 //Newton Raphson Method
3 //Page no. 168
4 clc;clear;close;
5 def('x=f(x)', 'x=atan(x)')
6 def('x=f1(x)', 'x=1/(1+x^2)')
7 printf('n\txn\t\tf(xn)\t\tf1(xn)\t\tXn+1\t\tError\n'
8 printf(
    n')
9 x0=1.45;e=0.00001
10 for i=1:12
11     x1=x0-f(x0)/f1(x0)
12     e1=abs(x0-x1)
13     printf( '%i\t%.5g\t%.5g\t%.5g\t%.5g\t%.5g\t%.5g\n'
14         ,i-1,x0,f(x0),f1(x0),x1,e1)
15     x0=x1;

```

```

15      if abs(x0)<e then
16          break;
17      end
18  end
19 printf('n\nTherefore , it is divergent')

```

Scilab code Exa 5.13 Secant Method

```

1 //Example 5.13
2 //Secant Method
3 //Page no. 170
4 clc;clear;close;
5 deff( 'x=f(x) ', 'x=exp(x)-3*x-sin(x) ')
6 deff( 'x=f1(x) ', 'x=exp(x)-3-cos(x) ')
7 printf('n\txn\t\tf(xn)\t\tXn+1\t\tf(Xn+1)\t\tXn+2\t\
     tError\n')
8 printf(

```

```

         n')
9 x0=0.567123008;x1=1;e=0.00001
10 for i=1:9
11     x2=x1-f(x1)*(x1-x0)/(f(x1)-f(x0))
12     e1=abs(x0-x2)
13     printf(' %i\t%.10f\t%.10f\t%.10f\t%.10f\t%.10f\
         %.10f\n',i-1,x0,f(x0),x1,f(x1),x2,e1)
14     x0=x1;
15     x1=x2
16     if abs(x0)<e then
17         break;
18     end
19 end
20 printf('n\nTherefore , the root is %.10f ',x2)

```

Scilab code Exa 5.14 Kizner Method

```
1 //Example 5.14
2 //Kizner Method
3 //Page no. 172
4 clc;clear;close;
5 h2=0.00001
6 def(f,'x=f(x)', 'x=2*x-3-cos(x)')
7 def(f1,'y=f1(x,y)', 'y=h2/(-x+y)') // function for
differentiation
8 printf('n\th\tc\txn\t\tf(xn)\t\tF(xn)\t\tk1\t\tv\t\t
tXn+1\n')
9 printf('
_____
n')
10 x0=2;e=0.00001;h=0.5;c=0.5;
11 for i=1:11
12     h1=-f(x0);
13     F=f1(f(x0),f(x0+h2))
14     k1=h1*F/2;
15     v=h*f(x0)/(c*(f(x0+c+h)-f(x0+c)))-k1/c;
16     a=0;
17     for j=0:3
18         a=a+(v^j)/factorial(j+1)
19     end
20     x1=x0+k1*a
21     printf(' %i\t%g\t%g\t%.6f\t%.6f\t%.6f\t%.8f\t %
.5f\t%.6f\n',i-1,h,c,x0,f(x0),F,k1,v,x1)
22     x0=x1;
23     if abs(x0)<e then
24         break;
25     end
```

```
26 end
27 printf( '\n\nTherefore , the solution is %.10f ',x1)
```

Scilab code Exa 5.15 Brent Method

```
1 //Example 5.15
2 //Brent Method
3 //Page no. 173
4 clc;clear;close;
5 deff( 'y=f(x) ', 'y=x^2+x-2 ')
6 x1=0;x2=0.5;x3=2;
7 r=f(x2)/f(x3);s=f(x2)/f(x1);t=f(x1)/f(x3);
8 q=(t-1)*(r-1)*(s-1);
9 p=r*t*(s-1)*(x2-x3)-s*(1-r)*(x2-x1)+(t*s-r)*x2
10 printf( 'Root is : %.10g ',x2+(p/q))
```

Scilab code Exa 5.19 Horner Method

```
1 //Example 5.19
2 //Horner Method
3 //Page no. 177
4 clc;clear;close;
5 deff( 'y=f(x,a1,a2,a3,a4) ', 'y=a1*x^3+a2*x^2+a3*x+a4 ')
6
7 k=1;m=2;
8 a=[4;-13;-31;-275];
9 for i=1:10
10     s=1;
```

```

11      si=f(s,a(1),a(2),a(3),a(4))*abs(1/f(s,a(1),a(2),
12          a(3),a(4)))
13      while 1
14          a1=f(s,a(1),a(2),a(3),a(4))*abs(1/f(s,a(1),a
15              (2),a(3),a(4)))
16          if si~=a1 then
17              d(i)=s-1
18              break
19          end
20      end
21      b(1)=a(1)
22      for j=1:3
23          for k=1:4-j
24              b(k+1)=a(k+1)+b(k)*d(i)
25              a(k+1)=b(k+1)
26          end
27      end
28      for j=1:3
29          a(j+1)=10^j*a(j+1)
30      end
31  end
32  printf('The positive root is %i.',d(1))
33  for i=2:10
34      printf('%i',d(i))
35  end

```

Scilab code Exa 5.20 Laguerre Method

```

1 //Example 5.20
2 //Laguerre Method
3 //Page no. 180

```

```

4 clc;clear;close;
5 deff( 'y=f(x) ', 'y=x^3+x^2+10*x-20 ')
6 deff( 'y=f1(x) ', 'y=3*x^2+2*x+10 ')
7 deff( 'y=f2(x) ', 'y=6*x+2 ')
8 n=3;
9 printf( 'i\t n\t xi\t tP(x)\t tP1(x)\t tP2(x)\t tProot\
          tNroot\n' )
10 printf( '
          _____
          n')
11 xi=1
12 for i=0:9
13     Proot=xi-(n*f(xi))/(f1(xi)+sqrt((n-1)*f1(xi)^2-n
          *f(xi)*f2(xi)))
14     Nroot=xi-(n*f(xi))/(f1(xi)-sqrt((n-1)*f1(xi)^2-n
          *f(xi)*f2(xi)))
15     printf( ' %i\t %i\t %f\t %f\t %f\t %f\t %f\t %f\n' , i , n ,
          xi , f(xi) , f1(xi) , f2(xi) , Proot , Nroot )
16     xi=Proot
17 end
18 printf( '\n\nProot = %f\nNroot = %f' , Proot , Nroot )

```

Scilab code Exa 5.21 Mullers Method

```

1 //Example 5.21
2 //Mullers Method
3 //Page no. 182
4 clc;clear;close;
5
6 deff( 'y=f(x) ', 'y=x^3-x-4 ')
7 zi=[1;2;3];
8 s=["i","z2","z0","z1","f2","f0","f1","a0","a1","a2",
      "zr+","zr-"]

```

```

9 li(1)=(zi(3,1)-zi(2,1))/(zi(2,1)-zi(1,1))
10 hi(1)=zi(3,1)-zi(2,1);
11 for i=2:6
12   for j=1:3
13     fz(j,i-1)=f(z(i,j-1))
14   end
15   di(i-1)=1+li(i-1)
16   gi(i-1)=fz(1,i-1)*li(i-1)^2-fz(2,i-1)*di(i-1)^2+
17     fz(3,i-1)*(li(i-1)+di(i-1))
18   D1(i-1)=gi(i-1)+sqrt(gi(i-1)^2-4*fz(3,i-1)*di(i-1)*li(i-1)*(fz(1,i-1)*li(i-1)-fz(2,i-1)*di(i-1)+fz(3,i-1)))
19   D2(i-1)=gi(i-1)-sqrt(gi(i-1)^2-4*fz(3,i-1)*di(i-1)*li(i-1)*(fz(1,i-1)*li(i-1)-fz(2,i-1)*di(i-1)+fz(3,i-1)))
20   if abs(D1(i-1))>abs(D2(i-1)) then
21     li(i)=-2*fz(3,i-1)*di(i-1)/D1(i-1)
22   else
23     li(i)=-2*fz(3,i-1)*di(i-1)/D2(i-1)
24   end
25   hi(i)=li(i)*hi(i-1);
26   z(i-1)=zi(3,i-1)+hi(i)
27   for j=1:2
28     zi(j,i)=zi(j+1,i-1)
29   end
30   zi(3,i)=z(i-1)
31 end
32 for i=1:12
33   if i==1 then
34     printf(s(i))
35     for j=1:5
36       printf(' \t\t\t\t%i ',j-1)
37     end
38   elseif i<=4
39     printf(' \n %s ',s(i))
40     for j=1:5
41       printf(' \t\t\t%.10f ',zi(i-1,j))
42     end

```

```

42     elseif i<=7
43         printf( '\n %s' ,s(i))
44         for j=1:5
45             printf( '\t\t%.10f' ,fz(i-4,j))
46         end
47     elseif i<=8
48         printf( '\n %s' ,s(i))
49         for j=1:5
50             printf( '\t\t%.10f' ,li(j))
51         end
52     elseif i<=9
53         printf( '\n %s' ,s(i))
54         for j=1:5
55             printf( '\t\t%.10f' ,di(j))
56         end
57     elseif i<=10
58         printf( '\n %s' ,s(i))
59         for j=1:5
60             printf( '\t\t%.10f' ,gi(j))
61         end
62     elseif i<=11
63         printf( '\n %s' ,s(i))
64         for j=1:5
65             printf( '\t\t%.10f' ,z(j))
66         end
67     elseif i<=12
68         printf( '\n %s' ,s(i))
69         for j=1:5
70             printf( '\t\t%.10f' ,zi(j))
71         end
72     end
73 end
74 printf( '\n\nAt the end of the %i iteration , the root
          of the equation is %.10f' ,j-2,z(j))

```

Scilab code Exa 5.22 Mullers Method

```
1 //Example 5.22
2 //Mullers Method
3 //Page no. 183
4 clc;clear;close;
5
6 deff( 'y=f(x)', 'y=x^3-x-4')
7 zi=[1;2;3];
8 s=[ "i", "z0", "z1", "z2", "f0", "f1", "f2", "li", "di", "gi",
     "li+1", "hi", "hi+1", "zi+1", "D+", "D-"]
9 li(1)=(zi(3,1)-zi(2,1))/(zi(2,1)-zi(1,1))
10 hi(1)=zi(3,1)-zi(2,1);
11 for i=2:6
12     for j=1:3
13         fz(j,i-1)=f(z(i,j-1))
14     end
15     di(i-1)=1+li(i-1)
16     gi(i-1)=fz(1,i-1)*li(i-1)^2-fz(2,i-1)*di(i-1)^2+
               fz(3,i-1)*(li(i-1)+di(i-1))
17     D1(i-1)=gi(i-1)+sqrt(gi(i-1)^2-4*fz(3,i-1)*di(i-
               1)*li(i-1)*(fz(1,i-1)*li(i-1)-fz(2,i-1)*di(i-
               1)+fz(3,i-1)))
18     D2(i-1)=gi(i-1)-sqrt(gi(i-1)^2-4*fz(3,i-1)*di(i-
               1)*li(i-1)*(fz(1,i-1)*li(i-1)-fz(2,i-1)*di(i-
               1)+fz(3,i-1)))
19     if abs(D1(i-1))>abs(D2(i-1)) then
20         li(i)=-2*fz(3,i-1)*di(i-1)/D1(i-1)
21     else
22         li(i)=-2*fz(3,i-1)*di(i-1)/D2(i-1)
23     end
24     hi(i)=li(i)*hi(i-1);
```

```

25      z(i-1)=zi(3,i-1)+hi(i)
26      for j=1:2
27          zi(j,i)=zi(j+1,i-1)
28      end
29      zi(3,i)=z(i-1)
30  end
31  for i=1:16
32      if i==1 then
33          printf(s(i))
34          for j=1:5
35              printf(' \t\t\t%i ',j-1)
36          end
37      elseif i<=4
38          printf(' \n %s ',s(i))
39          for j=1:5
40              printf(' \t\t%.10f ',zi(i-1,j))
41          end
42      elseif i<=7
43          printf(' \n %s ',s(i))
44          for j=1:5
45              printf(' \t\t%.10f ',fz(i-4,j))
46          end
47      elseif i<=8
48          printf(' \n %s ',s(i))
49          for j=1:5
50              printf(' \t\t%.10f ',li(j))
51          end
52      elseif i<=9
53          printf(' \n %s ',s(i))
54          for j=1:5
55              printf(' \t\t%.10f ',di(j))
56          end
57      elseif i<=10
58          printf(' \n %s ',s(i))
59          for j=1:5
60              printf(' \t\t%.10f ',gi(j))
61          end
62      elseif i<=11

```

```

63     printf( '\n %s' ,s(i))
64     for j=1:5
65         printf( '\t\t%.10f' ,li(j+1))
66     end
67 elseif i<=12
68     printf( '\n %s' ,s(i))
69     for j=1:5
70         printf( '\t\t%.10f' ,hi(j))
71     end
72 elseif i<=13
73     printf( '\n %s' ,s(i))
74     for j=1:5
75         printf( '\t\t%.10f' ,hi(j+1))
76     end
77 elseif i<=14
78     printf( '\n %s' ,s(i))
79     for j=1:5
80         printf( '\t\t%.10f' ,z(j))
81     end
82 elseif i<=15
83     printf( '\n %s' ,s(i))
84     for j=1:5
85         printf( '\t\t%.10f' ,D1(j))
86     end
87 elseif i<=16
88     printf( '\n %s' ,s(i))
89     for j=1:5
90         printf( '\t\t%.10f' ,D2(j))
91     end
92 end
93 end
94 printf( '\n\nAt the end of the %ith iteration , the
    root of the equation is %.10f' ,j-1,z(j))

```

Scilab code Exa 5.23 Bairstow Hitchcock Method

```
1 //Example 5.23
2 //Bairstow Hitchcock Method
3 //Page no. 187
4 clc;clear;close;
5 deff( 'y=f(x,p,q)', 'y=x^2+p*z+q' )
6 a=[1,-1,1,-1,1]
7 a=a';a=[a,a,a,a,a]
8 printf('Iteration -->')
9 for i=1:5
10     printf('\t%i\t',i)
11 end
12 printf('\n
')
13 p(1,1)=-1.2;q(1,1)=0.95;
14 s=["b1","b2","b3","b4","c1","c2","c3","c4","c","dp",
    "dq","p","q"]
15 //s1=[b1;b2;b3;b4;c1;c2;c3;c4;c;dp;dq;p;q]
16 for i=1:5
17     b(1,i)=0;b(2,i)=a(1,i);c(1,i)=0;c(2,i)=a(1,i);
18     for k=1:4
19         b(k+2,i)=a(k+1,i)-p(1,i)*b(k+1,i)-q(1,i)*
            b(k,i)
20         c(k+2,i)=b(k+2,i)-p(1,i)*c(k+1,i)-q(1,i)*
            c(k,i)
21     end
22     cb(1,i)=c(6,i)-b(6,i);
23     dq(1,i)=(b(6,i)*c(4,i)-b(5,i)*cb(1,i))/(c(4,i)
        ^2-cb(1,i)*c(3,i))
24     dp(1,i)=(b(5,i)*c(4,i)-b(6,i)*c(3,i))/(c(4,i)
        ^2-cb(1,i)*c(3,i))
25     p(1,i+1)=p(1,i)+dp(1,i);q(1,i+1)=q(1,i)+dq(1,i)
        ;
26 end
27 for j=1:13
28     printf ('\n      %s\t\t',s(j))
```

```

29 if j<5 then
30     for i=1:5
31         printf( '%.9f\t', b(j+2,i))
32     end
33 elseif j<9 then
34     for i=1:5
35         printf( '%.9f\t', c(j-2,i))
36     end
37 elseif j<10
38     for i=1:5
39         printf( '%.9f\t', cb(1,i))
40     end
41 elseif j<11
42     for i=1:5
43         printf( '%.9f\t', dp(1,i))
44     end
45 elseif j<12
46     for i=1:5
47         printf( '%.9f\t', dq(1,i))
48     end
49 elseif j<13
50     for i=1:5
51         printf( '%.9f\t', p(1,i+1))
52     end
53 else
54     for i=1:5
55         printf( '%.9f\t', q(1,i+1))
56     end
57 end
58 end
59 z=poly(0,'z');
60 a=f(z,p(1,i+1),q(1,i+1));
61 printf( '\n\nRoots for Quadratic Equation Q = ')
62 disp(a)
63 a=roots(a)
64 printf( '\n\tare\n')
65 disp(a(1))
66 disp(a(2))

```

Scilab code Exa 5.24 Bernoulli Method

```
1 //Example 5.24
2 //Bernoulli Method
3 //Page no. 189
4 clc;clear;close;
5
6 a=[1,-8,-15,10];
7 for i=1:2
8     c(i)=0;
9 end
10 c(3)=1;
11 for k=4:13
12     c(k)=-(a(2)*c(k-1)+a(3)*c(k-2)+a(4)*c(k-3))
13     r(k-3)=c(k)/c(k-1)
14 end
15 disp(c, 'Ck Values')
16 disp(r, 'Rk Values')
17 disp(r(k-3), 'Therefore the exact root is =')
```

Scilab code Exa 5.25 Graeffe Method

```
1 //Example 5.25
2 //Graeffe Method
3 //Page no. 191
4 clc;clear;close;
5
```

```

6  a=[1,-6,11,-6]
7  k=0;
8  for k=2:6
9    for i=1:4
10      a(k,i)=(-1)^(i-1)*(a(k-1,i))^2
11      j=1;
12      while i+j<5 & i+j>2
13        a(k,i)=a(k,i)+(-1)^(i-j-1)*2*(a(k-1,i-j)
14          )*a(k-1,i+j)
15        break
16      end
17    end
18  end
19  printf ('\t\t\t\t\t\ta1\t\t\t\t\ta2\t\t\t\t\ta3\n  k\ta0\ta1\t
20    \t--\t\ta2\t\t--\t\ta3\t\t--\t\n\t\t\t\ta0\t\t\t\
21  ta1\t\t\t\t\ta2')
22  printf (' %i\t%g\t%.4g\t\t%.5g\t\t%.9g\t\t%.8g\
23    \t\t%.10g\n',i-1,a(i,1),a(i,2),abs(a(i,2)/
24    a(i,1))^(1/(2^(i-1))),a(i,3),abs(a(i,3)/a(i
25    ,2))^(1/(2^(i-1))),a(i,4),abs(a(i,4)/a(i,3))
26    ^(1/(2^(i-1))))
27  end
28  for i=5:6
29    printf (' %i\t%g\t%.4g\t\t%.5g\t\t%.9g\t\t%.8g\t\t%.7g\
30    \t\t%.10g\n',i-1,a(i,1),a(i,2),abs(a(i,2)/a(i,1)
31    )^(1/(2^(i-1))),a(i,3),abs(a(i,3)/a(i,2))
32    ^(1/(2^(i-1))),a(i,4),abs(a(i,4)/a(i,3))
33    ^(1/(2^(i-1))))
34  end
35  printf ('\n\nThe Absolute Values of the roots are %g,
36  %.8g and %g',abs(a(i,2)/a(i,1))^(1/(2^(i-1))),
37  abs(a(i,3)/a(i,2))^(1/(2^(i-1))),abs(a(i,4)/a(i
38  ,3))^(1/(2^(i-1))))
```

Scilab code Exa 5.26 QD Method

```
1 //Example 5.26
2 //QD Method
3 //Page no. 194
4 clc;clear;close;
5
6 a=[32,-48,18,-1]
7 for i=1:5
8     e(i,1)=0;
9     e(i,4)=0;
10 end
11 q(1,1)=-a(2)/a(1);
12 q(1,2)=0;q(1,3)=0;
13 e(1,2)=a(3)/a(2);
14 e(1,3)=a(4)/a(3);
15 for i=2:16
16     for j=1:3
17         q(i,j)=e(i-1,j+1)+q(i-1,j)-e(i-1,j)
18     end
19     for j=1:2
20         e(i,j+1)=e(i-1,j+1)*q(i,j+1)/q(i,j)
21     end
22 end
23 printf('e0\t\tq1\t\tel1\t\tq2\t\tel2\t\tq3\t\tel3\n')
24 printf(
25
26
27
28 n')
25 for i=1:14
26     for j=1:3
27         printf('\t\t%.10f\t',q(i,j))
28     end
```

```

29     printf( '\n' )
30     for j=1:4
31         printf( '%.10f\t\t\t', e(i,j) )
32     end
33     printf( '\n' )
34 end
35 printf( '\t\t%.10f\t\t\t%.10f\t\t\t%.10f\n', q(15,1), q(15,2), q(15,3) )
36 printf( '\nThe exact roots are \t%.10f      and      %.10f ', q(15,1), q(15,3) )

```

Scilab code Exa 5.27 Linear Iteration Method

```
1 //Example 5.27
2 //Linear Iteration Method
3 //Page no. 198
4 clc;clear;close;
5
6 def(f,'x=f(x)', 'x=20/(x^2+2*x+10)')
7 printf('n\tx\t\tf(x)\n')
8 printf('-----\n')
9 x=1;
10 for i=1:19
11     printf(' %i\t%.10f\t%.10f\n', i, x, f(x))
12     x1=x;
13     x=f(x);
14 end
15 printf('\n\nx = %.10f', x1)
```

Scilab code Exa 5.28 Aitkens Method

```

1 //Example 5.28
2 //Aitkens Method
3 //Page no. 199
4 clc;clear;close;
5
6 def(f,'x=f(x)', 'x=20/(x^2+2*x+10)')
7 printf('n\tx0\tx1\tx2\tx3\ty\ttx0\n')
8 printf(
n')
9 x0=1;e=0.0001
10 for i=1:3
11     x1=f(x0);x2=f(x1);x3=f(x2);
12     y=x3-((x3-x2)^2)/(x3-2*x2+x1)
13     dx0=y-x0;
14
15     printf(' %i\t%.10f\t%.10f\t%.10f\t%.10f\t%.10f\t%.10f\
t%.10f\n',i,x0,x1,x2,x3,y,dx0)
16     x0=y;
17     if abs(x0)<e then
18         break;
19     end
20 end
21 printf('\n\nThe solution of this equation after %i
Iterations is %.10f',i,y)

```

Scilab code Exa 5.29 Newton Raphson Method

```
1 //Example 5.29  
2 //Newton Raphson Method  
3 //Page no. 199
```

```

4 clc;clear;close;
5 deff( 'x=f(x) ', 'x=x^3+2*x^2+10*x-20 ')
6 deff( 'x=f1(x) ', 'x=3*x^2+4*x+10 ')
7 printf( 'n\txn\t\tf(xn)\t\tf1(xn)\t\tXn+1\t\tError\n'
    ')
8 printf(


---


    n')
9 x0=0.1; e=0.00001
10 for i=1:4
11     x1=x0-f(x0)/f1(x0)
12     e1=abs(x0-x1)
13     printf( ' %i\t%.10f\t%.10f\t%.10f\t%.10f\t%.10f\n'
        ,i-1,x0,f(x0),f1(x0),x1,e1)
14     x0=x1;
15     if abs(x0)<e then
16         break;
17     end
18 end
19 printf( '\n\nThe solution of this equation after %i
Iterations is %.10f ',i,x1)

```

Scilab code Exa 5.31 Secant Method

```

1 //Example 5.31
2 //Secant Method
3 //Page no. 200
4 clc;clear;close;
5 deff( 'x=f(x) ', 'x=(x-0.6)*(x-1.3)^2*(x-2)^3+0.01234*
    log(x)')
6 printf( 'n\txn\t\tf(xn)\t\tXn+1\t\tf(Xn+1)\t\tXn+2\t\t
    Error\n')
7 printf(

```

```

        n')
8 x0=0.1;x1=1.2;e=0.00001
9 for i=1:7
10      x2=x1-f(x1)*(x1-x0)/(f(x1)-f(x0))
11      e1=abs(x0-x2)
12      printf(' %i\t%.10f\t%.10f\t%.10f\t%.10f\t%.10f\
13          %.10f\n',i-1,x0,f(x0),x1,f(x1),x2,e1)
14      x0=x1;
15      x1=x2
16      if abs(x0)<e then
17          break;
18      end
19 printf('\n\nTherefore , the root is %.10f',x2)

```

Scilab code Exa 5.32 Regula Falsi Newton Raphson and Mullers Method

```

1 //Example 5.32
2 //Regula Falsi , Newton Raphson and Mullers Method
3 //Page no. 201
4 clc;clear;close;
5 deff('x=f(x)', 'x=x^5-3.7*x^4+7.4*x^3-10.8*x^2+10.8*x
-6.8')
6 deff('x=f1(x)', 'x=5*x^4-4*3.7*x^3+3*7.4*x^2-21.6*x
+10.8')
7 //newton raphson
8 printf('n\txn\t\t\tf(xn)\t\t\tf1(xn)\t\t\tXn+1\t\tError\n
')
9 printf(
        n')
10 x0=1.5;e=0.00001

```

```

11 for i=1:4
12     x1=x0-f(x0)/f1(x0)
13     e1=abs(x0-x1)
14     printf( ' %i\t%.10f\t%.10f\t%.10f\t%.10f\t%.10f\n'
15             ,i-1,x0,f(x0),f1(x0),x1,e1)
16     x0=x1;
17     if abs(x0)<e then
18         break;
19     end
20 end
21 printf('
22 //regula falsi
23 x1=1;x2=2;e=0.00001
24 printf('n\tx1\t\tf(x1)\t\tx2\t\tf(x2)\t\tx3\t\tf(x3)
25 printf('
26 for i=0:7
27     x3=x2*f(x1)/(f(x1)-f(x2))+x1*f(x2)/(f(x2)-f(x1))
28     printf( ' %i\t%f\t%f\t%f\t%f\t%f\n' ,i,x1,f(x1)
29             ,x2,f(x2),x3,f(x3))
30     if f(x1)*f(x3)>0 then
31         x1=x3
32     else
33         x2=x3
34     end
35     if abs(f(x3))<e then
36         break
37     end
38 end
39 printf('
40 //mullers method

```

```

41 zi=[1;2;3];
42 s=["i","z0","z1","z2","f0","f1","f2","li","di","gi",
     "li+1","hi","hi+1","zi+1","D+","D_"]
43 li(1)=(zi(3,1)-zi(2,1))/(zi(2,1)-zi(1,1))
44 hi(1)=zi(3,1)-zi(2,1);
45 for i=2:6
46   for j=1:3
47     fz(j,i-1)=f(zi(j,i-1))
48   end
49   di(i-1)=1+li(i-1)
50   gi(i-1)=fz(1,i-1)*li(i-1)^2-fz(2,i-1)*di(i-1)^2+
      fz(3,i-1)*(li(i-1)+di(i-1))
51   D1(i-1)=gi(i-1)+sqrt(gi(i-1)^2-4*fz(3,i-1)*di(i-1)*li(i-1)*(fz(1,i-1)*li(i-1)-fz(2,i-1)*di(i-1)+fz(3,i-1)))
52   D2(i-1)=gi(i-1)-sqrt(gi(i-1)^2-4*fz(3,i-1)*di(i-1)*li(i-1)*(fz(1,i-1)*li(i-1)-fz(2,i-1)*di(i-1)+fz(3,i-1)))
53   if abs(D1(i-1))>abs(D2(i-1)) then
54     li(i)=-2*fz(3,i-1)*di(i-1)/D1(i-1)
55   else
56     li(i)=-2*fz(3,i-1)*di(i-1)/D2(i-1)
57   end
58   hi(i)=li(i)*hi(i-1);
59   z(i-1)=zi(3,i-1)+hi(i)
60   for j=1:2
61     zi(j,i)=zi(j+1,i-1)
62   end
63   zi(3,i)=z(i-1)
64 end
65 printf('\n\n ')
66 for i=1:16
67   if i==1 then
68     printf(s(i))
69     for j=1:5
70       printf(' \t\t\t %i ',j-1)
71     end
72     printf(' \n')

```

```

        ')
73    elseif i<=4
74        printf( '\n %s', s(i))
75        for j=1:5
76            printf( '\t\t%.10f', zi(i-1, j))
77        end
78    elseif i<=7
79        printf( '\n %s', s(i))
80        for j=1:5
81            printf( '\t\t%.10f', fz(i-4, j))
82        end
83    elseif i<=8
84        printf( '\n %s', s(i))
85        for j=1:5
86            printf( '\t\t%.10f', li(j))
87        end
88    elseif i<=9
89        printf( '\n %s', s(i))
90        for j=1:5
91            printf( '\t\t%.10f', di(j))
92        end
93    elseif i<=10
94        printf( '\n %s', s(i))
95        for j=1:5
96            printf( '\t\t%.10f', gi(j))
97        end
98    elseif i<=11
99        printf( '\n %s', s(i))
100       for j=1:5
101           printf( '\t\t%.10f', li(j+1))
102       end
103   elseif i<=12
104       printf( '\n %s', s(i))
105       for j=1:5
106           printf( '\t\t%.10f', hi(j))
107       end
108   elseif i<=13

```

```

109     printf( '\n %s' ,s(i))
110     for j=1:5
111         printf( '\t\t%.10f' ,hi(j+1))
112     end
113     elseif i<=14
114     printf( '\n %s' ,s(i))
115     for j=1:5
116         printf( '\t\t%.10f' ,z(j))
117     end
118     elseif i<=15
119     printf( '\n %s' ,s(i))
120     for j=1:5
121         printf( '\t\t%.10f' ,D1(j))
122     end
123     elseif i<=16
124     printf( '\n %s' ,s(i))
125     for j=1:5
126         printf( '\t\t%.10f' ,D2(j))
127     end
128     end
129 end
130 printf( '\n\nAt the end of the %ith iteration by
    mullers method, the root of the equation is %.10f
    ',j-1,z(j))

```

Scilab code Exa 5.33 Newton Raphson and Mullers Method

```

1 //Example 5.33
2 //Newton Raphson and Mullers Method
3 //Page no. 202
4 clc;clear;close;
5 deff( 'x=f(x)' , 'x=x^4-8*x^3+18*x^2+0.12*x-24.24' )
6 deff( 'x=f1(x)' , 'x=4*x^3-24*x^2+36*x+0.12' )

```

```

7
8 //newton raphson
9 x9=[1.5 ,2.5 ,2.7 ,3.1;4 ,5 ,14 ,10]
10 for h=1:4
11     x0=x9(1,h);e=0.00001
12 for i=1:x9(2,h)
13     x1=x0-f(x0)/f1(x0)
14     e1=abs(x0-x1)
15     x0=x1;
16     if abs(x0)<e then
17         break;
18     end
19 end
20 printf ('\nThe solution of this equation by newton
raphshon after %i Iterations is %.5f\n',i,x1)
21 end
22
23 //mullers method
24 zx=[1 ,2 ,2.7 ,3.1;2 ,3 ,3.7 ,4.1;3 ,4 ,4.7 ,5.1]
25 zi=[1;2;3];
26 s=["i","z0","z1","z2","f0","f1","f2","li","di","gi",
      "li+1","hi","hi+1","zi+1","D+","D_"]
27 li(1)=(zi(3,1)-zi(2,1))/(zi(2,1)-zi(1,1))
28 hi(1)=zi(3,1)-zi(2,1);
29 for i=2:4
30     for j=1:3
31         fz(j,i-1)=f(zi(j,i-1))
32     end
33     di(i-1)=1+li(i-1)
34     gi(i-1)=fz(1,i-1)*li(i-1)^2-fz(2,i-1)*di(i-1)^2+
                  fz(3,i-1)*(li(i-1)+di(i-1))
35     D1(i-1)=gi(i-1)+sqrt(gi(i-1)^2-4*fz(3,i-1)*di(i-1)*li(i-1)*(fz(1,i-1)*li(i-1)-fz(2,i-1)*di(i-1)+fz(3,i-1)))
36     D2(i-1)=gi(i-1)-sqrt(gi(i-1)^2-4*fz(3,i-1)*di(i-1)*li(i-1)*(fz(1,i-1)*li(i-1)-fz(2,i-1)*di(i-1)+fz(3,i-1)))
37     if abs(D1(i-1))>abs(D2(i-1)) then

```

```

38         li(i)=-2*fz(3,i-1)*di(i-1)/D1(i-1)
39     else
40         li(i)=-2*fz(3,i-1)*di(i-1)/D2(i-1)
41     end
42     hi(i)=li(i)*hi(i-1);
43     z(i-1)=zi(3,i-1)+hi(i)
44     for j=1:2
45         zi(j,i)=zi(j+1,i-1)
46     end
47     zi(3,i)=z(i-1)
48 end
49 printf('\n\nAt the end of the %ith iteration by
    mullers method, the root of the equation is %.10f
    ',j+2,z(j))

```

Scilab code Exa 5.34 QD Method

```

1 //Example 5.34
2 //QD Method
3 //Page no. 202
4 clc;clear;close;
5 a=[1,2,10,-20]
6 for i=1:5
7     e(i,1)=0;
8     e(i,4)=0;
9 end
10 q(1,1)=-a(2)/a(1);
11 q(1,2)=0;q(1,3)=0;
12 e(1,2)=a(3)/a(2);
13 e(1,3)=a(4)/a(3);
14 for i=2:7
15     for j=1:3
16         q(i,j)=e(i-1,j+1)+q(i-1,j)-e(i-1,j)

```

Scilab code Exa 5.35 Newton Raphson Method

```
1 //Example 5.35
2 //Newton Raphson Method
3 //Page no. 203
4 clc;clear;close;
5 deff( 'x=f(x)' , 'x=x^3-30*x^2+2552' )
6 deff( 'x=f1(x)' , 'x=3*x^2-60*x' )
7 //newton raphson
```

```

8 printf( 'n\txn\t\tf(xn)\t\tf1(xn)\t\tXn+1\t\tError\n
')
9 printf(


---


    n')
10 x0=10; e=0.00001
11 for i=1:4
12     x1=x0-f(x0)/f1(x0)
13     e1=abs(x0-x1)
14     printf(' %i\t%.10f\t%.10f\t%.10f\t%.10f\t%.10f\n
        ',i-1,x0,f(x0),f1(x0),x1,e1)
15     x0=x1;
16     if abs(x0)<e then
17         break;
18     end
19 end
20 printf('\n\nThus the ball is submerged upto height
of %.10f cm\n\n',x1)

```

Scilab code Exa 5.36 Secant Method

```

1 //Example 5.36
2 //Secant Method
3 //Page no. 204
4 clc;clear;close;
5 a=8670;c=10^-8;t2=1.4*10^-4;
6 def(f,'x=f(x)', 'x=-t2+log((1-2*x/a)/(2-x/a))*(a*x*c)
/(a+x)')
7
8 printf('n\txn\t\tf(xn)\t\tXn+1\t\tf(Xn+1)\t\tXn+2\t\tError\n
')
9 printf(


---



```

```

        n ')
10 x0=20000; x1=25000; e=0.00001
11 for i=1:8
12     x2=x1-f(x1)*(x1-x0)/(f(x1)-f(x0))
13     e1=abs(x0-x2)
14     printf( '%i\t%.10f\t%.10f\t%.10f\t%.10f\n' ,
15             i-1, x0, f(x0), x1, f(x1), x2, e1)
16     x0=x1;
17     x1=x2
18     if abs(x0)<e then
19         break;
20     end
21 printf( '\n\nTherefore , Rb = %.10f ohm' , x2)

```

Scilab code Exa 5.37 Newton Raphson Method

```

1 //Example 5.37
2 //Newton Raphson Method
3 //Page no. 204
4 clc;clear;close;
5 p=1.1;T=250;R=0.082;a=3.6;b=0.043;
6 deff('y=f(v)', 'y=p*v^3-(b*p+R*T)*v^2+a*v-a*b')
7 deff('y=f1(v)', 'y=3*p*v^2-2*(b*p+R*T)*v')
8 printf('n\txn\t\tf(xn)\t\tf1(xn)\t\tXn+1\t\tError\n
      ')
9 printf(
        n ')
10 x0=0.1; e=0.00001
11 for i=1:10
12     x1=x0-f(x0)/f1(x0)
13     e1=abs(x0-x1)

```

```

14     printf( ' %i\t%.10f\t%.10f\t%.10f\t%.10f\t%.10f\n'
15             ',i-1,x0,f(x0),f1(x0),x1,e1)
16     x0=x1;
17     if abs(x0)<e then
18         break;
19     end
20 printf('\n\nTherefore , Volume v = %.10f ltr ',x1)

```

Scilab code Exa 5.38 Newton Raphson Method

```

1 //Example 5.38
2 //Newton Raphson Method
3 //Page no. 205
4 clc;clear;close;
5 deff('y=f(p)', 'y=p^3-9*p^2+33*p-65')
6 deff('y=f1(p)', 'y=3*p^2-18*p+33')
7 printf('n\txn\t\tf(xn)\t\tf1(xn)\t\tXn+1\t\tError\n'
8 )
8 printf(
9
10 for i=1:10
11     x1=x0-f(x0)/f1(x0)
12     e1=abs(x0-x1)
13     printf( ' %i\t%.10f\t%.10f\t%.10f\t%.10f\t%.10f\n'
14             ',i-1,x0,f(x0),f1(x0),x1,e1)
15     x0=x1;
16     if abs(x0)<e then
17         break;
18     end

```

```
19 printf( '\n\nTherefore , Market Price at equilibrium =\n        Rs. %.f ',x1)
```

Scilab code Exa 5.39 Newton Raphson Method

```

1 //Example 5.39
2 //Newton Raphson Method
3 //Page no. 205
4 clc;clear;close;
5 deff('y=f(v)', 'y=v^3-20*v+30')
6 deff('y=f1(v)', 'y=3*v^2-20')
7 printf('n\txn\t\tf(xn)\t\tf1(xn)\t\tXn+1\t\tError\n')
8 printf(


---


    n')
9 x0=10;e=0.00001
10 for i=1:10
11     x1=x0-f(x0)/f1(x0)
12     e1=abs(x0-x1)
13     printf(' %i\t%.10f\t%.10f\t%.10f\t%.10f\t%.10f\n',
    ,i-1,x0,f(x0),f1(x0),x1,e1)
14     x0=x1;
15     if abs(x0)<e then
16         break;
17     end
18 end
19 printf('\n\nTherefore , sides are = %.5f m x %.5f m
    x %.5f m',x1,x1,20/x1^2)

```

Scilab code Exa 5.40 Newton Raphson Method

```
1 //Example 5.40
2 //Newton Raphson Method
3 //Page no. 206
4 clc;clear;close;
5 deff( 'y=f(F) ', 'y=-10*F^3-21*F+10 ')
6 deff( 'y=f1(F) ', 'y=-21-30*F^2 ')
7 printf( '\n\txn\t\tf(xn)\t\tf1(xn)\t\tXn+1\t\tError\n'
          ')
8 printf( '
          n')
9 x0=1;e=0.00001
10 for i=1:10
11     x1=x0-f(x0)/f1(x0)
12     e1=abs(x0-x1)
13     printf( ' %i\t%.10f\t%.6f\t%.5f\t%.10f\t%.10f\n' ,
14             i-1,x0,f(x0),f1(x0),x1,e1)
15     x0=x1;
16     if abs(x0)<e then
17         break;
18     end
19 printf( '\n\t\t\tt2\n Therefore , Magnetic Flux = %
          .5f Wb m',x1)
```

Chapter 6

Numerical Methods of Linear Equations Direct Methods

Scilab code Exa 6.1 Gaussian Elimination Method

```
1 //Example 6.1
2 //Gaussian Elimination Method
3 //Page no. 220
4 clc;clear;close;
5
6 A=[5,10,1,28;1,1,1,6;4,8,3,29]; // augmented matrix
7
8 //triangularization
9 for i=1:4
10    B(1,i)=A(1,i)
11    B(2,i)=A(2,i)-(A(2,1)/A(1,1))*A(1,i)
12    B(3,i)=A(3,i)-(A(3,1)/A(1,1))*A(1,i)
13 end
14 disp(A, 'Augmented Matrix=')
15 disp(B, 'Triangulated Matrix=')
16 //back substitution
```

```

17 x(3)=B(3,4)/B(3,3);
18 printf ('\n x(3)=%f\n',x(3))
19 for i=2:-1:1
20     k=0
21     for j=i+1:3
22         k=k+B(i,j)*x(j)
23     end
24     x(i)=(1/B(i,i))*(B(i,4)-k)
25     printf ('\n x(%i)=%f\n',i,x(i))
26 end

```

Scilab code Exa 6.2 Gaussian Elimination Method for TriDiagonal System

```

1 //Example 6.2
2 //Gaussian Elimination Method for Tri-Diagonal
   System
3 //Page no. 222
4 clc;clear;close;
5
6 //equation matrix
7 A=[1,2,0,0;2,3,-1,0;0,4,2,3;0,0,2,-1];
8 K=[5;5;11;10];i=1;
9
10 //initialization
11 w(1)=A(1,2)/A(1,1);
12 g(1)=K(1)/A(1,1);
13 printf ('\n w(%i)=%f',i,w(i));printf ('\n g(%i)=%f',i,g(
   i))
14
15 //computation
16 for i=2:3
17     w(i)=(A(i,i+1))/(A(i,i)-A(i,i-1)*w(i-1))
18     g(i)=(K(i)-A(i,i-1)*g(i-1))/(A(i,i)-A(i,i-1)*w(i
   ))

```

```

        -1))
19      printf(' \nw(%i)=%f', i, w(i))
20      printf(' \ng(%i)=%f', i, g(i))
21  end
22 i=4
23 m=-2
24 g(i)=m*(K(i)-A(i,i-1)*g(i-1))/(A(i,i)-A(i,i-1)*w(i
   -1))
25 x(i)=g(i)
26 printf(' \ng(%i)=%f', i, g(i))
27 printf(' \n\ nx(%i)=%f', i, x(i))
28
29 //solution
30 for i=3:-1:1
31     x(i)=g(i)-w(i)*x(i+1)
32     printf(' \n\ nx(%i)=%f', i, x(i))
33 end

```

Scilab code Exa 6.3 Gauss Jordan Method

```

1 //Example 6.3
2 //Gauss-Jordan Method
3 //Page no. 224
4
5 clc;clear;close;
6
7 A=[5,10,1,28;4,8,3,29;1,1,1,6];           //augmented
   matrix
8
9 for i=1:3
10    j=i
11    while (A(i,i)==0 & j<=3)
12        for k=1:4

```

```

13      B(1,k)=A(j+1,k)
14      A(j+1,k)=A(i,k)
15      A(i,k)=B(1,k)
16    end
17    disp(A)
18    j=j+1
19  end
20  disp(A)
21  for k=4:-1:i
22    A(i,k)=A(i,k)/A(i,i)
23  end
24  disp(A)
25  for k=1:3
26    if(k~=i) then
27      l=A(k,i)/A(i,i)
28      for m=i:4
29        A(k,m)=A(k,m)-l*A(i,m)
30      end
31    end
32
33  end
34  disp(A)
35 end
36
37 for i=1:3
38   printf ('\nx(%i) = %g\n',i,A(i,4))
39 end

```

Scilab code Exa 6.4 Gaussian Elimination Method without Pivoting

```

1 //Example 6.4
2 //Gaussian Elimination Method without Pivoting
3 //Page no. 227

```

```

4 clc;clear;close;
5
6 A=[0.3*10^-11,1,0.7;1,1,0.9]; //augmented
    matrix
7
8 //triangularization
9 for i=1:3
10    B(1,i)=A(1,i)
11    B(2,i)=A(2,i)-(A(2,1)/A(1,1))*A(1,i)
12 end
13 disp(A,'Augmented Matrix=')
14 disp(B,'Triangulated Matrix=')
15
16 //back substitution
17 x(2)=B(2,3)/B(2,2);
18 printf ('\n x(2)=%f\n',x(2))
19 for i=1:-1:1
20    k=0
21    for j=i+1:2
22       k=k+B(i,j)*x(j)
23    end
24    x(i)=(1/B(i,i))*(B(i,3)-k)
25    printf ('\n x(%i)=%f\n',i,x(i))
26 end

```

Scilab code Exa 6.5 Dolittle Factorization Method

```

1 //Example 6.5
2 //Dolittle Factorization Method
3 //Page no. 233
4 clc;clear;close;
5
6 A=[2,1,1;1,3,1;1,1,4];

```

```

7  printf( '\tL\t\t * \t\U\t\t =\t\tA ')
8  U(2,1)=0;U(3,1)=0;U(3,2)=0;
9  L(1,2)=0;L(1,3)=0;L(2,3)=0;
10 for i=1:3
11     L(i,i)=1
12 end
13 for i=1:3
14     U(1,i)=A(1,i)
15 end
16 L(2,1)=1/U(1,1);
17 for i=2:3
18     U(2,i)=A(2,i)-U(1,i)*L(2,1);
19 end
20 L(3,1)=1/U(1,1);
21 L(3,2)=(A(3,2)-U(1,2)*L(3,1))/U(2,2);
22 U(3,3)=A(3,3)-U(1,3)*L(3,1)-U(2,3)*L(3,2);
23 printf( '\n')
24 for i=1:3
25     for j=1:3
26         printf( '%.2f\t',L(i,j))
27     end
28
29     if (i==2)
30         printf( ' *      ')
31     else
32         printf( '\t')
33     end
34
35     for j=1:3
36         printf( '%.2f\t',U(i,j))
37     end
38     if (i==2)
39         printf( ' =      ')
40     else
41         printf( '\t')
42     end
43     for j=1:3
44         printf( '%.2f\t',A(i,j))

```

```
45     end
46     printf( '\n' )
47 end
```

Scilab code Exa 6.6 Trangularization Method

```
1 //Example 6.6
2 //Trangularization Method
3 //Page no. 236
4 clc;clear;close;
5
6 A=[2,1,1;1,3,1;1,1,4];
7 B=[7;10;15];
8 printf('A can be factorized as follows:\n')
9 printf('\tL\t\t * \t\tU\t\t = \t\tA ')
10 U(2,1)=0;U(3,1)=0;U(3,2)=0;
11 L(1,2)=0;L(1,3)=0;L(2,3)=0;
12 for i=1:3
13     L(i,i)=1
14 end
15 for i=1:3
16     U(1,i)=A(1,i)
17 end
18 L(2,1)=1/U(1,1);
19 for i=2:3
20     U(2,i)=A(2,i)-U(1,i)*L(2,1);
21 end
22 L(3,1)=1/U(1,1);
23 L(3,2)=(A(3,2)-U(1,2)*L(3,1))/U(2,2);
24 U(3,3)=A(3,3)-U(1,3)*L(3,1)-U(2,3)*L(3,2);
25 printf( '\n' )
26 for i=1:3
27     for j=1:3
```

```

28         printf( '%.2f\t', L(i,j))
29     end
30
31     if(i==2)
32         printf( ' *      ')
33     else
34         printf( '\t')
35     end
36
37     for j=1:3
38         printf( '%.2f\t', U(i,j))
39     end
40     if(i==2)
41         printf( ' =      ')
42     else
43         printf( '\t')
44     end
45     for j=1:3
46         printf( '%.2f\t', A(i,j))
47     end
48     printf( '\n')
49 end
50 printf( '\nY=U*X')
51 Y=inv(L)*B
52 X=inv(U)*Y
53 printf( '\n\nX=')
54 for i=1:3
55     printf( '\n    %i', X(i,1))
56 end

```

Scilab code Exa 6.7 Wilkinson Method

1 //Example 6.7

```

2 //Wilkinson Method
3 //Page no. 240
4 clc;clear;close;
5
6 A
    =[0.3*10^5,0.212,0.332;0.216,0.376,0.477;0.173,0.663,0.626];

7 B=[0.235;0.128;0.285];
8 X=inv(A)
9 disp(X*B,'Final Solution = ')

```

Scilab code Exa 6.8 Choleskys Factorization

```

1 //Example 6.8
2 //Cholesky 's Factorization
3 //Page no. 243
4 clc;clear;close;
5
6 A=[1,2,3;2,5,8;3,8,22];
7 U(2,1)=0;U(3,1)=0;U(3,2)=0;
8 for i=1:3
9     for j=1:3
10         if(i==j)
11             k=0;
12             for m=1:i-1
13                 k=k+U(m,i)^2;
14             end
15             U(i,j)=sqrt(A(i,j)-k)
16         end
17         if(j>i)
18             k=0;
19             for m=1:i-1
20                 k=k+U(m,j)*U(m,i);

```

```
21         end
22         U(i,j)=(A(i,j)-k)/U(i,i)
23     end
24 end
25 disp(U, 'Required Matrix (U)=')
```

Scilab code Exa 6.9 Complex System of Linear Equation

```
1 //Example 6.9
2 //Complex System of Linear Equation
3 //Page no. 244
4 clc;clear;close;
5
6 for i=1:7
7     s=0;
8     for j=1:7
9         A(i,j)=j^i
10        s=s+(-1)^(j+1)*A(i,j)
11    end
12    B(i,1)=s;
13 end
14 X=inv(A)*B
15 disp(X, 'The Solution = ')
```

Scilab code Exa 6.10 Solving Matrices

```
1 //Example 6.10
2 //Solving Matrices
```

```
3 //Page no. 244
4 clc;close;clear;
5 warning('off')
6 for i=1:7
7     s=0;
8     for j=1:7
9         A(i,j)=360360/(i+j)
10    end
11    B(i,1)=1;
12 end
13 X=inv(A)*B
14 disp(360360*X,'The Solution by 360360*X= ')
15 disp(X,'Final Solution = ')
```

Chapter 7

Numerical Solutions for Matrix Inversion

Scilab code Exa 7.1 Gauss Jordan Two Array Method

```
1 //Example 7.1
2 //Gauss-Jordan Two Array Method
3 //Page no. 254
4 clc;clear;close;
5
6 A=[2,6,1;3,9,2;0,-1,3];      //matrix
7 C=eye(3,3);                  //Unit Matrix
8 for i=1:3                      //interchange of row 1
    and 2
9     B(1,i)=A(1,i);
10    A(1,i)=A(2,i);
11    A(2,i)=B(1,i);
12    B(2,i)=C(1,i);
13    C(1,i)=C(2,i);
14    C(2,i)=B(2,i);
15 end
16 printf( '\n' )
```

```

17
18 // printing of matrices A and C
19 for i=1:3
20     for j=1:3
21         printf( '%f\t',A(i,j))
22     end
23     printf( '|\\t');
24     for j=1:3
25         printf( '%f\t',C(i,j))
26     end
27     printf( '\n')
28 end
29 printf( '\n\n');
30
31
32 for i=1:3
33     A(1,i)=A(1,i)/3;
34     C(1,i)=C(1,i)/3;
35 end
36
37 // printing of matrices A and C
38 for i=1:3
39     for j=1:3
40         printf( '%f\t',A(i,j))
41     end
42     printf( '|\\t');
43     for j=1:3
44         printf( '%f\t',C(i,j))
45     end
46     printf( '\n')
47 end
48 printf( '\n\n');
49
50 for i=1:3
51     A(2,i)=A(2,i)-2*A(1,i);
52     C(2,i)=C(2,i)-2*C(1,i);
53 end
54

```

```

55 // printing of matrices A and C
56 for i=1:3
57     for j=1:3
58         printf ('%f\t',A(i,j))
59     end
60     printf ('\t');
61     for j=1:3
62         printf ('%f\t',C(i,j))
63     end
64     printf ('\n')
65 end
66 printf ('\n\n');
67
68 for i=1:3          //interchange of row 2 and 3
69     B(1,i)=A(2,i);
70     A(2,i)=A(3,i);
71     A(3,i)=B(1,i);
72     B(2,i)=C(2,i);
73     C(2,i)=C(3,i);
74     C(3,i)=B(2,i);
75 end
76
77 // printing of matrices A and C
78 for i=1:3
79     for j=1:3
80         printf ('%f\t',A(i,j))
81     end
82     printf ('\t');
83     for j=1:3
84         printf ('%f\t',C(i,j))
85     end
86     printf ('\n')
87 end
88 printf ('\n\n');
89
90 for i=1:3
91     A(2,i)=-1*A(2,i);
92     C(2,i)=-1*C(2,i);

```

```

93 end
94 for i=1:3
95     A(1,i)=A(1,i)-3*A(2,i);
96     C(1,i)=C(1,i)-3*C(2,i);
97 end
98
99 // printing of matrices A and C
100 for i=1:3
101     for j=1:3
102         printf('%.f\t',A(i,j))
103     end
104     printf('\t');
105     for j=1:3
106         printf('%.f\t',C(i,j))
107     end
108     printf('\n')
109 end
110 printf('\n\n');
111
112 for i=1:3
113     A(3,i)=-3*A(3,i);
114     C(3,i)=-3*C(3,i);
115 end
116
117 // printing of matrices A and C
118 for i=1:3
119     for j=1:3
120         printf('%.f\t',A(i,j))
121     end
122     printf('\t');
123     for j=1:3
124         printf('%.f\t',C(i,j))
125     end
126     printf('\n')
127 end
128 printf('\n\n');
129
130 for i=1:3

```

```

131     A(1,i)=A(1,i)-A(3,i)*(29/3);
132     C(1,i)=C(1,i)-29*C(3,i)/3;
133 end
134 for i=1:3
135     A(2,i)=A(2,i)+A(3,i)*3;
136     C(2,i)=C(2,i)+C(3,i)*3;
137 end
138
139 // printing of matrices A and C
140 for i=1:3
141     for j=1:3
142         printf('%.f\t',A(i,j))
143     end
144     printf('\n');
145     for j=1:3
146         printf('%.f\t',C(i,j))
147     end
148     printf('\n')
149 end
150 printf('\n\n');
151
152 disp(C,'Inverse Matrix of A')

```

Scilab code Exa 7.2 Inverse in Place without Pivoting

```

1 //Example 7.2
2 //Inverse in Place without Pivoting
3 //Page no. 256
4 clc;clear;close;
5
6 A=[3,-6,7;9,0,-5;5,-8,6];      //matrix
7 B=[3,-6,7;9,0,-5;5,-8,6];      //copied matrix
8 for i=1:3

```

```

9     printf ('\n\nStage %i',i);
10    for j=1:3
11        if(i==j)
12            B(i,j)=1/B(i,j);
13        else
14            B(i,j)=A(i,j)/A(i,i);
15        end,
16    end
17    disp(B)
18    for j=1:3
19        for k=1:3
20            if(i~=j)
21                B(j,k)=A(j,k)-A(j,i)*B(i,k);
22            end,
23        end
24    end
25    disp(B)
26    for j=1:3
27        if(i~=j)
28            B(j,i)=-1*A(j,i)*B(i,i);
29        end,
30    end
31    end
32    disp(B)
33    A=B;
34 end
35 disp(B, 'Inverse of Matrix A=')

```

Scilab code Exa 7.3 Inverse in Place with Pivoting

```

1 //Example 7.3
2 //Inverse in Place with Pivoting
3 //Page no. 258

```

```

4 clc;clear;close;
5
6 A=[3,-6,7;9,0,-5;5,-8,6];           //matrix
7 B=[3,-6,7;9,0,-5;5,-8,6];           //copied matrix
8
9 for i=1:3
10    printf ('\n\nStage %i',i)
11    if(i<3)
12        for j=1:3           //interchange of rows
13            C(i,j)=A(i,j);
14            A(i,j)=A(i+1,j);
15            A(i+1,j)=C(i,j);
16            C(i,j)=B(i,j);
17            B(i,j)=B(i+1,j);
18            B(i+1,j)=C(i,j);
19        end
20    end
21    disp(B)
22    for j=1:3
23        if(i==j)
24            B(i,j)=1/B(i,j);
25        else
26            B(i,j)=A(i,j)/A(i,i);
27        end,
28    end
29    for j=1:3
30        for k=1:3
31            if(i~=j)
32                B(j,k)=A(j,k)-A(j,i)*B(i,k);
33            end,
34        end
35    end
36    for j=1:3
37        if(i~=j)
38            B(j,i)=-1*A(j,i)*B(i,i);
39        end,
40    end
41 end

```

```

42      disp(B)
43      A=B;
44 end
45 for j=1:3      //interchange of column 2 and 3
46      C(j,1)=A(j,2);
47      A(j,2)=A(j,3);
48      A(j,3)=C(j,1);
49 end
50 for j=1:3      //interchange of column 2 and 1
51      C(j,1)=A(j,2);
52      A(j,2)=A(j,1);
53      A(j,1)=C(j,1);
54 end
55 disp(A, 'Inverse of Matrix A=')

```

Scilab code Exa 7.4 Inverse of Triangular Matrices

```

1 //Example 7.4
2 //Inverse of Triangular Matrices
3 //Page no. 260
4 clc;clear;close;
5
6 R=[2,4,-4,0;0,3,-3,-3;0,0,4,2;0,0,0,3];      //matrix
    R
7 for i=4:-1:1
8     for j=4:-1:1
9         if(i>j)
10             Y(i,j)=0;
11         end
12         if(i==j)
13             Y(i,j)=1/R(i,j);
14         end
15         if(i<j)

```

```

16      l=0;
17      for k=i+1:j
18          l=l-R(i,k)*Y(k,j);
19      end
20      Y(i,j)=l/R(i,i);
21  end
22 end
23 end
24 disp(Y, 'Inverse of Matrix R=')

```

Scilab code Exa 7.5 Inverse of Complex Matrices

```

1 //Example 7.5
2 //Inverse of Complex Matrices
3 //Page no. 262
4 clc;clear;close;
5
6 A=[1,-1,0;2,3,4;0,1,2];
7 B=[1,1,3;1,3,-3;-2,-4,-4];
8 P=A+%i*B;
9 disp(P, 'Matrix P=')
10 disp(A, 'Matrix A=');disp(B, 'Matrix B=');
11 A1=inv(A);B1=inv(B);
12 disp(A1, 'Inverse of Matrix A=');
13 disp(B1, 'Inverse of Matrix B=');
14 B1A=B1*A;disp(B1A, 'Inverse(B)*A=');
15 AB1A_B=A*B1A+B;disp(AB1A_B, 'A*Inverse(B)*A+B=');
16 AB1A_B1=inv(AB1A_B);disp(AB1A_B1, 'Inverse(A*Inverse(
    B)*A+B)=');
17 X=B1A*AB1A_B1;disp(X, 'X=');
18 Y=-1*AB1A_B1;disp(Y, 'Y=');
19 Q=X+%i*Y;disp(Q, 'Inverse of Matrix P=')

```

Scilab code Exa 7.6 Iterative Procedure

```
1 //Example 7.6
2 //Iterative Procedure
3 //Page no. 265
4 clc;clear;close;
5
6 A=[3,1,3/2;-5/4,-1/4,-3/4;-1/4,-1/4,-1/4];
7 disp(A, 'Matrix A=');
8 B=[1,1,3.5;1,3,-3;-2,-3,-4];
9 disp(B, 'Assumed Matrix B=');
10 e=0.1;
11
12 // iterations
13 E1=e;k=1;
14 while(E1>=e)
15     printf ('\n\n\nIteration %i\n',k)
16     C=B*(2*eye(3,3)-A*B);disp(C, 'Matrix C=');
17     E=A*C-eye(3,3);disp(E, 'Matrix E=');
18     B=C;printf ('\nInverse of Matrix A after %i
19         iterations=',k);disp(B);
20     E1=0;
21     for i=1:3
22         for j=1:3
23             E1=E1+E(i,j)^2;
24         end
25     end
26     E1=sqrt(E1);
27     k=k+1;
28 end
```

Chapter 8

Numerical Solutions of Linear Systems of Equations Iterative Methods

Scilab code Exa 8.1 Jacobi Method

```
1 //Example 8.1
2 //Jacobi Method
3 //Page no. 273
4 clc;clear;close;
5
6 A=[8,-3,2;4,11,-1;6,3,12];           //equation matrix
7 B=[20;33;36]                         //solution matrix
8 for i=0:19
9     X(i+1,1)=i;
10 end
11 for i=2:4
12     X(1,i)=0;
13 end
14 for r=1:19
15     for i=1:3
```

```

16      k=0;
17      for j=1:3
18          if(i~=j)
19              k=k-A(i,j)*X(r,j+1);
20          end
21      end
22      X(r+1,i+1)=(k+B(i,1))/A(i,i);
23  end
24 end
25 printf(' r \t x(r) \t y(r) \t z(r) ');
26 printf('\n _____ , )
27 disp(X)
28 printf('\n\nAfter 18 iterations exact solution is:\\
nx=%i\ty=%i\tz=%i ',X(19,2),X(19,3),X(19,4))

```

Scilab code Exa 8.2 Gauss Seidel Method

```

1 //Example 8.2
2 //Gauss-Seidel Method
3 //Page no. 274
4 clc;clear;close;
5
6 A=[8,-3,2;4,11,-1;6,3,12];           //equation matrix
7 B=[20;33;36]                         //solution matrix
8 for i=0:10
9     X(i+1,1)=i;
10 end
11 for i=2:4
12     X(1,i)=0;
13 end
14 for r=1:10
15     for i=1:3
16         k1=0;

```

```

17      for j=1:i-1
18
19          k1=k1-A(i,j)*X(r+1,j+1);
20
21      end
22      k2=0;
23      for j=i+1:3
24
25          k2=k2-A(i,j)*X(r,j+1);
26
27      end
28      X(r+1,i+1)=(k1+k2+B(i,1))/A(i,i);
29  end
30 end
31 printf('r\t x(r)\ty(r)\t z(r) ');
32 printf('\n _____, )
33 disp(X)
34 printf('\n\nAfter 9 iterations exact solution is:\nx
= %i\ty=%i\tz=%i',X(10,2),X(10,3),X(10,4))

```

Scilab code Exa 8.3 SOR Method

```

1 //Example 8.3
2 //SOR Method
3 //Page no. 275
4 clc;clear;close;
5
6 A=[5,2,1;-1,4,2;2,-3,10];
7 B=[-12;20;3];
8 w=0.9;
9 for i=0:13
10     X(i+1,1)=i;
11 end

```

```

12 X(1,2)=-2.4;
13 X(1,3)=5;
14 X(1,4)=0.3;
15 for r=1:13
16     for i=1:3
17         k1=0;
18         for j=1:i-1
19
20             k1=k1-A(i,j)*X(r+1,j+1);
21
22         end
23         k2=0;
24         for j=i+1:3
25
26             k2=k2-A(i,j)*X(r,j+1);
27
28         end
29         X(r+1,i+1)=(1-w)*X(r,i+1)+(w*k1+w*k2+w*B(i
29             ,1))/A(i,i);
30     end
31 end
32 printf('    r \t x(r) \t ty(r) \t z(r) ');
33 printf('\n' _____, )
34 ;
34 disp(X);
35 printf('\n\nAfter 12 iterations exact solution is:\'
35 nx=%i\ty=%i\tz=%i',X(13,2),X(13,3),X(13,4));

```

Scilab code Exa 8.4 Gauss Seidel Point Iterative Method

```

1 //Example 8.4
2 //Gauss-Seidel Point Iterative Method
3 //Page no. 278

```

r	x1	x2	x3	x4	x5	x6
0.	0.	0.	0.	0.	0.	0.
1.	0.5	0.95	0.45	-0.045	-0.095	0.245
2.	0.4295	0.91205	0.4593	-0.05543	-0.115705	0.2439725
3.	0.4331923	0.9107508	0.4594523	-0.0575157	-0.1154723	0.2442043
4.	0.4333454	0.9107202	0.4595413	-0.0575014	-0.1154925	0.2442009
5.	0.4333481	0.9107111	0.4595403	-0.0575033	-0.1154912	0.2442012
6.	0.4333490	0.9107111	0.4595403	-0.0575031	-0.1154912	0.2442012
7.	0.4333490	0.9107111	0.4595403	-0.0575031	-0.1154912	0.2442012

After 6 iterations exact solution is:
x1=0.433349 x2=0.910711 x3=0.459540 x4=-0.057503 x5=-0.115491 x6=0.244201
-->

Figure 8.1: Gauss Seidel Point Iterative Method

```

4 clc;clear;close;
5
6
7 A
=[10,1,0,0,0,-1;1,10,1,0,0,0;2,0,20,1,0,0;0,0,1,10,-1,0;0,3,0,0,3
 //equation matrix
8 B=[5;10;10;0;0;5] //solution
matrix
9 for i=1:6
10   for j=1:6
11     if(A(j,j)==0)
12       for k=1:6
13         C(j,k)=A(j,k);
14         A(j,k)=A(j+1,k);
15         A(j+1,k)=C(j,k);
16       end
17     end
18   end
19 end
20 for i=0:7
21   X(i+1,1)=i;
22 end
23 for i=2:7
24   X(1,i)=0;
25 end

```

```

26 for r=1:7
27     for i=1:6
28         k1=0;
29         for j=1:i-1
30
31             k1=k1-A(i,j)*X(r+1,j+1);
32
33         end
34         k2=0;
35         for j=i+1:6
36
37             k2=k2-A(i,j)*X(r,j+1);
38
39         end
40         X(r+1,i+1)=(k1+k2+B(i,1))/A(i,i);
41     end
42 end
43 printf('r      ');
44 for i=1:6
45     printf('x%ii      ',i);
46 end
47 printf('\n
')
48 disp(X)
49 printf('\n\nAfter 6 iterations exact solution is:\n');
50 for i=1:6
51     printf('x%ii=%f      ',i,X(7,i+1));
52 end

```

Scilab code Exa 8.5 Gauss Seidel Point Iterative Method

```

1 //Example 8.5
2 //Gauss-Seidel Point Iterative Method
3 //Page no. 279
4 clc;clear;close;
5
6 A=[2,3,-4,1;1,-2,-5,1;5,-3,1,-4;10,2,-1,2];           // equation matrix
7 B=[3;2;1;-4];                      // solution matrix
8
9 //transformation of the equations
10 for i=1:4
11     A1(1,i)=A(4,i);
12     B1(1,1)=B(4,1);
13 end
14 for i=1:4
15     A1(3,i)=A(2,i);
16     B1(3,1)=B(2,1);
17 end
18 for i=1:4
19     A1(2,i)=A(1,i)-A(2,i);
20     B1(2,1)=B(1,1)-B(2,1);
21 end
22 for i=1:4
23     A1(4,i)=2*A(1,i)-A(2,i)+2*A(3,i)-A(4,i);
24     B1(4,1)=2*B(1,1)-B(2,1)+2*B(3,1)-B(4,1);
25 end
26
27 //printing of transformed equations
28 printf ('\nTransformed Equations are=\n\n')
29 for i=1:4
30     for j=1:4
31         printf ('(%ix(%i)) ',A1(i,j),j);
32         if(j<4)
33             printf (' + ')
34         end
35     end
36     printf (' = %i\n',B1(i,1));
37 end

```

```

38
39 for i=1:4
40     for j=1:4
41         if(A(j,j)==0)
42             for k=1:4
43                 C(j,k)=A(j,k);
44                 A(j,k)=A(j+1,k);
45                 A(j+1,k)=C(j,k);
46             end
47         end
48     end
49 end
50 for i=0:12
51     X(i+1,1)=i;
52 end
53 for i=2:5
54     X(1,i)=0;
55 end
56 for r=1:12
57     for i=1:4
58         k1=0;
59         for j=1:i-1
60
61             k1=k1-A1(i,j)*X(r+1,j+1);
62
63         end
64         k2=0;
65         for j=i+1:4
66
67             k2=k2-A1(i,j)*X(r,j+1);
68
69         end
70         X(r+1,i+1)=(k1+k2+B1(i,1))/A1(i,i);
71     end
72 end
73 printf('\n\n      r      ');
74 for i=1:4
75     printf('x%i      ',i);

```

```

76 end
77 printf( '\n


---


    ')
78 disp(X)
79 printf( '\n\nAfter 11 iterations exact solution is:\n
    ');
80 for i=1:4
81     printf('x%ii=%f      ',i,X(12,i+1));
82 end


---



```

Scilab code Exa 8.6 Block Jacobi Method

```

1 //Example 8.6
2 //Block Jacobi Method
3 //Page no. 281
4 clc;clear;close;
5
6 A
    =[10 ,1 ,0 ,0 ,0 ,-1;1 ,10 ,1 ,0 ,0 ,0;2 ,0 ,20 ,1 ,0 ,0 ;0 ,0 ,0 ,1 ,10 ,-1 ,0 ;0 ,3 ,0 ,0 ,30
        //equation matrix
7 B=[5;10;10;0;0;5]                                //solution
    matrix
8 disp(B, 'B=' ,A , 'A=' )
9 for i=1:3
10     for j=1:3
11         A11(i,j)=A(i,j);
12     end
13     B1(i,1)=B(i,1);
14 end
15 for i=1:3
16     for j=1:3
17         A12(i,j)=A(i,j+3);

```

```

18     end
19 end
20 for i=1:3
21   for j=1:3
22     A21(i,j)=A(i+3,j);
23   end
24 end
25 for i=1:3
26   for j=1:3
27     A22(i,j)=A(i+3,j+3);
28   end
29 B2(i,1)=B(i+3,1);
30 end
31 disp(B2,'B2='),B1,'B1='),A22,'A22='),A21,'A21='),A12,'
      A12='),A11,'A11=');
32 A11_1=inv(A11);A22_1=inv(A22);
33 disp(A22_1,'Inverse of A22='),A11_1,'Inverse of A11='
      )
34 for i=1:3
35   X1(i,1)=0;
36   X2(i,1)=0;
37 end
38 for r=1:2
39   X11=A11_1*(-1*A12*X2+B1);
40   X22=A22_1*(-1*A21*X1+B2);
41   X1=X11;
42   X2=X22;
43   disp(X1,'X1=')
44   disp(X2,'X2=')
45 end
46 for i=1:6
47   if(i<4)
48     X(i,1)=X1(i,1);
49   else
50     X(i,1)=X2(i-3,1);
51   end
52 end
53 disp(X,'X=')

```

```
54 printf ('\n\n\nNote : There is a computation error in  
calculation of X1(2)')
```

Scilab code Exa 8.7 Block Gauss Seidel Method

```
1 //Example 8.7  
2 //Block Gauss-Seidel Method  
3 //Page no. 283  
4 clc;clear;close;  
5  
6 A  
=[10 ,1 ,0 ,0 ,0 ,-1;1 ,10 ,1 ,0 ,0 ;2 ,0 ,20 ,1 ,0 ,0 ;0 ,0 ,1 ,10 ,-1 ,0 ;0 ,3 ,0 ,0 ,30  
//equation matrix  
7 B=[5;10;10;0;0;5] //solution  
matrix  
8 disp(B, 'B=' ,A , 'A=' )  
9  
10 for i=1:2  
11     for j=1:2  
12         A11(i,j)=A(i,j);  
13     end  
14     B1(i,1)=B(i,1);  
15 end  
16 for i=1:2  
17     for j=1:2  
18         A12(i,j)=A(i,j+2);  
19     end  
20     B2(i,1)=B(i+2,1);  
21 end  
22 for i=1:2  
23     for j=1:2  
24         A13(i,j)=A(i,j+4);  
25     end
```

```

26      B3(i,1)=B(i+4,1);
27  end
28  for i=1:2
29      for j=1:2
30          A21(i,j)=A(i+2,j);
31      end
32  end
33  for i=1:2
34      for j=1:2
35          A22(i,j)=A(i+2,j+2);
36      end
37  end
38  for i=1:2
39      for j=1:2
40          A23(i,j)=A(i+2,j+4);
41      end
42  end
43  for i=1:2
44      for j=1:2
45          A31(i,j)=A(i+4,j);
46      end
47  end
48  for i=1:2
49      for j=1:2
50          A32(i,j)=A(i+4,j+2);
51      end
52  end
53  for i=1:2
54      for j=1:2
55          A33(i,j)=A(i+4,j+4);
56      end
57  end
58 disp(B3,'B3=' ,B2,'B2=' ,B1,'B1=' ,A33,'A33=' ,A32,'A32='
      ,A31,'A31=' ,A23,'A23=' ,A22,'A22=' ,A21,'A21=' ,A13
      , 'A13=' ,A12,'A12=' ,A11,'A11=' );
59 A11_1=inv(A11);A22_1=inv(A22);A33_1=inv(A33);
60 disp(A33_1,'Inverse of Matrix A33=' ,A22_1,'Inverse
      of Matrix A22=' ,A11_1,'Inverse of Matrix A11=' );

```

```

61 for i=1:2
62     X1(i,1)=0;
63     X2(i,1)=0;
64     X3(i,1)=0;
65 end
66 for i=1:6
67     X(i,1)=i-1;
68 end
69 for i=2:7
70     X(1,i)=0;
71 end
72 for r=1:5
73     X11=A11_1*(-1*A12*X2+(-1)*A13*X3+B1);
74     X22=A22_1*(-1*A21*X11+(-1)*A23*X3+B2);
75     X33=A33_1*(-1*A31*X11+(-1)*A32*X22+B3);
76     X1=X11;
77     X2=X22;
78     X3=X33;
79     disp(X3, 'X3=' ,X2, 'X2=' ,X1, 'X1=' )
80     for i=2:7
81         if(i<4)
82             X(r+1,i)=X1(i-1,1);
83         end
84         if(i<6 & i>3)
85             X(r+1,i)=X2(i-3,1);
86         end
87         if(i<8 & i>5)
88             X(r+1,i)=X3(i-5,1);
89         end
90     end
91 end
92 printf('\n\nIteration ');
93 for i=1:6
94     printf('      x%i      ',i);
95 end
96 printf('\n
')

```

```

97 disp(X)
98 printf('n\nAfter 4 iterations exact solution is:\n')
      );
99 for i=1:6
100    printf('x%ii=%f      ',i,X(5,i+1));
101 end

```

Scilab code Exa 8.8 Block SOR Method

```

1 //Example 8.8
2 //Block SOR Method
3 //Page no. 284
4 clc;clear;close;
5
6
7 A
     =[10 ,1 ,0 ,0 ,0 ,-1;1 ,10 ,1 ,0 ,0 ,0;2 ,0 ,20 ,1 ,0 ,0;0 ,0 ,1 ,10 ,-1 ,0 ;0 ,3 ,0 ,0 ,3
          //equation matrix
8 B=[5;10;10;0;0;5]                                //solution
          matrix
9 disp(B, 'B=' ,A , 'A=' )
10 w=0.8
11 for i=1:2
12    for j=1:2
13       A11(i,j)=A(i,j);
14    end
15    B1(i,1)=B(i,1);
16 end
17 for i=1:2
18    for j=1:2
19       A12(i,j)=A(i,j+2);
20    end
21    B2(i,1)=B(i+2,1);

```

```

22 end
23 for i=1:2
24     for j=1:2
25         A13(i,j)=A(i,j+4);
26     end
27     B3(i,1)=B(i+4,1);
28 end
29 for i=1:2
30     for j=1:2
31         A21(i,j)=A(i+2,j);
32     end
33 end
34 for i=1:2
35     for j=1:2
36         A22(i,j)=A(i+2,j+2);
37     end
38 end
39 for i=1:2
40     for j=1:2
41         A23(i,j)=A(i+2,j+4);
42     end
43 end
44 for i=1:2
45     for j=1:2
46         A31(i,j)=A(i+4,j);
47     end
48 end
49 for i=1:2
50     for j=1:2
51         A32(i,j)=A(i+4,j+2);
52     end
53 end
54 for i=1:2
55     for j=1:2
56         A33(i,j)=A(i+4,j+4);
57     end
58 end
59 disp(B3,'B3=' ,B2,'B2=' ,B1,'B1=' ,A33,'A33=' ,A32,'A32=

```

```

    ,A31 , 'A31= ',A23 , 'A23= ',A22 , 'A22= ',A21 , 'A21= ',A13
    , 'A13= ',A12 , 'A12= ',A11 , 'A11= ') ;
60 A11_1=inv(A11);A22_1=inv(A22);A33_1=inv(A33);
61 disp(A33_1,'Inverse of Matrix A33= ',A22_1,'Inverse
of Matrix A22= ',A11_1,'Inverse of Matrix A11= ');
62 for i=1:2
63     X1(i,1)=0;
64     X2(i,1)=0;
65     X3(i,1)=0;
66 end
67 for i=1:7
68     X(i,1)=i-1;
69 end
70 for i=2:7
71     X(1,i)=0;
72 end
73 for r=1:6
74     X11=A11_1*((1-w)*X1+(-1)*w*A12*X2+(-1)*w*A13*X3+
        w*B1);
75     X22=A22_1*((1-w)*X2+(-1)*w*A21*X11+(-1)*w*A23*X3
        +w*B2);
76     X33=A33_1*((1-w)*X3+(-1)*w*A31*X11+(-1)*w*A32*
        X22+w*B3);
77     X1=X11;
78     X2=X22;
79     X3=X33;
80     disp(X3,'X3= ',X2,'X2= ',X1,'X1= ')
81     for i=2:7
82         if(i<4)
83             X(r+1,i)=X1(i-1,1);
84         end
85         if(i<6 & i>3)
86             X(r+1,i)=X2(i-3,1);
87         end
88         if(i<8 & i>5)
89             X(r+1,i)=X3(i-5,1);
90         end
91     end

```

```
92 end
93 printf( '\n\nIteration ');
94 for i=1:6
95     printf( '      x%i      ', i );
96 end
97 printf( '\n
_____
')
98 disp(X)
99 printf( '\n\nAfter 5 iterations exact solution is:\n'
);
100 for i=1:6
101     printf( 'x%i=%f      ', i, X(6, i+1));
102 end
```

Chapter 9

Linear Least Squares Problem

Scilab code Exa 9.1 Moore Penrose Generalized Inverse

```
1 //Example 9.1
2 //Moore-Penrose Generalized Inverse
3 //Page no. 292
4 clc;clear;close;
5
6 AT=[3,0,3;0,3,3];
7 A=AT';           // transpose
8 I=inv(AT*A);    // inverse
9 disp(I, 'Inverse of AT*A=' , AT*A, 'AT*A=' , A, 'A=' , AT, 'AT
=') ;
10 A#=I*AT;
11 disp(A#, 'Moore-Penrose Generalized Inverse of A=')
```

Scilab code Exa 9.2 Curve Fitting

```

1 //Example 9.2
2 //Curve Fitting
3 //Page no. 293
4 clc; clear; close;
5 x(1)=0.25;
6 for i=2:6
7     x(1,i)=x(1,i-1)+0.25;
8 end //x values
9 y(1,1)=3.1; y(1,2)=1.7; y(1,3)=1; y(1,4)=0.68; y(1,5)
    =0.42; y(1,6)=0.26; //y values
10
11 //construction of normal equations
12 for i=1:6
13     Y(1,i)=log10(y(1,i));
14 end
15 Ex=0;
16 for i=1:6
17     Ex=Ex+x(1,i);
18 end
19 EY=0;
20 for i=1:6
21     EY=EY+Y(1,i);
22 end
23 Ex2=0;
24 for i=1:6
25     Ex2=Ex2+x(1,i)^2;
26 end
27 ExY=0;
28 for i=1:6
29     ExY=ExY+x(1,i)*Y(1,i);
30 end
31 printf('E x(k)\t y(k)\t Y(k)\t x2(k)\t x(k)*Y(k)')
32 printf('\n')
33 for i=1:6
34     printf('\n%f\t%f\t%f\t%f', x(1,i), y(1,i), Y(1,
        '))

```

```

        i) ,x(1,i)^2,x(1,i)*Y(1,i))

35 end
36 printf('\
')
37 printf('\n%f\t%f\t%f\t%f\t%f',Ex,0,EY,Ex2,ExY)
38 printf('\n
')

n\n')
39 A=[6,Ex;Ex,Ex2];           //system of normal equations
40 B=[EY;ExY];
41 X=inv(A)*B;
42 a=exp(X(1,1));
43 b=-1*X(2,1);
44 for i=1:2
45     for j=1:2
46         printf('%f      ',A(i,j))
47     end
48     if(i==1)
49         printf('*')
50     end
51
52     printf('\ta%i',i);
53     if(i==1)
54         printf(' =')
55     end
56
57     printf('\t%f\n',B(i,1))
58 end
59 printf('\n\nna1=%f\nna2=%f\nna=%f\nnb=%f\n\n',X(1,1),X
(2,1),a,b)
60 printf('The fitted curve is:\n%fx\ny=%f
e',b,a)

```

Scilab code Exa 9.3 Gram Schmidt Orthogonalization or Orthonormalization Process

```
1 //Example 9.3
2 //Gram-Schmidt Orthogonalization / Orthonormalization
3 //Page no. 294
4 clc;clear;close;
5 deff( 'y=f(x,a)' , 'y=sqrt(x(1,a)^2+x(2,a)^2+x(3,a)^2+x
(4,a)^2)');
6 deff( 'y=f1(g,a,h,b)' , 'y=g(1,a)*h(1,b)+g(2,a)*h(2,b)+
g(3,a)*h(3,b)+g(4,a)*h(4,b)');
7
8 U=[1/sqrt(3),-2/sqrt(7),1,0,0,0;0,1/sqrt(7)
,0,1,0,0;1/sqrt(3),1/sqrt(7),0,0,1,0;-1/sqrt(3)
,-1/sqrt(7),0,0,0,1];
9 for i=1:4
10     V(i,1)=U(i,1);
11 end
12 for i=1:4
13     if(f(V,1) ~= 0)
14         W(i,1)=V(i,1)/f(V,1);
15     else
16         W(i,1)=0;
17     end
18 end
19 for j=2:6
20     for i=1:4
21         for l=1:4
22             k(l,1)=0;
23         end
24         for l=1:j-1
25             for m=1:4
26                 w(m,1)=W(m,l);
27             end
28             k=k-(f1(U,j,W,l))*w;
29         end
30         V(i,j)=U(i,j)+k(i,1);
31     end
```

```

32     for i=1:4
33         if(j ~=4)
34             if(f(V,j) ~=0)
35                 W(i,j)=V(i,j)/f(V,j);
36             else
37                 W(i,j)=0;
38             end
39         else
40             W(i,j)=0;
41         end
42     end
43
44 end
45 disp(U, 'U=')
46 disp('W=')
47 printf('\n')
48 for i=1:4
49     for j=1:6
50         printf('%.4f\t\t',W(i,j))
51     end
52     printf('\n')
53 end
54 disp('V=')
55 printf('\n')
56 for i=1:4
57     for j=1:6
58         printf('%.4f\t\t',V(i,j))
59     end
60     printf('\n')
61 end

```

Scilab code Exa 9.4 QR Decomposition

```

1 //Example 9.4
2 //QR Decomposition
3 //Page no. 296
4 clc;clear;close;
5
6 A=[2,1,1;1,3,1;1,1,4];
7 B=A*A';
8 disp(B, 'AT*A=')
9 //cholesky factorization to find R
10 R(2,1)=0;R(3,1)=0;R(3,2)=0;
11 for i=1:3
12     for j=1:3
13         if(i==j)
14             k=0;
15             for m=1:i-1
16                 k=k+R(m,i)^2;
17             end
18             R(i,j)=sqrt(B(i,j)-k)
19         end
20         if(j>i)
21             k=0;
22             for m=1:i-1
23                 k=k+R(m,j)*R(m,i);
24             end
25             R(i,j)=(B(i,j)-k)/R(i,i)
26         end
27     end
28 end
29 //cholesky factorization end
30 disp(R, 'Upper Triangular Matrix (R)=')
31 R_1=inv(R);
32 disp(R_1, 'Inverse of R')
33 Q=A*R_1;
34 disp(Q, 'Orthogonal Matrix Q=')

```

Scilab code Exa 9.5 Vector Computation

```
1 //Example 9.5
2 //Vector Computation
3 //Page no. 299
4 clc;clear;close;
5
6 X=[2,3,0,1];
7 n=X(1);
8 for i=2:4
9     if(n<X(i))
10        n=X(i);
11    end
12 end
13 printf('Maximum Value (n)=%i\n',n)
14 for i=1:4
15    X(i)=X(i)/n;
16 end
17 disp(X, 'Normalized X=')
18 k=0;
19 for i=1:4
20    k=k+X(i)^2;
21 end
22 sigma=X(1)*abs(1/X(1))*sqrt(k);
23 printf('nsigma=%f\n',sigma);
24 X(1)=X(1)+sigma;
25 printf('\nModified x1 = %g\n',X(1))
26 for i=1:4
27    U(1,i)=X(i);
28 end
29 disp(U, 'U=')
30 p=sigma*X(1); sigma=n*sigma;
```

```
31 printf ('\n p = %f\n\n sigma = %f',p,sigma);
32 printf ('\n\nNote : There is a computation error in
calculation of U1')
```

Scilab code Exa 9.6 House Holder Transformation

```
1 //Example 9.6
2 //House Holder Transformation
3 //Page no. 300
4 clc;clear;close;
5
6 A=[4 ,2 ,1;2 ,5 ,-2;1 ,-2 ,7]
7 disp(A, 'A=')
8 k=0;
9 for j=2:3
10     k=k+A(j ,1) ^2;
11 end
12 a=A(2 ,1)*abs(1/A(2 ,1))*sqrt(k);
13 disp(a, 'alpha=')
14 U=[0;a+A(2 ,1);A(3 ,1)];
15 disp(U, 'U=')
16 U1=U'*U;
17 disp(U1, 'UT*U=')
18 U2=U*U';
19 disp(U2, 'U*UT=')
20 P=eye(3 ,3)-(2*U2)/U1;
21 disp(P, 'P=');
22 B=P*A*P;
23 disp(B, 'B=');
24 printf ('\n\nThere are computation error in the
answers given by the book in this example\n\n(a22
value error in U*UT)')
```

Scilab code Exa 9.7 Givens QR Method

```
1 //Example 9.7
2 //Givens QR Method
3 //Page no. 303
4 clc;clear;close;
5
6 A=[4 ,2 ,1 ;2 ,5 ,-2 ;1 ,-2 ,7]
7 defc('y=c(i ,j )','y=A(j ,j )/sqrt((A(i ,j )^2+A(j ,j )^2))'
)
8 defc('y=s(i ,j )','y=A(i ,j )/sqrt((A(i ,j )^2+A(j ,j )^2))'
)
9 disp(A,'A=')
10 R=A;Q=eye(3 ,3 );
11 m=1;
12 for j=1:2
13     for i=j+1:3
14         for k=1:3
15             for l=1:3
16                 if(k==l)
17                     if(k==i | k==j)
18                         C(k,l)=c(i ,j )
19                     else
20                         C(k,l)=1
21                     end
22                 end
23                 if(k>l)
24                     if(k==i & l==j)
25                         C(k,l)=-1*s(i ,j )
26                     else
27                         C(k,l)=0
28                     end
```

```

29         end
30         if(k<1)
31             if(k==j & l==i)
32                 C(k,l)=s(i,j)
33             else
34                 C(k,l)=0
35             end
36         end
37     end
38
39     printf( '\n\n Iteration %i ',m)
40     m=m+1
41     disp(C, 'C=');
42     R=C*R;
43     Q=Q*C ;
44     disp(Q, 'Q=' ,R, 'R=')
45 end
46 end
47 disp(Q*R, 'Q*R=A=' ) // verification

```

Scilab code Exa 9.8 Recursive Least Square Method

```

1 //Example 9.8
2 //Recursive Least-Square Method
3 //Page no. 308
4 clc;clear;close;
5
6 A0=[3,0;0,3;3,3];
7 B0=[2;2;2];
8 A1=[6,3];B1=[6];
9 A0T=A0';
10 G0=A0T*A0;
11 disp(G0, 'G0=')

```

```

12 G0_1=inv(G0);
13 disp(G0_1, 'Inverse of G0=')
14 X0=G0_1*A0T*B0;
15 disp(X0, 'X0=')
16
17 //by recursive least square algorithm
18 G1=G0+A1'*A1;
19 disp(G1, 'G1=');
20 G1_1=inv(G1);
21 disp(G1_1, 'Inverse of G1')
22 X1=X0+G1_1*A1'*(B1-A1*X0);
23 disp(X1, 'X1=')
24
25 //verification
26 A=[3,0;0,3;3,3;6,3];
27 B=[2;2;2;6];
28 AT=A';
29 G=AT*A;
30 disp(G, 'G=')
31 G_1=inv(G);
32 disp(G_1, 'Inverse of G=')
33 X=G_1*AT*B;
34 disp(X, 'X=')
35 disp('Thus X and X1 are Same')

```

Chapter 10

Numerical Solutions of System of Non Linear Equations

Scilab code Exa 10.1 System of Non Linear Equations

```
1 //Example 10.1
2 //System of Non Linear Equations
3 //Page no. 311
4 clc;clear;close;
5
6 deff( 'y=f(x) ', 'y=x^2-exp(2*x)-4 ')
7 deff( 'y=f1(x) ', 'y=2*x-2*exp(2*x) ')
8 x0=0;e=0.00001
9 for i=1:10
10     x1=x0-f(x0)/f1(x0)
11     e1=abs(x0-x1)
12     x0=x1;
13     if abs(x0)<e then
14         break;
15     end
16 end
17 printf('\n\nThe solution of this equation after %i
```

Iterations by newton raphshon method is %.10f', i, x1)

Scilab code Exa 10.2 Contraction Method and Seidel Method

```

1 //Example 10.2
2 //Contraction Method and Seidel Method
3 //Page no. 315
4 clc;clear;close;
5 x(1)=0;y(1)=0
6 printf('(a) Contraction Mapping\n\n\n\txn\t\tnyn\n',x
      _____
      (1),y(1))
7 for i=2:9
8     x(i)=sin(x(i-1)+y(i-1))
9     y(i)=cos(x(i-1)-y(i-1))
10    printf(' %i\t%f\t%f\n',i-1,x(i),y(i))
11 end
12 printf('\n\n\n(b) Seidel Method\n\n\n\txn\t\tnyn\n',x
      _____
      (1),y(1))
13 for i=2:9
14     x(i)=sin(x(i-1)+y(i-1))
15     y(i)=cos(x(i)-y(i-1))
16     printf(' %i\t%f\t%f\n',i-1,x(i),y(i))
17 end

```

Scilab code Exa 10.3 Non Linear System of Equation

Scilab code Exa 10.4 Newton Method

```
1 //Example 10.4
2 //Newton Method
3 //Page no. 317
4 clc; clear; close;
5
6 deff( 'y=f1 (x1 ,x2 )' , 'y=x1+3*log10 (x1 )-x2 ^2 ')
7 deff( 'y=f2 (x1 ,x2 )' , 'y=2*x1 ^2-x1*x2 -5*x1+1 ')
8 deff( 'y=f11 (x1 ,x2 )' , 'y=1+3/(log (10 )*x1 )')
9 deff( 'y=f12 (x1 ,x2 )' , 'y=-2*x2 ')
10 deff( 'y=f21 (x1 ,x2 )' , 'y=4*x1-x2 -5 ')
11 deff( 'y=f22 (x1 ,x2 )' , 'y=-x1 ')
12 x=[3.4 ;2.2 ];
13 disp(x , 'x(0) = ')
14 for i=1:3
15     fx=[f1(x(1),x(2));f2(x(1),x(2))]
16     printf( '\n fx(%i) = \n ',i)
17     disp(fx)
18     A=[f11(x(1),x(2)),f12(x(1),x(2));f21(x(1),x(2)),
19         f22(x(1),x(2)),]
20     disp(A , 'A = ')
21     A_1=inv(A)
22     disp(A_1 , 'Inverse of A = ')
23     x=x-A_1*fx
24     printf( '\n x(%i) = \n ',i)
25     disp(x)
26 end
```

Scilab code Exa 10.5 Newton Raphshon Method

```
1 //Example 10.5
2 //Newton Raphshon Method
```

```

3 //Page no. 320
4 clc;clear;close;
5
6 deff( 'y=f1 (x,y)', 'y=x^3-3*x*y^2+1' )
7 deff( 'y=f2 (x,y)', 'y=3*x^2*y-y^3' )
8 deff( 'y=f11 (x,y)', 'y=3*x^2-6*y^2' )
9 deff( 'y=f12 (x,y)', 'y=-6*x*y' )
10 deff( 'y=f21 (x,y)', 'y=6*x*y' )
11 deff( 'y=f22 (x,y)', 'y=3*x^2-3*y^2' )
12 x=[0;1];
13 printf( '\nx(0) = %g\ny(0) = %g\n',x(1),x(2))
14 for i=1:3
15     fx=[f1(x(1),x(2));f2(x(1),x(2))]
16     printf( '\n fx(%i) = \n',i)
17     disp(fx)
18     J=[f11(x(1),x(2)),f12(x(1),x(2));f21(x(1),x(2)),
19         f22(x(1),x(2)),]
20     disp(J,'J = ')
21     d=det(J);
22     if d==0 then
23         dx1=0;dx2=0;
24     else
25         dx1=(fx(1)*J(2,2)-fx(2)*J(1,2))/d;
26         dx2=(fx(2)*J(1,1)-fx(1)*J(2,1))/d;
27     end
28     x(1)=x(1)+dx1;
29     x(2)=x(2)+dx2;
30     printf( '\nx(%i) = %g\ny(%i) = %g\n',i,x(1),i,x
(2))
31 end

```

Scilab code Exa 10.6 Newton Method

```

1 //Example 10.6
2 //Newton Method
3 //Page no. 322
4 clc; clear; close;
5
6 deff( 'y=f1 (x,y,z) ', 'y=x-0.1*y^2+0.05*z^2-0.7 ')
7 deff( 'y=f2 (x,y,z) ', 'y=y+0.3*x^2-0.1*x*z-0.5 ')
8 deff( 'y=f3 (x,y,z) ', 'y=z+0.4*y^2+0.1*x*y-1.2 ')
9 deff( 'y=f11(x,y,z) ', 'y=1 ')
10 deff( 'y=f12(x,y,z) ', 'y=-0.2*y ')
11 deff( 'y=f13(x,y,z) ', 'y=0.1*z ')
12 deff( 'y=f21(x,y,z) ', 'y=0.6*x-0.1*z ')
13 deff( 'y=f22(x,y,z) ', 'y=1 ')
14 deff( 'y=f23(x,y,z) ', 'y=-0.1*x ')
15 deff( 'y=f31(x,y,z) ', 'y=0.1*y ')
16 deff( 'y=f32(x,y,z) ', 'y=0.8*y+0.1*x ')
17 deff( 'y=f33(x,y,z) ', 'y=1 ')
18 x=[0;0;0];
19 printf('n\txn\t\tn\tn\tn\n
          _____\
          n')
20 for i=1:6
21     fx=[f1(x(1),x(2),x(3));f2(x(1),x(2),x(3));f3(x
              (1),x(2),x(3))]
22     J=[f11(x(1),x(2),x(3)),f12(x(1),x(2),x(3)),f13(x
              (1),x(2),x(3));f21(x(1),x(2),x(3)),f22(x(1),x
              (2),x(3)),f23(x(1),x(2),x(3));f31(x(1),x(2),x
              (3)),f32(x(1),x(2),x(3)),f33(x(1),x(2),x(3)) ]
23     J_1=inv(J)
24     printf(' %i\t%f\t%f\t%f\n',i-1,x(1),x(2),x(3))
25     x=x-J_1*fx
26 end
27 printf('\n\nThe solution is x = %f, y = %f and z =
          %f',x(1),x(2),x(3))
28
29 printf('\n\nNote : There are computation errors in
          calculation given by the book')

```

Scilab code Exa 10.7 Iterative Method

```
1 //Example 10.7
2 //Iterative Method
3 //Page no. 326
4 clc;clear;close;
5
6 x=[0;0;0];
7 printf('n\txn\t\tyn\t\tzn\n
-----\
n')
8 for i=1:7
9     printf(' %i\t%.10f\t%.10f\t%.10f\n',i-1,x(1),x
(2),x(3))
10    x(1)=0.7+0.1*x(2)^2-0.05*x(3)^2
11    x(2)=0.5-0.3*x(1)^2+0.1*x(1)*x(3)
12    x(3)=1.2-0.4*x(2)^2-0.1*x(1)*x(2)
13 end
14 printf('\n\nThe solution is x = %.10f , y = %.10f and
z = %.10f ',x(1),x(2),x(3))
```

Scilab code Exa 10.8 Steepest Descent

```
1 //Example 10.8
2 //Steepest Descent
3 //Page no. 328
4 clc;clear;close;
5
```

```

6  def f( 'y=f( x1 ,x2 ) ' , 'y=(x1 -2)^4+3*(x2 +3)^2 ')
7  x=[1 ; -2];
8  printf( 'n\ t      x1\ t\ t      x2\ t\ t      F(x1 ,x2 )\ n
   _____\
   n ')
9  for i=1:11
10    Fx=[f(x(1) ,x(2)) ];
11    J=[4*(x(1) -2)^3 ,6*(x(2)+3) ];
12    u=(Fx *J*J' *Fx) /(J*J'*Fx *J*J' *Fx)
13    printf( ' %i\ t%.10 f\ t%.10 f\ t%.10 f\ n ' ,i-1 ,x(1) ,x
        (2) ,Fx)
14    x=x-u*J'*Fx
15 end
16 printf( '\n\nThis shows that the solution tends to x1
= %i      and      x2 = %i ' ,ceil(x(1)) ,floor(x(2)))

```

Chapter 11

Eigenvalues and Eigenvectors

Scilab code Exa 11.1 Eigenvalues and Eigenvectors

```
1 //Example 11.1
2 //Eigenvalues and Eigenvectors
3 //Page no. 333
4 clc;clear;close;
5
6 A1=[0.6;0.2];A2=[-0.2;0.6];A3=[-0.6;-0.2];A4
    =[0.2;-0.6];
7 T=[1.1,-0.3;-0.3,1.9];
8 B1=T*A1;B2=T*A2;B3=T*A3;B4=T*A4;
9 disp(B4,B3,B2,B1,'The transformed vectors are :')
10 disp('These points lie on the ellipse:')
11 printf('      2      2\n(x-3y)+(3x+y)\n-----\n'
    n   16      4\n\n')
12 A5=[0;2/sqrt(10)];
13 disp('The vector (0,2/10^(1/2)) lies on the circle:'
    )
14 printf(' 2      2\nx + y = 4\n          -\n          10\n\n'
    ')
15 B5=T*A5;
```

```
16 disp('Also lies on the same ellipse',B5)
17 printf('\n\nWe can see that there is a linear
       relationship between the first 4 vectors and
       their respective transformend vectors through the
       scalars known as eigenvectors and eigenvalues
       respectively')
```

Scilab code Exa 11.2 Leverrier's Method

```
1 //Example 11.2
2 //Leverrier's Method
3 //Page no. 337
4 clc;close;clear;
5
6 A=[2,2,2;2,5,5;2,5,1];
7 A1=A;
8 C(1)=0;
9   for j=1:3
10     for k=1:3
11       if(j==k)
12         C(1)=C(1)+A1(j,k)
13       end
14     end
15   end
16 disp(A,'A=')
17   disp(A1,'A1=')
18 printf('\nC1=')
19 disp(C(1));
20 for i=2:3
21   A2=A*(A1-C(i-1)*eye(3,3));
22   printf('\n\n\nA%i=',i)
23   disp(A2);
24   C(i)=0;
```

```

25      for j=1:3
26          for k=1:3
27              if(j==k)
28                  C(i)=C(i)+A2(j,k)/i
29              end
30          end
31      end
32      printf ('\nC%i=%',i)
33      disp(C(i))
34      A1=A2;
35  end
36  printf ('\n\n\nTherefore the characteristic
            polynomial is:\n 3      2\ nx - %ix - %ix %i = 0 ',C
            (1),C(2),C(3))
37
38 // verification
39 printf ('\n\nVerification:')
40 s=poly(0,"s");
41 p=poly(A,'x');
42 A=A-eye(3,3)*%s;
43 disp(p,'=',A)

```

Scilab code Exa 11.3 Danilevsky Method

```

1 //Example 11.3
2 //Danilevsky Method
3 //Page no. 341
4 clc;close;clear;
5
6 A=[-1,0,0;1,-2,3;0,2,-3];
7 G=[A;eye(3,3)];
8 disp(G);
9 //transformation to frobenius matrix

```

```

10  for k=3:-1:2
11      g(k)=0;
12      for j=1:k-1
13          if(g(k)<G(k,j))
14              g(k)=G(k,j)
15              p=j;
16          end
17      end
18      if(g(k) ~=0)
19          for j=1:3
20              r(1,j)=G(k,j)
21          end
22          for i=1:6
23              G(i,k-1)=G(i,k-1)/g(k)
24          end
25          disp(G)
26          for j=1:3
27              if(j ~=k-1)
28                  l=G(k,j)
29                  for i=1:6
30                      G(i,j)=G(i,j)-l*G(i,k-1)
31                  end
32              end
33          end
34          disp(G)
35      end
36      for j=1:3
37          for i=1:3
38              c(i,1)=G(i,j)
39          end
40          G(k-1,j)=0
41          for i=1:3
42              G(k-1,j)=G(k-1,j)+r(1,i)*c(i,1)
43          end
44      end
45      disp(G)
46  end
47

```

```

48 // partition g
49 for i=4:6
50     for j=1:3
51         T(i-3,j)=G(i,j)
52     end
53 end
54 disp(T, 'T=')
55
56 // eigenvalues computation
57 printf ('\n\n\nCharacteristic polynomial: ')
58 p=poly(A, 'x')
59 disp(p)
60 printf ('\n\n\nEigenvalues: ')
61 a=roots(p)
62 disp(a)
63 // eigenvectors computation
64 for k=1:3
65     m=2
66     for l=1:3
67         y(l,k)=a(k,1)^(m)
68         m=m-1;
69     end
70 end
71 printf ('\n\n')
72 disp(y, 'y=')
73
74 // eigenvector computation
75
76 for k=1:3
77     for l=1:3
78         y1(l,1)=y(l,1)
79         y2(l,1)=y(l,2)
80         y3(l,1)=y(l,3)
81     end
82     x1=T*y3;
83     x2=T*y2;
84     x3=T*y1;
85 end

```

```

86 printf ('\n\nEigenvectors :\n')
87 for i=1:3
88     printf ('|%.1f|\t|%.1f|\t|%.1f| ',x1(i,1),x2(i
89     ,1),x3(i,1))
90     printf ('\n')
91 end

```

Scilab code Exa 11.4 Power Method

```

1 //Example 11.4
2 //Power Method
3 //Page no. 345
4 clc;close;clear;
5
6 A=[1,2;3,4];
7 e=0.001;
8 q0=[1;1];
9 for i=1:5
10    q1=A*q0;
11    a=max(q1)
12    for j=1:2
13        q2(j)=q1(j)/a;
14    end
15    printf ('nq(%i) = %.4f      a = %.4f      Scaled
16          q(%i) = %.4f\n      %.4f
17          %i\n', 
18          i,q1(1),a,i,q2(1),q1(2),q2(2))
16    q1=q2;
17    q0=q1;
18 end
19 printf ('Hence the largest eigenvalue is %.4f with
20          the corresponding eigenvector as %.4f\n'

```

```
%i ',a,q0(1),q0(2))
```

Scilab code Exa 11.5 Inverse Power Method

```
1 //Example 11.5
2 //Inverse Power Method
3 //Page no. 347
4 clc;close;clear;
5
6 A=[7,6,-3;-12,-20,24;-6,-12,16];
7 e=10^-6;
8 X=[1;1;1];
9 B=0;
10 Y=[0;0;0]
11 a=0;l=0;
12 for i=1:2
13     printf('When a=%i\n',a);
14     C=A-a*eye();
15     disp(C,"C=");
16     C_1=inv(C);
17     disp(C_1,"Inverse of C=");
18     printf('\n\nItr          lambda
19         printf('\n
20
21         for j=1:10
22             printf(' \n%10.5f      %10.5f      %10.5f
23             %10.5f',j-1,l,X(1),X(2),X(3));
24             Y=C_1*X;
25             B=max(Y);
26             e1=abs(l-B);
27             X=Y/B;
```

```

26      m=0;
27      for k=1:3
28          m=m+(Y(k)-X(k))^2;
29      end
30      e2=sqrt(m);
31      er=max(e1,e2);
32      if(er<e)
33          break
34      end
35      l=B;
36
37  end
38  a=-3;
39  printf('
40 end
41 printf('
Note : Computation of Y is wrong given
in the book')

```

Scilab code Exa 11.6 Rayleigh Quotient

```

1 //Example 11.6
2 //Rayleigh Quotient
3 //Page no. 348
4 clc;close;clear;
5
6 A=[10,7,8,7;7,5,6,5;8,6,10,9;7,5,9,10];
7 q0=[1;1;1;1];
8 for i=0:4
9     X=(A^i)*q0;
10    l=(X'*A*X)/(X'*X)
11    printf('
Lambda(%i) = %f
',i+1,l)
12 end
13 printf('
Dominant Eigenvalue = %f
',l)

```

```

14
15 e=0.001;
16 for i=1:5
17     q1=A*q0;
18     a=max(q1)
19     for j=1:4
20         q2(j)=q1(j)/a;
21     end
22
23     q1=q2;
24     q0=q1;
25 end
26 disp(q2, 'Corresponding Eigenvector = ')

```

Scilab code Exa 11.7 Jacobi Method

```

1 //Example 11.7
2 //Jacobi 's Method
3 //Page no. 355
4 clc;close;clear;
5
6 A=[1 ,1 ,1/2;1 ,1 ,1/4;1/2 ,1/4 ,2];
7 C=A;
8 V=[sqrt(2) ,0 ,1/2;sqrt(2) ,0 ,1/4;3/(4*sqrt(2)) ,-1/(4*
    sqrt(2)) ,2]
9 S=eye(3 ,3)
10 disp(A,"A =" )
11 VI=0;
12 for i=1:3
13     for j=1:3
14         if(i ~= j)
15             VI=VI+A(i,j)^2
                //initial off diag norm

```

```

16         end
17     end
18 end
19 VI=sqrt(VI);
20 VF=VI*10^-7;           // final threshold
21 V1=VI/3;
22 o=poly(0,"o");
23 for i=1:3
24 for q=2:3
25     for p=q-1:-1:1
26         if(A(p,q)>V1)
27             a=-A(p,q);
28             b=(A(p,p)-A(q,q))/2
29             if(b^=0)
30                 w=b*abs(1/b)*(a/sqrt(a^2+b^2));
31             else
32                 w=(a/sqrt(a^2+b^2));
33             end
34             sin0=w/sqrt(2*(1+sqrt(1-w^2)));
35             cos0=sqrt(1-sin0^2)
36         end
37         B(p,p)=A(p,p)*cos0^2+A(q,q)*sin0^2-2*A(p,q)*
38             sin0*cos0
39             B(q,q)=A(p,p)*sin0^2+A(q,q)*cos0^2+2*A(p
40             ,q)*sin0*cos0
41             B(p,q)=(A(p,p)-A(q,q))*sin0*cos0+A(p,q)
42             *(cos0^2-sin0^2)
43             S(i,i)=S(i,i)
44             S(i,p)=S(i,p)*cos0-S(i,q)*sin0
45             S(i,q)=S(i,p)*sin0+S(i,q)*cos0
46         end
47     end
48 end
49 disp(B,"B =")
50 disp(S,"S =")
51 printf('\n\n\nComputation error in the solution
      provided by book')

```

Scilab code Exa 11.8 Recursive Formula

```
1 //Example 11.8
2 //Recursive Formula
3 //Page no. 357
4 clc;close;clear;
5
6 A=[2,-1,0,0;-1,2,-1,0;0,-1,2,-1;0,0,-1,2];
7 l=poly(0,"l");
8 p0=1;
9 p1=A(1,1)-1;
10 for i=2:4
11     p2=(A(i,i)-1)*p1-A(i,i-1)^2*p0;
12     p0=p1;
13     p1=p2;
14     printf ('\n\nnp%i(l) = ',i);
15     disp(p2)
16 end
```

Scilab code Exa 11.9 QR Method

```
1 //Example 11.9
2 //QR Method
3 //Page no. 360
4 clc;close;clear;
5
6 A=[2,-1,0;-1,2,-1;0,-1,2];
```

```

7  def c('y=c(i,j)', 'y=A(j,j)/sqrt((A(i,j)^2+A(j,j)^2))'
8  def s2('y=s2(i,j)', 'y=A(i,j)/sqrt((A(i,j)^2+A(j,j)^2))'
9  disp(A, 'A=')
10 l0=0; f=1; m=0; s=0; w=0;
11 for n=1:5
12     for j=1:2
13         for k=1:2
14             V(j,k)=A(j,k)
15         end
16     end
17     disp(V, 'V=')
18     p=poly(V, 'x');
19     disp('=0', p);
20     a=roots(p);
21     for j=1:2
22         printf('\na(%i) = %f', j, a(j))
23     end
24     if (abs(a(1)-V(1,1))<=abs(a(2)-V(1,1)))
25         a=a(1)
26     else
27         a=a(2)
28     end
29     printf('\n a = %f\n', a)
30     s=s+a;
31     A=A-a*eye()
32     R=A; Q=eye(3,3);
33
34 for j=1:2
35     for i=j+1:3
36         for k=1:3
37             for l=1:3
38                 if(k==l)
39                     if(k==i | k==j)
40                         C(k,l)=c(i,j)
41                 else

```

```

42          C(k,1)=1
43      end
44  end
45  if(k>1)
46      if(k==i & l==j)
47          C(k,1)=-1*s2(i,j)
48      else
49          C(k,1)=0
50      end
51  end
52  if(k<1)
53      if(k==j & l==i)
54          C(k,1)=s2(i,j)
55      else
56          C(k,1)=0
57      end
58  end
59 end
60
61
62 R=C*R;
63 Q=Q*C';
64
65 end
66 end
67 disp(Q,'Q='),R,'R=')
68 disp(Q*R,'Q*R=')
69 A=R*Q;
70 disp(A,'A=')
71 end
72 l1=10+s;
73 for i=2:3
74     for j=2:3
75         V(i-1,j-1)=A(i,j)
76     end
77 end
78 disp(V,'V=')
79 p=poly(V,'x');

```

```

80      disp('=0',p);
81      a=roots(p);
82      for j=1:2
83          printf('\na(%i) = %f',j,a(j))
84      end
85      l2=l1+a(1)
86      l3=l1+a(2)
87      disp(l3,'l3=',l2,'l2=',l1,'l1=')
88 printf('\n\n\nNote : Values of V varies in each step
           resulting in different results due to error in
           book calculation')

```

Scilab code Exa 11.10 LU Method

```

1 //Example 11.10
2 //LU Method
3 //Page no. 363
4 clc;close;clear;
5
6 A
    =[120,80,40,-16;80,120,16,-40;40,16,120,-80;-16,-40,-80,120];
7 disp(A,"A =")
8 L=eye(4,4);
9 for l=1:20
10 for j=1:4
11     for i=1:j
12         k=0
13         for p=1:i-1
14             k=k-A(i,p)*A(p,j)
15         end
16         A(i,j)=A(i,j)+k
17     end

```

```

18     for i=j+1:4
19         k=0;
20         for p=1:j-1
21             k=k-A(i,p)*A(p,j)
22         end
23         A(i,j)=(A(i,j)+k)/A(j,j)
24     end
25 end
26 disp(A,"Modified A = ")
27 for i=1:4
28 for j=1:4
29 if i>j then
30     L(i,j)=A(i,j)
31 else
32     U(i,j)=A(i,j)
33 end
34 end
35 end
36 disp(U,"U =" ,L,"L =" )
37 A=U*L;
38 printf ('\n\nAfter %i iterations, matrix A =\n\n',1)
39 for i=1:4
40     for j=1:4
41         printf ('    %.2f\t',A(i,j))
42     end
43     printf ('\n')
44 end
45 end
46 printf ('\n\nTherefore the eigenvalues are the
        diagonal elements f the transformed triangular
        matrix are:\n\n')
47 for i=1:4
48     printf (' %.2f , ',A(i,i))
49 end

```

Scilab code Exa 11.11 Generalized Eigenvalue Problem

```
1 //Example 11.11
2 //Generalized Eigenvalue Problem
3 //Page no. 365
4 clc;close;clear;
5
6 A=[1 ,1 ,0.5;1 ,1 ,0.25;0.5 ,0.25 ,2]
7 B=[2 ,2 ,2;2 ,5 ,5;2 ,5 ,11]
8 disp(B,"B =",A,"A =")
9 for i=1:3
10    G(i,i)=sqrt(B(i,i))
11 end
12 G=[B;eye(3,3)];
13
14 //transformation to frobenius matrix
15 for k=3:-1:2
16    g(k)=0;
17    for j=1:k-1
18        if(g(k)<G(k,j))
19            g(k)=G(k,j)
20            p=j;
21        end
22    end
23    if(g(k) ~=0)
24        for j=1:3
25            r(1,j)=G(k,j)
26        end
27        for i=1:6
28            G(i,k-1)=G(i,k-1)/g(k)
29        end
30        for j=1:3
```

```

31         if (j ~= k-1)
32             l=G(k,j)
33             for i=1:6
34                 G(i,j)=G(i,j)-l*G(i,k-1)
35             end
36         end
37     end
38 end
39 for j=1:3
40     for i=1:3
41         c(i,1)=G(i,j)
42     end
43     G(k-1,j)=0
44     for i=1:3
45         G(k-1,j)=G(k-1,j)+r(1,i)*c(i,1)
46     end
47 end
48 end
49
50 // partition g
51 for i=4:6
52     for j=1:3
53         T(i-3,j)=G(i,j)
54     end
55 end
56
57 // eigenvalues computation
58 p=poly(B,'x')
59 a=roots(p)
60 printf('\n\nDiagonalized Matrix B = \n\n')
61 for i=1:3
62     for j=1:3
63         if i ~= j then
64             B(i,j)=0
65         else
66             B(i,j)=a(i)
67         end
68     end

```

```

69 end
70 disp(B)
71 //eigenvectors computation
72 for k=1:3
73     m=2
74     for l=1:3
75         y(l,k)=a(k)^(m)
76         m=m-1;
77     end
78 end
79 printf('\n\n')
80
81
82 for k=1:3
83     for l=1:3
84         y1(l,1)=y(l,1)
85         y2(l,1)=y(l,2)
86         y3(l,1)=y(l,3)
87     end
88     x1=T*y3;
89     x2=T*y2;
90     x3=T*y1;
91 end
92 printf('\n\nEigenvectors of B are :\n\n')
93 for i=1:3
94     printf('%.5f|\t%.5f|\t%.5f|\n',x3(i,1),x2(i
95     ,1),x1(i,1))
96     printf('\n')
97 end
98 x=[x3,x2,x1]
99
100
101
102
103 B=[2,2,2;2,5,5;2,5,11]
104 G=0
105 for i=1:3

```

```
106     for j=1:3
107         if i==j then
108             G(i,j)=sqrt(B(i,j))
109         else
110             G(i,j)=0;
111         end
112     end
113 end
114
115 B=inv(G)*x'*A*x*inv(G)
116 disp(B,"Eigenvectors of A =")
117
118 printf('\n\n\nNote : Computation Error in book in
    caculation of eigenvector of B thus for A')
```

Chapter 12

Interpolation and Extrapolation

Scilab code Exa 12.1 Linear Interpolation Technique

```
1 //Example 12.1
2 //Linear Interpolation Technique
3 //Page no. 372
4 clc;close;clear;
5
6 printf('x:      ')
7 f=[1,4,9,16,25];
8 for i=1:5
9     printf('%i\t',i)
10 end
11 printf('\n f(x):    ')
12 for i=1:5
13     printf('%i\t',f(i))
14 end
15 x=2.5;
16 x1=2;x2=3;printf ('\n\n for (2,4) and (3,9)')
17 f(2.5)=f(x1)+(f(x2)-f(x1))*(x-x1)/(x2-x1)
```

```

18 printf( '\n f(2.5) = %.1f ', f(2.5))
19
20 x=2.5;
21 x1=2;x2=4;printf( '\n\n for (2,4) and (4,16) ')
22 f(2.5)=f(x1)+(f(x2)-f(x1))*(x-x1)/(x2-x1)
23 printf( '\n f(2.5) = %.1f ', f(2.5))
24
25 x=2.5;
26 x1=1;x2=3;printf( '\n\n for (1,1) and (3,9) ')
27 f(2.5)=f(x1)+(f(x2)-f(x1))*(x-x1)/(x2-x1)
28 printf( '\n f(2.5) = %.1f ', f(2.5))
29
30 printf( '\n\n Exact value = %.2f ', 2.5^2)

```

Scilab code Exa 12.2 Lagarangian Method

```

1 //Example 12.2
2 //Lagarangian Method
3 //Page no. 373
4 clc;close;clear;
5
6 xk=[-1,0,2,5];
7 yk=[10,7,7,22];
8
9 P=0;
10 x=poly(0,"x");
11 for k=0:3
12     p=yk(k+1)
13     for j=0:3
14         if(j~=k)
15             p=p*((x-xk(j+1))/(xk(k+1)-xk(j+1)))
16         end
17     end

```

```
18     P=P+p;
19 end
20 disp(P, 'P=')
```

Scilab code Exa 12.3 Aitken Nevilles Method

```
1 //Example 12.3
2 //Aitken–Neville 's Method
3 //Page no. 378
4 clc;close;clear;
5
6 function [x,y,z]=tran(a,b)           // function for
   exchanging values
7 z=a;y=b;x=z;
8 endfunction
9 deff( 'y=P(a,b,c,d,e)', 'y=(c(d)*b(d+1)-c(d+e)*b(d))/(a(d+e)-a(d))' ) //function for finding
   polynomials
10 xi=[0.8,1,1.2,1.4,1.6];
11 yi=[2.2255,2.7183,3.3201,4.0552,4.9530];
12 x=1.23
13 [xi(5),xi(1),a]=tran(xi(1),xi(5))
14 [xi(4),xi(1),a]=tran(xi(1),xi(4))
15 [xi(3),xi(2),a]=tran(xi(2),xi(3))
16 [xi(2),xi(1),a]=tran(xi(1),xi(2))
17 [yi(5),yi(1),a]=tran(yi(1),yi(5))
18 [yi(4),yi(1),a]=tran(yi(1),yi(4))
19 [yi(3),yi(2),a]=tran(yi(2),yi(3))
20 [yi(2),yi(1),a]=tran(yi(1),yi(2))
21 for i=1:5
22     x_xi(i)=x-xi(i);
23 end
24 printf('xi      x-xi      yi\n')
```

```

25 printf('-----\n')
26 for i=1:5
27     printf('%.1f      %.2f\t%f\n',xi(i),x_xi(i),yi(i))
28 end
29 printf('\n\nPolynomials\n')
30 printf('-----\n')
31 for i=1:4
32     for j=1:5-i
33         printf('%f\n',P(xi,yi,x_xi,j,i))
34         yi(j)=P(xi,yi,x_xi,j,i)
35     end
36     printf('\n\n\n')
37 end

```

Scilab code Exa 12.4 Newtons Divided Difference Interpolation

```

1 //Example 12.4
2 //Newton's Divided Difference Interpolation
3 //Page no. 381
4 clc;close;clear;
5
6 x=[0,1,2,3,4,5]
7 y=[1,2,5,10,17,26];
8 y1=y;
9 def('yi=P(a,b,d,e)', 'yi=(b(d+1)-b(d))/(a(d+e)-a(d))'
    ') //function for finding polynomials
10 for i=1:3
11     for j=1:6-i
12         z(j,i)=P(x,y,j,i)
13         y(j)=z(j,i)
14     end
15 end

```

```

16 z(6,1)=0;
17 printf('x      y      f(x0,x1)      f(x0,x1,x3)      f
           (x0,x1,x2,x3)\n')
18 printf(
19     n')
20     for j=1:6
21         printf(' %i      %i \t%i\t\t%i\t\t%i\t\t%i\n',x(1,j)
22             ,y1(1,j),z(j,1),z(j,2),z(j,3))
23     end
24 x1=2.6;
25 f=y1(4)+(x1-x(4))*(z(4,1))+(x1-x(4))*(x1-x(5))*z
26     (4,2)
27 printf('\n\nf(2.6)=%.2f',f)

```

Scilab code Exa 12.5 Interpolation Methods

```
1 //Example 12.5
2 //Interpolation Methods
3 //Page no. 403
4 clc; close; clear;
5
6 x=[0,1,2,3,4];
7 y=[0,1,8,27,64];
8
9 //Inverse lagrange Method
10 P=0;
11 y1=20;
12 for k=0:4
13     p=x(k+1)
14     for j=0:4
15         if(j ~= k)
16             p=p*((y1-y(j+1))/(y(k+1)-y(j+1)))
```

```

17         end
18     end
19     P=P+p;
20 end
21 disp(P, 'Inverse Lagrange interpolation x=')
22
23
24 //Newton's divide difference interpolation
25 x1=x;
26 def('xi=P(a,b,d,y)', 'xi=(b(d+1)-b(d))/(a(d+y)-a(d))')
    //function for finding polynomials
27 for i=1:2
28     for j=1:5-i
29         z(j,i)=P(y,x,j,i)
30         x(j)=z(j,i)
31     end
32 end
33 z(5,1)=0;
34 printf('\n\n y\tx          f(y0 ,y1)          f(y0 ,y1 ,y3 )\n
')
35 printf('-----\n
')
36     for j=1:5
37         printf(' %i\t%i \t%i\t%i\t%i\t\n',y(1,j),x1(1,
            j),z(j,1),z(j,2))
38     end
39 y1=20;
40 f=x1(4)+(y1-y(4))*(z(4,1))+(y1-y(4))*(y1-y(5))*z
    (4,2)
41 printf('\n\nNewton Divide Difference x(20)=% .2f ',f)
42
43 x=x1;
44 //Iterated Linear Interpolation
45 function [x,y,z]=tran(a,b)           // function for
    exchanging values
46     z=a;y=b;x=z;
47 endfunction
48 def('y=P(a,b,c,d,e)', 'y=(c(d)*b(d+1)-c(d+e)*b(d))/(d*(d+1))')

```

```

        a(d+e)-a(d))') // function for finding
        polynomials
49 y1=20
50
51 [y(4),y(1),a]=tran(y(1),y(4))
52 [y(3),y(2),a]=tran(y(2),y(3))
53 [x(4),x(1),a]=tran(x(1),x(4))
54 [x(3),x(2),a]=tran(x(2),x(3))
55 for i=1:5
56     y1_y(i)=y1-y(i);
57 end
58 printf('y\ty1-y\tx\n')
59 printf('-----\n')
60 for i=1:5
61     printf('%.1f\t%i\t%i\n',y(i),y1_y(i),x(i))
62 end
63 printf('\n\nPolynomials\n')
64 printf('-----\n')
65 for i=1:4
66     for j=1:5-i
67         printf('%f\n',P(y,x,y1_y,j,i))
68         x(j)=P(y,x,y1_y,j,i)
69     end
70     printf('\n\n')
71 end
72 printf('Iterated Linear Interpolation x(20) = %f',x(
    j))
73
74 x=[0,1,2,3,4];
75 y=[0,1,8,27,64];
76 y1=y;
77 //Suggested Interpolation
78
79 for i=1:4
80     for j=1:5-i
81         z(j,i)=y(j+1)-y(j);
82         y(j)=z(j,i)
83     end

```

```

84 end
85 printf( '\n\n\n x\ty\tdy\td2y\td3y\td4y\n' )
86 printf( ,
87 for i=1:5
88     printf( '%i\t%i\t%i\t%i\t%i\n' ,x(i),y1(i),z(
89         i,1),z(i,2),z(i,3),z(i,4))
90 end
91 s=poly(0,'s')
92 p=y1(4);k=3;
93 for i=1:3
94     r=1;
95     for j=1:i
96         r=r*(s+(j-1))
97     end
98     r=r*z(k,i)/factorial(j);
99     k=k-1;
100    p=p+r;
101    printf( '\n\nStage %i : ',i)
102    disp(p)
103 end
104 s0=-7/19;
105 disp(s0,'s0=');
106 s1=(-7-s0*(s0+1)*6)/19
107 disp(s1,'s1=')
108 s2=(-7-s1*(s1+1)*6-s1*(s1+1)*(s1+2))/19
109 disp(s2,'s2=')
110 x2=3+s2;
111 disp(x2,'Suggested Interpolation x(20)=');

```

Scilab code Exa 12.6 Chebyshev Interpolating Polynomial

```

1 //Example 12.6
2 //Chebyshev Interpolating Polynomial
3 //Page no. 407
4 clc; close; clear;
5
6 deff( 'y=f(x)', 'y=1/(1+exp(-x))');
7 a=-2; b=2; n=3;
8 D=%pi/(2*n+2)
9 for k=0:n
10     t(k+1)=-cos(D*(2*k+1))
11     x(k+1)=((a+b)/2)+(b-a)*t(k+1)/2
12     y(k+1)=f(x(k+1))
13     C(k+1)=0
14 end
15 for j=0:n
16     for k=0:n
17         L=(2*k+1)*D
18         C(j+1)=C(j+1)+y(k+1)*cos(j*L)
19     end
20 end
21 C(1)=C(1)/(n+1);
22 for j=1:n
23     C(j+1)=2*C(j+1)/(n+1)
24 end
25
26 x=poly(0,"x")
27 T(1)=1; T(2)=x;
28 for j=1:n-1
29     T(j+2)=2*x*T(j+1)-T(j)
30 end
31 P=C(1)*T(1)
32 for j=1:n
33     P=P+C(j+1)*T(j+1)
34 end
35 disp(P, 'P3(x)=')
36 printf('\n\n\nNote : Book has Calculation errors in
calculation of coefficients')

```

Scilab code Exa 12.7 Double Interpolation

```
1 //Example 12.7
2 //Double Interpolation
3 //Page no. 409
4 clc;close;clear;
5
6 x=[0,1,2,3,4];
7 y=[0,1,2,3,4];
8 z
    =[0,1,8,27,64;1,3,11,31,69;4,7,16,37,76;9,13,23,45,85;16,21,32,55

9 printf('y / x')
10 for i=1:5
11     printf('\t%i',x(i))
12 end
13 for i=1:5
14     printf('\n %i',y(i))
15     for j=1:5
16         printf('\t%i',z(j,i))
17     end
18 end
19 printf('\n\n\n')
20 for i=1:5
21     x=2.5;
22     x1=2;x2=3;
23     z1(1,i)=z(i,x1+1)+(z(i,x2+1)-z(i,x1+1))*(x-x1)/(
24         x2-x1)
25 end
26 printf('Values of z at x=2.5:\n\n y')
27 for i=1:5
28     printf('\t%i',y(i))
```

```

28 end
29 printf( '\n z ')
30 for i=1:5
31     printf( '\t%g ',z1(i))
32 end
33 y=1.5;
34 y1=1; y2=2;
35 z2=z1(y1+1)+(z1(y2+1)-z1(y1+1))*(y-y1)/(y2-y1)
36 printf( '\n\nValue of z at x=2.5 and y=1.5 : %g',z2)

```

Scilab code Exa 12.8 Spline Interpolation

```

1 //Example 12.8
2 //Spline Interpolation
3 //Page no. 414
4 clc;close;clear;
5
6 xi=[0.10,0.11,0.12,0.13,0.14,0.15,0.16,0.17];
7 yi
    =[0.1110,0.1234,0.1361,0.1491,0.1623,0.1759,0.1897,0.2038];

8 h=0.01;
9
10 pi(1)=0;qi(1)=0;pi(8)=0;qi(8)=0;
11 for i=2:7
12     pi(i)=-1/(4*pi(i-1))
13     qi(i)=((6/h^2)*(yi(i+1)-2*yi(i)+yi(i-1))-qi(i-1))
        /(4*pi(i-1))
14 end
15 si2(8)=0;
16 si2(1)=0;si1(8)=0;
17 si1(1)=0;
18 for i=7:-1:2

```

```

19     si2(i)=pi(i)*si2(i+1)+qi(i)
20 end
21 for i=2:8
22     si1(i)=si1(i-1)+h*(si2(i)+si2(i-1))/2
23 end
24 printf( '\n i\t xi\t fi\t pi\t\t qi\t\t si2\t\t
25 si1')
26 printf( '\n')
27 for i=1:8
28     printf( '\n %i\t%g\t%g\t%f\t%f\t%f\t%f' ,i ,xi(i) ,
29 yi(i) ,pi(i) ,qi(i) ,si2(i) ,si1(i))
30 end
31 x=0.1325;
32 i=4;
33 s=yi(i)+(x-xi(i))*si1(i)+(si2(i)*(x-xi(i))^2)/2+((si2(i+1)-si2(i))/(xi(i+1)-xi(i)))*((x-xi(i))^3)/6
34 printf( '\n\nSpline Interpolated Value of s(0.1325)
35 is : %f' ,s)

```

Chapter 13

Numerical Differentiation

Scilab code Exa 13.1 Differentiation

```
1 //Example 13.1
2 //Differentiation
3 //Page no. 420
4 clc;close;clear;
5
6 deff( 'y=f(x) ', 'y=x^2+5 ')
7 deff( 'y=f1(x,h) ', 'y=(f(x+h)-f(x))/h ')
8 h=0.01;x=2.4
9 d=f1(x,h)
10 d1=(f1(x+h,h)-f1(x))/h
11 printf('dy\n --- = %g\n dx',d)
12 printf('\n\n\n d2y\n --- = %g\n dx2 ',d1)
```

Scilab code Exa 13.2 Calculation of x coordinate of Minimum Point

```

1 //Example 13.2
2 //Calculation of x-coordinate of Minimum Point
3 //Page no. 422
4 clc;close;clear;
5
6 for i=1:7
7     for j=1:6
8         z(i,j)=0
9     end
10 end
11 h=0.2
12 printf('x          y          d          d2
13 printf(
_____
')
14 for i=1:7
15     z(i,1)=i/5;
16 end
17 z(1,2)=2.10022
18 z(2,2)=1.98730
19 z(3,2)=1.90940
20 z(4,2)=1.86672
21 z(5,2)=1.85937
22 z(6,2)=1.88755
23 z(7,2)=1.95147
24 for i=3:6
25     for j=1:9-i
26         z(j,i)=z(j+1,i-1)-z(j,i-1)
27     end
28 end
29 disp(z)
30
31 s=poly(0,'s')
32 p=z(5,2);k=4;
33 for i=3:5
34     r=1;
35     for j=1:i-2

```

```

36 r=r*(s+(j-1))
37 end
38 r=r*z(k,i)/factorial(j);
39 k=k-1;
40 p=p+r;
41
42 end
43 disp(p)
44 s=(-z(4,3)+z(3,4)/2)/z(3,4)
45 disp(s,'s=')
46 x=z(5,1)+s*h
47 disp(x,'x=')

```

Scilab code Exa 13.3 Newton Forward Difference Formula

```

1 //Example 13.3
2 //Newton's Forward Difference Formula
3 //Page no. 423
4 clc;close;clear;
5 printf(' x\t\t y\t\t d\t\t d2\t\t d3\t
\ t\t\t d4\ n')
6 printf(
_____
')
7 h=0.05;
8 z
=[1.00,1.00000;1.05,1.02470;1.10,1.04881;1.15,1.07238;1.20,1.0954
9 deff('y=f1(x,s)', 'y=(z(x,3)+(s-1/2)*z(x,4)+z(x,5)
*(3*s^2-6*s+2)/6)/h')
10 deff('y=f2(x,s)', 'y=(z(x,4)+z(x,5)*(s-1))/h^2')
11 deff('y=f3(x,s)', 'y=z(x,5)/h^3')
12 for i=3:6

```

```

13      for j=1:9-i
14          z(j,i)=z(j+1,i-1)-z(j,i-1)
15      end
16  end
17 printf ('\n')
18 for i=1:7
19     for j=1:6
20         if z(i,j)==0 then
21             printf (' \t')
22         else
23             printf ('%.7f\t',z(i,j))
24         end
25     end
26     printf ('\n')
27 end
28 s=poly(0,'s')
29 p=z(5,2);k=4;
30 for i=3:5
31     r=1;
32     for j=1:i-2
33         r=r*(s+(j-1))
34     end
35     r=r*z(k,i)/factorial(j);
36     k=k-1;
37     p=p+r;
38
39 end
40 disp(p,'y(s) = ')
41 printf ('\n\ny1(1) = %g',f1(1,0))
42 printf ('\n\ny2(1) = %g',f2(1,0))
43 printf ('\n\ny3(1) = %g',f3(1,0))
44 printf ('\n\ny1(1.025) = %g',f1(1,0.5))

```

Scilab code Exa 13.4 Newton Backward Difference Formula

```
1 //Example 13.4
2 //Newton's Backward Difference Formula
3 //Page no. 425
4 clc;close;clear;
5 printf(' x\t\t y\t\t d\t\t d2\t\t d3\t
6 \t\t d4\n')
6 printf(
7 )
7 h=0.02;
8 z
9 = [0.96 ,1.8025;0.98 ,1.7939;1.00 ,1.7851;1.02 ,1.7763;1.04 ,1.7673];
10
11 def(f 'y=f1(x,s)', 'y=(z(x,3)+(s+1/2)*z(x,4))/h')
12 for i=3:6
13     for j=1:7-i
14         z(j,i)=z(j+1,i-1)-z(j,i-1)
15     end
16     printf('\n')
17 for i=1:5
18     for j=1:6
19         if z(i,j)==0 then
20             printf(' \t')
21         else
22             printf('%.7f\t',z(i,j))
23         end
24     printf('\n')
25 end
26 printf('\n\ny1(1) = %g',f1(2,0))
27 printf('\n\ny1(1.03) = %g',f1(4,0.5))
```

Scilab code Exa 13.5 Stirlings Central Difference Derivatives

```
1 //Example 13.5
2 //Stirlings Central Difference Derivatives
3 //Page no. 426
4 clc;close;clear;
5 printf(' x\t\t y\t\t d\t\t d2\t\t d3\n
      ')
6 printf('
      ')
7 h=0.01;s=0.5;
8 deff('y=f1(x,s)', 'y=((z(x,3)+z(x-1,3))/2+s*z(x-1,4)
      +(z(x-1,5)+z(x-2,5))*(3*s^2-1)/12)/h')
9 deff('y=f2(x,s)', 'y=(z(x-1,4))/h^2')
10 deff('y=f3(x,s)', 'y=(z(x-1,5)+z(x-2,5))/(2*h^3)')
11 z
      =[1.00,1.00000;1.01,1.00499;1.02,1.00995;1.03,1.01489;1.04,1.0198
12 for i=3:5
13   for j=1:19-i
14     z(j,i)=z(j+1,i-1)-z(j,i-1)
15   end
16 end
17 printf('\n')
18 for i=1:17
19   for j=1:5
20     if z(i,j)==0 then
21       printf(' \t')
22     else
23       printf('%.7f\t',z(i,j))
24     end
```

```
25      end
26      printf( '\n' )
27 end
28 printf( '\n\ny1(1.125) = %g ( exact value =
    0.4771404 ) ,f1(13,0.5))
29 printf( '\n\ny2(1.125) = %g ( exact value =
    -0.20951 ) ,f2(13,0.5))
30 printf( '\n\ny3(1.125) = %g ( exact value = 0.27935)
    ,f3(13,0.5))
```

Scilab code Exa 13.6 Extrapolation

```
1 //Example 13.6
2 //Extrapolation
3 //Page no. 430
4 clc;close;clear;
5 x=[-0.8,-0.6,-0.4,-0.2,0,0.2,0.4,0.6,0.8];
6 y
    =[0.2019,0.30119,0.44933,0.67032,1,1.49182,2.22554,3.32012,4.95303
7 for i=1:4
8     printf( '\nh = %g\n' ,x(10-i))
9     y1=(y(10-i)-y(i))/(2*x(10-i))
10    printf( 'f1(0) = %g\n\n' ,y1)
11 end
```

Scilab code Exa 13.7 Richardson Extrapolation

```
1 //Example 13.7
```

Scilab code Exa 13.8 Application

```
1 //Example 13.8
2 //Application
3 //Page no. 433
4 clc;close;clear;
5
6 def( 'y=f(x)', 'y=2/x^2' )
7 a=1;b=2;a1=1;b1=0;
```

```

8 N=4;
9 h=(b-a)/(N+1);
10 for j=1:N
11     s(j)=f(a+j*h)
12 end
13 for i=1:N
14     for j=1:N
15         if abs(i-j)==1 then
16             A(i,j)=-1
17         end
18         if i==j then
19             A(i,j)=2+s(i)*h^2
20         end
21     end
22     if i==1 then
23         k(i,1)=s(i)+a1/h^2
24     elseif i==N
25         k(i,1)=s(i)+b1/h^2
26     else
27         k(i,1)=s(i)
28     end
29 end
30 disp(A, 'A = ')
31 disp(k, 'k = ')

```

Chapter 14

Numerical Integration

Scilab code Exa 14.2 Simpsons 1 3rd Rule

```
1 //Example 14.2
2 //Simpsons 1/3rd Rule
3 //Page no 442
4 clc;clear;close;
5 x(1,1)=0
6 for i=2:9
7
8     x(1,i)=x(1,1)+(i-1)*10
9 end
10 y
= [30 ,31.63 ,33.44 ,35.47 ,37.75 ,40.33 ,43.29 ,46.69 ,50.67]

11
12 // trapezoidal rule
13 S=0;
14 h=(x(9)-x(1))/8
15 for j=1:9
16     S=0;
17     for i=1:j
```

```

18     if (i==1 | i==j)
19         S=S+y(i)
20     else
21         S=S+2*y(i)
22     end
23 end
24 S=S*h/2
25 printf ('\n Velocity at t (%i) = %.2f ',x(j),S)
26 y1(j)=S
27 end
28
29 y1(1)=0;
30 //Simpsons 1/3rd Rule
31 S=0;
32 h=(x(9)-x(1))/8
33 for i=1:9
34     if (i==1 | i==9)
35         S=S+y1(i)
36     elseif (((i)/2)-fix((i)/2)==0)
37         S=S+4*y1(i)
38     else
39         S=S+2*y1(i)
40     end
41 end
42 S=S*h/3;
43 S=S/1000
44 printf ('\n\nSimpsons 1/3rd Rule Sum = %g km ',S)

```

Scilab code Exa 14.3 Trapezoidal Rule and Simpsons Rule

```

1 //Example 14.3
2 //Trapezoidal Rule and Simpsons Rule
3 //Page no. 442

```

```

4 clc; close; clear;
5 n=2; a=0; b=1;
6 h=(b-a)/n
7 def(f,'y=f(x)', 'y=1/(1+x)')
8 for i=0:2
9     x(i+1)=i/2;
10    y(i+1)=f(x(i+1))
11 end
12 printf('xi\t')
13 for i=1:3
14     printf('%g\t',x(i))
15 end
16 printf('\n yi\t')
17 for i=1:3
18     printf('1/%g\t',1+(i-1)/2)
19 end
20
21 // trapezoidal rule
22 S=0;
23 for i=1:3
24     if(i==1 | i==3)
25         S=S+y(i)
26     else
27         S=S+2*y(i)
28     end
29 end
30 S=S*h/2
31 printf('\n\nTrapezoidal Rule Sum = %g',S)
32
33 //Simpsons 1/3rd Rule
34 S=0;
35 for i=1:3
36     if(i==1 | i==3)
37         S=S+y(i)
38     elseif(((i)/2)-fix((i)/2)==0)
39         S=S+4*y(i)
40     else
41         S=S+2*y(i)

```

```

42     end
43 end
44 S=S*h/3
45 printf( '\n\nSimpsons 1/3rd Rule Sum = %g' ,S)

```

Scilab code Exa 14.5 Romberg Method

```

1 //Example 14.5
2 //Romberg Method
3 //Page no. 457
4 clc;close;clear;
5
6 deff( 'y=f( x ) ' , 'y=1/(1+x) ' )
7
8 h=[0.5 ,0.25 ,0.125]
9 for k=1:3
10    for i=0:h(k):1
11       x(i/h(k)+1)=i;
12       y(i/h(k)+1)=f(x(i/h(k)+1))
13    end
14    n=1+(1/h(k))
15    //trapezoidal rule
16    S=0;
17    for i=1:n
18       if(i==1 | i==n)
19          S=S+y(i)
20       else
21          S=S+2*y(i)
22       end
23    end
24    S=S*h(k)/2
25    printf( '\n\nI(%g) = %g' ,h(k) ,S)
26    z(2*k-1 ,1)=S

```

```

27 end
28 for i=2:3
29     for k=1:4-i
30         z(k*2+i-2,i)=z(2*k-1+i,i-1)+(z(2*k-1+i,i-1)-z(2*
31             k-3+i,i-1))/3
32     end
33 end
34 printf ('\n\n')
35 disp(z,'The Table of values :')

```

Scilab code Exa 14.7 Gaussian Quadrature Formula

```

1 //Example 14.7
2 //Gaussian Quadrature Formula
3 //Page no. 463
4 clc;close;clear;
5
6 deff('y=f(x)', 'y=cos(x)*log(x)')
7 s=0;
8 for i=0:2:2000
9     s=s+integrate('((-1)^(i/2))*(x^i)/factorial(i)*
10         log(x)', 'x', 0, 1)
11 end
12 disp(s,'Till 1000 terms .... I =')

```

Scilab code Exa 14.8 Gauss Legendre Two Point Rule

```
1 //Example 14.8
```

```
2 //Gauss Legendre Two Point Rule
3 //Page no. 472
4 clc;close;clear;
5
6 def(f,'y=f(x)', 'y=1/(x+3)')
7 s=integrate('f(x)', 'x', -1, 1)
8 printf('By Direct Method, I = %g', s)
9 s=f(-1/sqrt(3))+f(1/sqrt(3))
10 printf('\n\n By Gauss-Legendre 2 point rule, I = %g',
, s)
```

Scilab code Exa 14.9 Gauss Legendre Three Point Rule

```
1 //Example 14.9
2 //Gauss Legendre Three Point Rule
3 //Page no. 473
4 clc;close;clear;
5
6 def(f,'y=f(x)', 'y=1/(x+3)')
7 s=integrate('f(x)', 'x', -1, 1)
8 printf('By Direct Method, I = %g', s)
9 s=5/9*f(-sqrt(3/5))+8/9*f(0)+5/9*f(sqrt(3/5))
10 printf('\n\n By Gauss-Legendre 3 point rule, I = %g',
, s)
```

Scilab code Exa 14.10 Spline Integration Method

```
1 //Example 14.10
2 //Spline Integration Method
```

```

3 //Page no. 478
4 clc;close;clear;
5
6 deff( 'y=f(x)', 'y=sind(%pi*x)')
7 deff( 'y=f1(x,h)', 'y=(f(x+h)-f(x))/h')
8 h=0.01;
9 n=2;h=0.5;a=0;b=1;
10 disp(integrate('f(x)', 'x', 0, 1), 'I = ')

```

Scilab code Exa 14.11 Trapezoidal Rule

```

1 //Example 14.1
2 //Trapezoidal Rule
3 //Page no 440
4 clc;clear;close;
5 x1=1.46
6 for i=1:6
7     x(1,i)=x1+i/100
8 end
9 y=[3.86,3.90,3.96,4.02,4.06,4.12]
10
11 // trapezoidal rule
12 S=0;
13 h=(x(6)-x1)/6
14 for i=1:6
15     if(i==1 | i==6)
16         S=S+y(i)
17     else
18         S=S+2*y(i)
19     end
20 end
21 S=S*h/2
22 printf('\n I = %g',S)

```

Scilab code Exa 14.14 Trapezoidal and Simpsons Rule

```
1 //Example 14.14
2 //Trapezoidal and Simpsons Rule
3 //Page no. 486
4 clc;close;clear;
5
6 x(1)=0.5;y(1)=0.5;h=0.25
7 for i=2:3
8     x(i)=x(i-1)+h
9     y(i)=y(i-1)+h
10 end
11 printf(' y/x\|t | \t%g\|t%g\|t%g' ,x(1),x(2),x(3))
12 printf('\n-----|-----')
13 for i=1:3
14     printf('\n%g\|t |\|t' ,y(i))
15     for j=1:3
16         z(i,j)=x(j)*y(i)
17         printf('%g\|t' ,z(i,j))
18     end
19 end
20
21 // trapezoidal rule
22 s=0;
23 for i=1:3
24     for j=1:3
25         if i==1 & j==1 then
26             s=s+z(i,j)
27         elseif i==3 & j==3
28             s=s+z(i,j)
29         else
30             s=s+2*z(i,j)
```

```

31         end
32     end
33 end
34 s=(s*(h^2))/4
35 printf ('\n\n')
36 disp(s, 'Trapezoidal Rule Sum = ')
37 printf ('\n\n')
38 //simpsons rule
39 s=0;
40 for i=1:3
41     for j=1:3
42         if i/2-int(i/2)==0 & j/2-int(j/2)==0 then
43             s=s+16*z(i,j)
44         elseif i/2-int(i/2)~=0 & j/2-int(j/2)~=0
45             s=s+z(i,j)
46         else
47             s=s+4*z(i,j)
48         end
49     end
50 end
51 s=(s*(h^2))/9
52 disp(s, 'Simpsons Rule Sum = ')

```

Scilab code Exa 14.15 Trapezoidal and Simpsons Rule

```

1 //Example 14.15
2 //Trapezoidal and Simpsons Rule
3 //Page no. 487
4 clc;close;clear;
5
6 x(1)=0;y(1)=0;h=0.25
7 for i=2:5
8     x(i)=x(i-1)+h

```

```

9      y(i)=y(i-1)+h
10     end
11     printf( ' y/x\t|\t%g\t%g\t%g\t%g\t%g' ,x(1),x(2),x(3),
12       x(4),x(5))
12     printf( '\n
13     for i=1:5
14       printf( '\n%g\t|',y(i))
15       for j=1:5
16         z(i,j)=x(j)*y(i)
17         printf( '%g\t',z(i,j))
18       end
19     end
20
21 // trapezoidal rule
22 s=0;
23 for i=1:5
24   for j=1:5
25     if i==1 & j==1 then
26       s=s+z(i,j)
27     elseif i==5 & j==5
28       s=s+z(i,j)
29     else
30       s=s+2*z(i,j)
31     end
32   end
33 end
34 s=(s*(h^2))/4
35 printf( '\n\n')
36 disp(s,'Trapezoidal Rule Sum = ')
37 printf( '\n\n')
38
39 // simpsons rule
40 s=0;
41 for i=1:5
42   for j=1:5
43     if i/2-int(i/2)==0 & j/2-int(j/2)==0 then

```

```

44         if i==j then
45             s=s+16*z(i,j)
46         else
47             s=s+4*z(i,j)
48         end
49
50         elseif i/2-int(i/2) ~=0 & j/2-int(j/2) ~=0
51             s=s+z(i,j)
52         else
53             s=s+4*z(i,j)
54         end
55     end
56 end
57 s=(s*(h^2))/9
58 disp(s,'Simpsons Rule Sum = ')

```

Scilab code Exa 14.16 Multiple Integration with Variable Limits

```

1 //Example 14.16
2 //Multiple Integration with Variable Limits
3 //Page no. 491
4 clc;close;clear;
5
6 deff( 'z=f(x)', 'z=x+1')
7 deff( 'z=f1(y)', 'z=(y+1)^3*(y+3)^2')
8 s=5/9*f(-sqrt(3/5))+8/9*f(0)+5/9*f(sqrt(3/5))
9 s=s*5/9*f1(-sqrt(3/5))+8/9*f1(0)+5/9*f1(sqrt(3/5))
10 s=s/256;
11 disp(s,'I = ')

```

Scilab code Exa 14.18 Integration

```
1 //Example 14.18
2 //Integration
3 //Page no. 494
4 clc;close;clear;
5
6 s=integrate('x^2*sin(x^2)', 'x', 0, 1)
7 disp(s, 'I = ')
```

Scilab code Exa 14.19 Integration

```
1 //Example 14.19
2 //Integration
3 //Page no. 494
4 clc;close;clear;
5
6 s=integrate('sin(t)/t', 't', 1, 999)
7 disp(s, 'I = ')
```

Chapter 15

Numerical Solutions of Ordinary Differential Equations Initial Value Problem

Scilab code Exa 15.1 Ordinary Differential Equation

```
1 //Example 15.1
2 //Ordinary Differential Equation
3 //Page no. 503
4 clc;clear;close;
5 s=log(2)/log(1.02)
6 disp(s, 'Time Taken = ')
```

Scilab code Exa 15.6 Taylor Method

```
1 //Example 15.6
```

```

2 //Taylor Method
3 //Page no. 510
4 clc;clear;close;
5
6 def(f 'y=f1 (x ,y) ', 'y=x^2+y^2 ')
7 def(f 'y=f2 (x ,y) ', 'y=2*x+2*y*f1 (x ,y) ')
8 def(f 'y=f3 (x ,y) ', 'y=2+2*f1 (x ,y)^2+2*y*f2 (x ,y) ')
9 def(f 'y=f4 (x ,y) ', 'y=6*f1 (x ,y)*f2 (x ,y)+2*y*f3 (x ,y) ')
10 h=0.2;
11 for l=1:2
12     a=0;y=0;x=0;
13     printf( '\n-----\nh = %g\n'
14             -----'\n' ,h)
14     for i=1:4
15         x=a+(i-1)*h
16         k=0;
17         for j=1:4
18             if j==1 then
19                 k=k+(h^j)*f1(x ,y)/factorial(j)
20             elseif j==2
21                 k=k+(h^j)*f2(x ,y)/factorial(j)
22             elseif j==3
23                 k=k+(h^j)*f3(x ,y)/factorial(j)
24             else
25                 k=k+(h^j)*f4(x ,y)/factorial(j)
26             end
27         end
28         y=y+k;
29         printf( '\nx = %g\ny(%g) = %g\n' ,x ,x+0.2 ,y)
30     end
31     h=h+0.2;
32 end

```

Scilab code Exa 15.7 Picard Method

```
1 //Example 15.7
2 //Picard Method
3 //Page no. 511
4 clc;clear;close;
5 deff( 'y=f(x,y) ', 'y=x^2+y^2 ')
6 y(1)=0;
7 for i=1:2
8     y(i+1)=y(1)+integrate('f(x,y(i)) ','x',0,i/10)
9     printf('\n y(%g) = %g\n',i/10,y(i+1))
10 end
```

Scilab code Exa 15.8 Euler Method

```
1 //Example 15.8
2 //Euler Method
3 //Page no. 513
4 clc;clear;close;
5 deff( 'y=f(x,y) ', 'y=x+y ')
6 y(1)=1;
7 h=0.1;
8 for i=1:6
9     printf ('\ny(%g) = %g\n',(i-1)/10,y(i))
10    y(i+1)=y(i)+h*f((i-1)/10,y(i))
11
12 end
```

Scilab code Exa 15.9 Trapezium Method

```

1 //Example 15.9
2 //Trapezium Method
3 //Page no. 516
4 clc;clear;close;
5 deff( 'y=f(x,y)', 'y=x*y^2')
6 y=1;
7 h=0.2;
8 y2=poly(0, 'y2')
9 for i=1:2
10     x=(i-1)*h;
11     x1=x+h
12     y1=roots(-y2+y+h*(f(x,y)+f(x1,y2))/2)
13     printf ('\n Y(%i) = %g or %g\n', i, y1(1), y1(2))
14 end

```

Scilab code Exa 15.10 Heun Method

```

1 //Example 15.10
2 //Heun Method
3 //Page no. 517
4 clc;clear;close;
5 deff( 'y=f(x,y)', 'y=y*2/x')
6 y=2;
7 h=0.25;
8 for i=1:4
9     x=1+(i-1)*h
10    x1=x+h
11    ye=y+h*f(x,y)
12    y=y+h*(f(x,y)+f(x1, ye))/2
13    printf ('\n y(%g) = %g\n', x1, y)
14 end

```

Scilab code Exa 15.11 Midpoint Method

```
1 //Example 15.11
2 //Midpoint Method
3 //Page no. 518
4 clc;clear;close;
5 deff( 'y=f(x,y)' , 'y=y+x' )
6 y=1;
7 h=0.2;
8 printf('i\txi\tyi\tslope1\tslope2\ty(i+1)\n
          ')

---


9 for i=1:3
10     x=(i-1)*h
11     s1=f(x,y);
12     s2=f(x+h/2,y+s1*h/2);
13     printf(' %i\t%g\t%g\t%g\t%g',i-1,x,y,s1,s2)
14     y=y+s2*h;
15     printf('\t%g\n',y)
16 end
```

Scilab code Exa 15.12 Modified Midpoint Method

```
1 //Example 15.12
2 //Modified Midpoint Method
3 //Page no. 519
4 clc;clear;close;
5 deff( 'y=f(x,y)' , 'y=y+x' )
```

```

6 y=1;
7 h=0.2;
8 Z(1)=y;
9 Z(2)=Z(1)+h*f(0,Z(1));
10 printf('Z(%i) = %g',1,Z(2));
11 for i=2:5
12     x=(i-1)*h;
13     Y(i-1)=(Z(i)+Z(i-1)+h*f(x,Z(i)))/2;
14     Z(i+1)=Z(i-1)+2*h*f(x,Z(i));
15     printf('\n Y(%i) = %g\n\n Z(%i) = %g',i-1,Y(i-1),i,Z(i+1));
16 end
17 printf('\n\n\n y4 = %g', (4*Y(4)-Y(2))/3)

```

Scilab code Exa 15.13 Single Step Method

```

1 //Example 15.13
2 //Single Step Method
3 //Page no. 521
4 clc;clear;close;
5
6 def(f,'y=f(x)', 'y=x^2')
7 def(f1,'y=f1(x)', 'y=1/(1-x)')
8 y=1;h=0.2;
9 printf('n\tXn\tYn (by single-step method)\tYn (computed)n
n')
10 for i=1:6
11     x=(i-1)*h
12     if i<6 then
13         printf(' %i\t%.2f\t%.5f\t%.5f\n',i-1,x,y,f1(x))

```

Scilab code Exa 15.14 Second Order Runge Kutta Method

```

1 //Example 15.14
2 //Second Order Runge Kutta Method
3 //Page no. 525
4 clc;clear;close;
5
6 def(f, 'y=f(x,y)', 'y=x-y')
7 y=1;x=1;h=0.1;
8 //simple runge kutta method
9 K1=h*f(x,y);
10 K2=h*f(x+h,y+K1);
11 y1=y+(K1+K2)/2
12 printf('ny(1.1) by simple runge kutta method = %g\n',y1)
13
14 //euler cauchy method
15 K1=h*f(x,y);
16 K2=h*f(x+h/2,y+K1/2);
17 y1=y+(K1+K2)
18 printf('y(1.1) by euler cauchy method = %g\n',y1)
19
20 //optimal method
21 K1=h*f(x,y);
22 K2=h*f(x+2*h/3,y+2*K1/3);
23 y1=y+(K1+3*K2)/4
24 printf('y(1.1) by optimal method = %g',y1)

```

Scilab code Exa 15.15 Third Order Runge Kutta Method

```
1 //Example 15.15
2 //Third Order Runge Kutta Method
3 //Page no. 526
4 clc;clear;close;
5 def(f, 'y=f(x,y)', 'y=x-y')
6 y=1;x=1;h=0.1;
7 //scheme 1
8 K1=h*f(x,y);
9 K2=h*f(x+h/2,y+K1/2);
10 K3=h*f(x+h/2,y-K1+2*K2);
11 y1=y+(K1+4*K2+K3)/6
12 printf('ny(1.1) by scheme 1 = %g\n\n',y1)
13
14 //scheme 2
15 K1=h*f(x,y);
16 K2=h*f(x+h/3,y+K1/3);
17 K3=h*f(x+2*h/3,y+2*K2/3);
18 y1=y+(K1+3*K3)/4
19 printf('ny(1.1) by scheme 2 = %.7f\n\n',y1)
```

Scilab code Exa 15.16 Fourth Order Runge Kutta Method

```
1 //Example 15.16
2 //Fourth Order Runge Kutta Method
3 //Page no. 528
```

```

4 clc;clear;close;
5 deff( 'y=f(x,y)', 'y=x-y' )
6 y=1;x=1;h=0.1;
7 K1=h*f(x,y);
8 K2=h*f(x+h/2,y+K1/2);
9 K3=h*f(x+h/2,y+K2/2);
10 K4=h*f(x+h,y+K3);
11 disp(K4, 'K4 = ',K3, 'K3 = ',K2, 'K2 = ',K1, 'K1 = ')
12 y1=y+(K1+2*K2+2*K3+K4)/6
13 printf( '\ny(1.1) = %.8f\n',y1)

```

Scilab code Exa 15.17 New Variant of Runge Kutta Method

```

1 //Example 15.17
2 //New Variant of Runge Kutta Method
3 //Page no. 530
4 clc;clear;close;
5 deff( 'y=f(x,y)', 'y=x-y' )
6 y=1;x=1;h=0.1;
7 K1=h*f(x,y);
8 K2=h*f(x+h/2,y+K1/2);
9 K3=h*f(x+h/2,y+K2/2);
10 K4=h*f(x+h,y+K3);
11 K5=h*f(x+3*h/4,y+(5*K1+7*K2+13*K3-K4)/32)
12 disp(K5, 'K5 = ',K4, 'K4 = ',K3, 'K3 = ',K2, 'K2 = ',K1, 'K1
    = ')
13 y1=y+(K1+2*K2+2*K3+K5)/6
14 printf( '\ny(1.1) = %.8f\n',y1)

```

Scilab code Exa 15.18 Runge Kutta Merson Method

```
1 //Example 15.18
2 //Runge Kutta Merson Method
3 //Page no. 532
4 clc;clear;close;
5 deff( 'y=f(x,y)' , 'y=x+y' )
6 y=1;x=0;h=0.1;
7 printf( '\n\t Xn\t Yn\t K1\t K2\t K3\t K4\t K5\tY(n+1)
          \n
          ')
8 for i=0:14
9     K1=h*f(x,y);
10    K2=h*f(x+h/3,y+K1/3);
11    K3=h*f(x+h/3,y+(K1+K2)/6);
12    K4=h*f(x+h/2,y+(K1+3*K3)/8);
13    K5=h*f(x+h,y+(K1-3*K3+4*K4)/2);
14    y1=y+(K1+4*K4+K5)/6
15    printf( '\n %i\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t',i,x,y,K1,K2,K3,K4,K5,y1)
16    y=y1;
17    x=x+h;
18 end
```

Scilab code Exa 15.19 Runge Kutta Fehlberg Method

```
1 //Example 15.19
2 //Runge Kutta Fehlberg Method
3 //Page no. 535
4 clc;clear;close;
5 deff( 'y=f(x,y)' , 'y=x-y' )
6 y=1;x=1;h=0.1;
```

```

7 K1=h*f(x,y);
8 K2=h*f(x+h/4,y+K1/4);
9 K3=h*f(x+3*h/8,y+3*(K1+3*K2)/32);
10 K4=h*f(x+12*h/13,y+1932*K1/2197-7200*K2/2197+7296*K3
    /2197);
11 K5=h*f(x+h,y+439*K1/216-8*K2+3680*K3/513-845*K4
    /4104)
12 K6=h*f(x+h/2,y-8*K1/27+2*K2-3544*K3/2565+1859*K4
    /4104-11*K5/40)
13 disp(K6,'K6 = ',K5,'K5 = ',K4,'K4 = ',K3,'K3 = ',K2,'K2
    = ',K1,'K1 = ')
14 y1=y+(25*K1/216+1408*K3/2565+2197*K4/4104-K5/5)
15 y11=y+(16*K1/135+6656*K3/12825+28561*K4/56430-9*K5
    /50+2*K6/55)
16 printf('\ny(1.1) = %.9f\n',y1)
17 printf('\ny~(1.1) = %.9f\n',y11)

```

Scilab code Exa 15.20 Carp Karp Runge Kutta Method

```

1 //Example 15.20
2 //Carp Karp Runge Kutta Method
3 //Page no. 537
4 clc;clear;close;
5 deff('y=f(x,y)', 'y=x-y')
6 y=1;x=1;h=0.1;printf('\n')
7 U=[0,1/5,3/10,3/5,1,7/8];
8 v
    =[0,0,0,0,0;1/5,0,0,0,0;3/40,9/40,0,0,0;3/10,-9/10,6/5,0,0;-11/54
    0,0,0,0,0];
9 a=[37/378,0,250/621,125/594,0,512/1771];
10 a1
    =[2825/27648,0,18575/48384,13525/55296,277/14336,1/4];

```

```

11 for l=1:5
12     K(1)=h*f(x,y);
13 for i=2:6
14     k=0;
15     for j=1:i-1
16         k=k+v(i,j)*K(j)
17     end
18     K(i)=h*f(x+U(i)*h,y+k)
19 end
20 k=0;
21 for i=1:6
22     k=k+a(i)*K(i)
23 end
24 y1=y+k;
25 k=0;
26 for i=1:6
27     k=k+a1(i)*K(i)
28 end
29 y11=y+k;
30 for i=1:6
31     printf('K%i = %.9f\n',i,K(i))
32 end
33 printf('\ny(1.1) = Y%i = %.9f\n',1,y1)
34 printf('y~(1.1) = Y~%i = %.9f\n',1,y11)
35 y=y1;
36 printf('\n\n')
37 end

```

Scilab code Exa 15.21 Implicit Runge Kutta Method

```

1 //Example 15.21
2 //Implicit Runge Kutta Method
3 //Page no. 539

```

```

4 clc;clear;close;
5 deff( 'y=f(x,y)', 'y=x-y' )
6 y=1;x=1;h=0.1;printf( '\n')
7 U=[0,1/5];
8 v=[0,0;1/2,1/2];
9 a2=1;
10 K(1)=h*f(x,y);
11 K(2)=(x+h/2-y-K(1)/2)/(1/h-1/2)
12 y1=y+(K(1)+a2*K(2))
13 printf( '\ny(1.1) = %.9f\n',y1)

```

Scilab code Exa 15.22 Linear Multi Step Method

```

1 //Example 15.22
2 //Linear Multi Step Method
3 //Page no. 540
4 clc;clear;close;
5 deff( 'y=f(x,y)', 'y=x+y' )
6 y(1)=1;y(2)=1;x(1)=0;h=0.1;
7 printf( '\n\tXn\tYn\tfn\n'
          '0\tg\t.3 f\t.3 f\n',x(1),y(1),f(x(1),y(1)))
          ;
8 for i=2:11
9     x(i)=(i-1)*h;
10    y(i+1)=(-y(i)-y(i-1)+h*(f(x(i),y(i))+f(x(i-1),y(
           i-1)))/2;
11    printf( '%i\t.3 f\t.3 f\n',i-1,x(i),y(
           i),f(x(i),y(i)))
12 end

```

Scilab code Exa 15.23 Milne Simpson Predictor Corrector Method

```
1 //Example 15.23
2 //Milne Simpson Predictor Corrector Method
3 //Page no. 544
4 clc;clear;close;
5 deff( 'y=f(x,y)' , 'y=y+exp(x)')
6 h=0.5;
7 y=[1,1.824,3.718,7.722]
8 for i=1:4
9     x=(i-1)*h;
10    f1(i)=f(x,y(i));
11    printf ('\n\nPredictor = %.9f\n\n',y(i))
12 end
13 y41=y(1)+4*h*(2*f1(4)-f1(3)+2*f1(2))/3
14 f4=f(x+h,y41);
15 y4=y(3)+h*(f4+4*f1(4)+f1(3))/3
16 printf ('\n\nEvaluator = %.9f\n\n',f4)
17 printf ('Corrector = %.9f\n',y4)
```

Scilab code Exa 15.24 Improved Milne Simpson Predictor Corrector Method

```
1 //Example 15.24
2 //Improved Milne Simpson Predictor Corrector Method
3 //Page no. 546
4 clc;clear;close;
5
```


Scilab code Exa 15.25 Hamming Predictor Corrector Method

```
1 //Example 15.25
2 //Hamming Predictor Corrector Method
3 //Page no. 548
4 clc; clear; close;
5
6 def f( 'y=f(x,y)', 'y=y-x^2' )
7 y(1)=1; h=0.25; x=0;
8 printf( '\n\tXn\tYn\tfn\tYc(n)\tYc(n+1)\tm(n+1)\tv( n+1)\tYc(n+1)\n'
9 n' )
10 f1(1)=f(x,y(1));
11 for i=1:3
12     K1=h*f(x,y(i));
13     K2=h*f(x+2*h/3,y(i)+2*K1/3);
14     y(i+1)=y(i)+(K1+3*K2)/4
15     printf(' %i\t%.3f\t%.3f\t%.3f\n',i-1,x,y(i),f1(i))
16     x=x+h
17     f1(i+1)=f(x,y(i+1))
18 end
19 Y31=y(4); Yc=0
20 for i=3:10
21     Y41=y(i-2)+4*h*(2*f1(4)-f1(3)+2*f1(2))/3    // predictor
22     m4=Y41+112*(Y31-Yc)/121      // modifier
23     v4=f(x+h,m4)                // evaluator
24     Y4c=(9*y(i+1)-y(i-1))/8+3*h*(v4+2*f1(4)-f1(3))/8
25     Y4=Y4c+9*(Y41-Y4c)/121      // final value
26     printf(' %i\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\n',
27         i,x,y(i+1),f1(4),Y31,Yc,
28         Y41,m4,v4,Y4c)
29 y(i+2)=Y4
30 Y31=Y41;
31 f1(2)=f1(3);
```

```

29     f1(3)=f1(4);
30     f1(4)=f(x+h,y(i+2))
31     Yc=Y4c
32     x=x+h
33 end

```

Scilab code Exa 15.26 Multi Valued Method

```

1 //Example 15.26
2 //Multi Valued Method
3 //Page no. 553
4 clc;clear;close;
5
6 deff('y=f1(x,y)', 'y=2*x^2-y')
7 h=0.1;x=0;y=-1;
8 deff('y=f2(x,y)', 'y=4*x-f1(x,y)')
9 deff('y=f3(x,y)', 'y=4-f2(x,y)')
10 B=[1,1,1,1;0,1,2,3;0,0,1,3;0,0,0,1];
11 y0=[y;h*f1(x,y);h^2*f2(x,y)/2;h^3*f3(x,y)/6]
12 y01=y0;
13 r=[0;1;3/4;1/6]
14
15 disp(r, 'If r = ')
16 printf('\n\n')

nx = 0\t\ttx = 0.1\t\tt\ttx = 0.2\n\t
17 for i=1:2
18     y11=B*y01
19     s(i)=h*(f1(x+h,y11(1)))-y11(2)
20     y1=y11+s(i)*r
21     if i==2 then
22         break
23     end

```

```

24      y2=y1;
25      y22=y11;
26      y01=y1
27  end
28  printf( '\t    ( s = %.5g )\t    ( s = %.9f )\n

```

n	Y0\t\t	Y' i\t\t	Y1\t\t	Y'2\t\t	Y2\n
n', s(1), s(2))					

```

29 for i=1:4
30     printf( '%.5f \t%.5f \t%.5f \t%.5f \t%.5f\n',y0(i)
            ),y22(i),y2(i),y11(i),y1(i))
31 end
32 y0=[y;h*f1(x,y);h^2*f2(x,y)/2;h^3*f3(x,y)/6]
33 y01=y0;
34 r=[5/12;1;3/4;1/6]
35 disp(r, 'If r = ')
36 printf( '\n\n

```

nx = 0\t\t\ttx = 0.1\t\t\ttx = 0.2\n\t')

```

37 for i=1:2
38     y11=B*y01
39     s(i)=h*(f1(x+h,y11(1))-y11(2))
40     y1=y11+s(i)*r
41     if i==2 then
42         break
43     end
44     y2=y1;
45     y22=y11;
46     y01=y1
47 end
48 printf( '\t    ( s = %.5g )\t    ( s = %.9f )\n

```

n	Y0\t\t	Y' i\t\t	Y1\t\t	Y'2\t\t	Y2\n
n', s(1), s(2))					

```

49 for i=1:4
50     printf( '%.5f \t%.5f \t%.5f \t%.5f \t%.5f\n',y0(i

```

```
),y22(i),y2(i),y11(i),y1(i))  
51 end
```

Scilab code Exa 15.27 First order ODE

```
1 //Example 15.27  
2 //First order ODE  
3 //Page no. 558  
4 clc;clear;close;  
5  
6 def(f1(x,y1,y2)',y=y1*y2+x')  
7 def(f2(x,y1,y2)',y=y1-x')  
8 h=0.2;x=0;y1=0;y2=1;  
9 //heun method  
10 printf('Heun Method:\n\n x\ty1\ty2\n  
11 Y=[y1;y2]  
12 for i=1:8  
13  
14 F=[f1(x,Y(1),Y(2));f2(x,Y(1),Y(2))]  
15 Y1=Y+h*F  
16 x=x+h;  
17 F1=[f1(x,Y1(1),Y1(2));f2(x,Y1(1),Y1(2))]  
18 Y=Y+(h/2)*(F+F1)  
19 printf(' %g\t%.3f\t%.3f\n',x-h,Y(1),Y(2))  
20  
21 end  
22  
23 // classical runge kutta method  
24 printf('\n\n\nClassical Runge Kutta Method:\n\n n\tx  
n\tyN\tnK1\tnK2\tnK3\tnK4\tnY(n+1)\n  
n')
```

```

25 Y=[y1;y2];x=0;
26 for i=1:6
27     K1=h*[f1(x,Y(1),Y(2));f2(x,Y(1),Y(2))]
28     K2=h*[f1(x+h/2,Y(1)+K1(1)/2,Y(2)+K1(2)/2);f2(x+h
29         /2,Y(1)+K1(1)/2,Y(2)+K1(2)/2)]
30     K3=h*[f1(x+h/2,Y(1)+K2(1)/2,Y(2)+K2(2)/2);f2(x+h
31         /2,Y(1)+K2(1)/2,Y(2)+K2(2)/2)]
32     K4=h*[f1(x+h,Y(1)+K3(1),Y(2)+K3(2));f2(x+h,Y(1) +
33         K3(1),Y(2)+K3(2))]
34     Y1=Y+(K1+2*K2+2*K3+K4)/6
35     printf(' %i\t%.2f\t%.3f\t%.3f\t%.3f\t%.3f\t%
36         %.3f\n\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%
37         \n
38         n',i-1,x,Y(1),K1(1),K2(1),K3(1),K4(1),Y1(1),Y
39         (2),K1(2),K2(2),K3(2),K4(2),Y1(2))
40
41 Y=Y1;
42 x=x+h
43 end

```

Scilab code Exa 15.28 Differential Equation

```

1 //Example 15.28
2 //Differential Equation
3 //Page no. 562
4 clc;clear;close;
5
6 deff('y=f(x,y)', 'y=2*y^2/(1+x)')
7 h=0.1;z(1)=-1;
8 for i=1:11
9     printf('\nZ(%g) = %g\n',(i-1)/10,z(i))
10    z(i+1)=z(i)+h*f((i-1)/10,z(i))
11 end

```


Chapter 16

Numerical Solutions of Ordinary Differential Equations Boundary Value Problems

Scilab code Exa 16.1 Outline of Linear Shooting Method

```
1 //Example 16.1
2 //Outline of Linear Shooting Method
3 //Page no. 572
4 clc;close;clear;
5
6 def(f ,y=f(x) ,y=x^2);
7 h=0.5;X0=0;Y0=1;Z1=[-1,-1.5,-1.1771];i=1;Y1=Y0;
8 for j=1:3
9   Z0=Z1(i);
10  i=i+1
11  Y0=1;
12  for n=1:2
13    printf('nFor n = %i\n'
14      _____\n
15      K1(1)=h*Z0;
```

```

15      printf( '\n K11 = %g',K1(1));
16      K1(2)=h*f(Y0);
17      printf( '\n K12 = %g',K1(2));
18      K2=h*f(Y0+K1(2))
19      printf( '\n K22 = %g',K2);
20      Z0=Z0+(K1(2)+K2)/2
21      printf( '\n Z%i = %g',n,Z0);
22      K2=h*Z0;
23      printf( '\n K21 = %g',K2);
24      Y0=Y0+(K1(1)+K2)/2
25      printf( '\n Y%i = %g',n,Y0);
26      printf( '\n\n\n')
27      if n==1 then
28          Y2=Y0
29      end
30  end
31  printf( '\n\n\n')
32 end
33 printf('Hence the solution is y(%g) = %i, y(%g) = %
.4f      and    y(%g) = %.1f ',X0,Y1,X0+h,Y2,X0+2*h,Y0)

```

Scilab code Exa 16.2 Linear Shooting Method

```

1 //Example 16.2
2 //Linear Shooting Method
3 //Page no. 576
4 clc;close;clear;
5
6 deff( 'y=f1 (x ,y ,y1 ) ', 'y=-x*y1+x^2*y+2*x^3 ')
7 deff( 'y=F1(x ,y ,y1 ) ', 'y=-x*y1+x^2*y+2*x^3 ')
8 deff( 'y=F2(x ,y ,y1 ) ', 'y=-x*y1+x^2*y ')
9 a=0;b=1;
10 y0=1;y1=-1;n=5;

```

```

11 h=(b-a)/n
12 y=y0; y01=0; x=a;
13 for i=0:5
14     yi1(1,i+1)=y
15     K1=h*y01;
16     R1=h*F1(x,y,y01);
17     K2=h*(y+R1/2);
18     R2=h*F1(x+h/2,y+K1/2,y01+R1/2)
19     K3=h*(y01+R2/2)
20     R3=h*F1(x+h/2,y+K2/2,y01+R2/2)
21     K4=h*(y+R3)
22     R4=h*F1(x+h,y+K3,y01+R3)
23     y=y+(K1+2*K2+2*K3+K4)/6
24     y01=y01+(R1+2*R2+2*R3+R4)/6
25     x=x+h
26 end
27 y=0; y01=1; x=a;
28 for i=0:5
29     yi2(1,i+1)=y
30     K1=h*y01;
31     R1=h*F2(x,y,y01);
32     K2=h*(y+R1/2);
33     R2=h*F2(x+h/2,y+K1/2,y01+R1/2)
34     K3=h*(y01+R2/2)
35     R3=h*F2(x+h/2,y+K2/2,y01+R2/2)
36     K4=h*(y+R3)
37     R4=h*F2(x+h,y+K3,y01+R3)
38     y=y+(K1+2*K2+2*K3+K4)/6
39     y01=y01+(R1+2*R2+2*R3+R4)/6
40     x=x+h
41 end
42 for i=1:6
43     yi(i)=yi1(1,i)+((y1-yi1(6))/yi2(6))*yi2(i)
44 end
45 y=1; x=a; y01=y1
46 for i=0:5
47     yir(1,i+1)=y;
48     K1=h*y01;

```

```

49     R1=h*f1(x,y,y01);
50     K2=h*(y+R1/2);
51     R2=h*f1(x+h/2,y+K1/2,y01+R1/2)
52     K3=h*(y01+R2/2)
53     R3=h*f1(x+h/2,y+K2/2,y01+R2/2)
54     K4=h*(y+R3)
55     R4=h*f1(x+h,y+K3,y01+R3)
56     y=y+(K1+2*K2+2*K3+K4)/6
57     y01=y01+(R1+2*R2+2*R3+R4)/6
58     x=x+h
59 end
60 x=a;
61 printf( '\n
          n\tx ')
62 for i=1:6
63   printf( '\t%.1f\t',x)
64   x=x+h
65 end
66 printf( '\n\ty ')
67 for i=1:6
68   printf( '\t%.4f\t',yi(i))
69 end
70 printf( '\n      by RK')
71 for i=1:6
72   printf( '\t%.4f\t',yir(i))
73 end
74 printf( '\n
          ')
75 printf( '\n\n\nNote: Computation error in calculation
          of values by RK method performed in book')

```

Scilab code Exa 16.3 Multiple Shooting Method

```
1 //Example 16.3
2 //Multiple Shooting Method
3 //Page no. 577
4 clc;close;clear;
5
6 h=0.25;x=0;y1=0;
7 deff( 'y=f (x)' , 'y=-(4*h^2)/(1+x)^2' )
8 deff( 'y=f1 (x)' , 'y=-2*(1+(h^2)/(1+x)^2) ' )
9
10 for i=1:4
11     x=x+h
12     B(i)=f(x);
13     for j=1:4
14         if i==4 & i==j
15             A(i,j)=f1(x)+1/4
16             A(i,j-1)=2
17         elseif j==i then
18             A(i,j)=f1(x)
19             A(i,j+1)=1
20         if j-1~=0 then
21             A(i,j-1)=1
22         end
23     end
24 end
25 end
26 y=inv(A)*B
27 disp(B,"B =",A,'A = ')
28 printf('\n\n\n x :')
29 for i=1:5
30     printf('\t%.2f',x)
31     x=x+h
32 end
33 x=0;printf('\n y :\t%.2f',y1);
34 for i=1:4
35     printf('\t%.4f',y(i))
36 end
```

Scilab code Exa 16.4 Finite Difference Method

```
1 //Example 16.4
2 //Finite Difference Method
3 //Page no. 582
4 clc;close;clear;
5
6 x=0;h=0.25;q=-1;Y(1)=-2;Y(5)=1;
7 printf('n i\txi\tYi\tpi\tqi\tri\n
     _____\n
      ')
8 for i=1:5
9     r(i)=-x^2
10    if i>1 & i<5 then
11        printf(' %i\t%g\t%s\t%g\t%i\t%g\n',i-1,x,"?"
12             ,x,q,r(i))
13    else
14        printf(' %i\t%g\t%g\t%g\t%i\t%g\n',i-1,x,Y(i)
15             ,x,q,r(i))
16    end
17    x=x+h
18 end
19 x=0;
20 printf('
     _____\n
      ')
21 for i=1:3
22     x=x+h
23     for j=1:3
24         if i==j then
25             A(i,j)=2+h^2*q
26         elseif i<j & abs(i-j)^=2
27             A(i,j)=2*h*(q+1)
28         else
29             A(i,j)=h*(q+1)
30         end
31     end
32 end
33
34 for i=1:3
35     for j=1:3
36         if i==j then
37             A(i,j)=2+h^2*q
38         elseif i<j & abs(i-j)^=2
39             A(i,j)=2*h*(q+1)
40         else
41             A(i,j)=h*(q+1)
42         end
43     end
44 end
45
46 for i=1:3
47     for j=1:3
48         if i==j then
49             A(i,j)=2+h^2*q
50         elseif i<j & abs(i-j)^=2
51             A(i,j)=2*h*(q+1)
52         else
53             A(i,j)=h*(q+1)
54         end
55     end
56 end
57
58 for i=1:3
59     for j=1:3
60         if i==j then
61             A(i,j)=2+h^2*q
62         elseif i<j & abs(i-j)^=2
63             A(i,j)=2*h*(q+1)
64         else
65             A(i,j)=h*(q+1)
66         end
67     end
68 end
69
70 for i=1:3
71     for j=1:3
72         if i==j then
73             A(i,j)=2+h^2*q
74         elseif i<j & abs(i-j)^=2
75             A(i,j)=2*h*(q+1)
76         else
77             A(i,j)=h*(q+1)
78         end
79     end
80 end
81
82 for i=1:3
83     for j=1:3
84         if i==j then
85             A(i,j)=2+h^2*q
86         elseif i<j & abs(i-j)^=2
87             A(i,j)=2*h*(q+1)
88         else
89             A(i,j)=h*(q+1)
90         end
91     end
92 end
93
94 for i=1:3
95     for j=1:3
96         if i==j then
97             A(i,j)=2+h^2*q
98         elseif i<j & abs(i-j)^=2
99             A(i,j)=2*h*(q+1)
100            elseif i>j & abs(i-j)^=2
101                A(i,j)=2*h*(q+1)
102            else
103                A(i,j)=h*(q+1)
104            end
105        end
106    end
107 end
108
109 for i=1:3
110     for j=1:3
111         if i==j then
112             A(i,j)=2+h^2*q
113         elseif i<j & abs(i-j)^=2
114             A(i,j)=2*h*(q+1)
115         else
116             A(i,j)=h*(q+1)
117         end
118     end
119 end
120
121 for i=1:3
122     for j=1:3
123         if i==j then
124             A(i,j)=2+h^2*q
125         elseif i<j & abs(i-j)^=2
126             A(i,j)=2*h*(q+1)
127         else
128             A(i,j)=h*(q+1)
129         end
130     end
131 end
132
133 for i=1:3
134     for j=1:3
135         if i==j then
136             A(i,j)=2+h^2*q
137         elseif i<j & abs(i-j)^=2
138             A(i,j)=2*h*(q+1)
139         else
140             A(i,j)=h*(q+1)
141         end
142     end
143 end
144
145 for i=1:3
146     for j=1:3
147         if i==j then
148             A(i,j)=2+h^2*q
149         elseif i<j & abs(i-j)^=2
150             A(i,j)=2*h*(q+1)
151         else
152             A(i,j)=h*(q+1)
153         end
154     end
155 end
156
157 for i=1:3
158     for j=1:3
159         if i==j then
160             A(i,j)=2+h^2*q
161         elseif i<j & abs(i-j)^=2
162             A(i,j)=2*h*(q+1)
163         else
164             A(i,j)=h*(q+1)
165         end
166     end
167 end
168
169 for i=1:3
170     for j=1:3
171         if i==j then
172             A(i,j)=2+h^2*q
173         elseif i<j & abs(i-j)^=2
174             A(i,j)=2*h*(q+1)
175         else
176             A(i,j)=h*(q+1)
177         end
178     end
179 end
180
181 for i=1:3
182     for j=1:3
183         if i==j then
184             A(i,j)=2+h^2*q
185         elseif i<j & abs(i-j)^=2
186             A(i,j)=2*h*(q+1)
187         else
188             A(i,j)=h*(q+1)
189         end
190     end
191 end
192
193 for i=1:3
194     for j=1:3
195         if i==j then
196             A(i,j)=2+h^2*q
197         elseif i<j & abs(i-j)^=2
198             A(i,j)=2*h*(q+1)
199         else
200             A(i,j)=h*(q+1)
201         end
202     end
203 end
204
205 for i=1:3
206     for j=1:3
207         if i==j then
208             A(i,j)=2+h^2*q
209         elseif i<j & abs(i-j)^=2
210             A(i,j)=2*h*(q+1)
211         else
212             A(i,j)=h*(q+1)
213         end
214     end
215 end
216
217 for i=1:3
218     for j=1:3
219         if i==j then
220             A(i,j)=2+h^2*q
221         elseif i<j & abs(i-j)^=2
222             A(i,j)=2*h*(q+1)
223         else
224             A(i,j)=h*(q+1)
225         end
226     end
227 end
228
229 for i=1:3
230     for j=1:3
231         if i==j then
232             A(i,j)=2+h^2*q
233         elseif i<j & abs(i-j)^=2
234             A(i,j)=2*h*(q+1)
235         else
236             A(i,j)=h*(q+1)
237         end
238     end
239 end
240
241 for i=1:3
242     for j=1:3
243         if i==j then
244             A(i,j)=2+h^2*q
245         elseif i<j & abs(i-j)^=2
246             A(i,j)=2*h*(q+1)
247         else
248             A(i,j)=h*(q+1)
249         end
250     end
251 end
252
253 for i=1:3
254     for j=1:3
255         if i==j then
256             A(i,j)=2+h^2*q
257         elseif i<j & abs(i-j)^=2
258             A(i,j)=2*h*(q+1)
259         else
260             A(i,j)=h*(q+1)
261         end
262     end
263 end
264
265 for i=1:3
266     for j=1:3
267         if i==j then
268             A(i,j)=2+h^2*q
269         elseif i<j & abs(i-j)^=2
270             A(i,j)=2*h*(q+1)
271         else
272             A(i,j)=h*(q+1)
273         end
274     end
275 end
276
277 for i=1:3
278     for j=1:3
279         if i==j then
280             A(i,j)=2+h^2*q
281         elseif i<j & abs(i-j)^=2
282             A(i,j)=2*h*(q+1)
283         else
284             A(i,j)=h*(q+1)
285         end
286     end
287 end
288
289 for i=1:3
290     for j=1:3
291         if i==j then
292             A(i,j)=2+h^2*q
293         elseif i<j & abs(i-j)^=2
294             A(i,j)=2*h*(q+1)
295         else
296             A(i,j)=h*(q+1)
297         end
298     end
299 end
300
301 for i=1:3
302     for j=1:3
303         if i==j then
304             A(i,j)=2+h^2*q
305         elseif i<j & abs(i-j)^=2
306             A(i,j)=2*h*(q+1)
307         else
308             A(i,j)=h*(q+1)
309         end
310     end
311 end
312
313 for i=1:3
314     for j=1:3
315         if i==j then
316             A(i,j)=2+h^2*q
317         elseif i<j & abs(i-j)^=2
318             A(i,j)=2*h*(q+1)
319         else
320             A(i,j)=h*(q+1)
321         end
322     end
323 end
324
325 for i=1:3
326     for j=1:3
327         if i==j then
328             A(i,j)=2+h^2*q
329         elseif i<j & abs(i-j)^=2
330             A(i,j)=2*h*(q+1)
331         else
332             A(i,j)=h*(q+1)
333         end
334     end
335 end
336
337 for i=1:3
338     for j=1:3
339         if i==j then
340             A(i,j)=2+h^2*q
341         elseif i<j & abs(i-j)^=2
342             A(i,j)=2*h*(q+1)
343         else
344             A(i,j)=h*(q+1)
345         end
346     end
347 end
348
349 for i=1:3
350     for j=1:3
351         if i==j then
352             A(i,j)=2+h^2*q
353         elseif i<j & abs(i-j)^=2
354             A(i,j)=2*h*(q+1)
355         else
356             A(i,j)=h*(q+1)
357         end
358     end
359 end
360
361 for i=1:3
362     for j=1:3
363         if i==j then
364             A(i,j)=2+h^2*q
365         elseif i<j & abs(i-j)^=2
366             A(i,j)=2*h*(q+1)
367         else
368             A(i,j)=h*(q+1)
369         end
370     end
371 end
372
373 for i=1:3
374     for j=1:3
375         if i==j then
376             A(i,j)=2+h^2*q
377         elseif i<j & abs(i-j)^=2
378             A(i,j)=2*h*(q+1)
379         else
380             A(i,j)=h*(q+1)
381         end
382     end
383 end
384
385 for i=1:3
386     for j=1:3
387         if i==j then
388             A(i,j)=2+h^2*q
389         elseif i<j & abs(i-j)^=2
390             A(i,j)=2*h*(q+1)
391         else
392             A(i,j)=h*(q+1)
393         end
394     end
395 end
396
397 for i=1:3
398     for j=1:3
399         if i==j then
400             A(i,j)=2+h^2*q
401         elseif i<j & abs(i-j)^=2
402             A(i,j)=2*h*(q+1)
403         else
404             A(i,j)=h*(q+1)
405         end
406     end
407 end
408
409 for i=1:3
410     for j=1:3
411         if i==j then
412             A(i,j)=2+h^2*q
413         elseif i<j & abs(i-j)^=2
414             A(i,j)=2*h*(q+1)
415         else
416             A(i,j)=h*(q+1)
417         end
418     end
419 end
420
421 for i=1:3
422     for j=1:3
423         if i==j then
424             A(i,j)=2+h^2*q
425         elseif i<j & abs(i-j)^=2
426             A(i,j)=2*h*(q+1)
427         else
428             A(i,j)=h*(q+1)
429         end
430     end
431 end
432
433 for i=1:3
434     for j=1:3
435         if i==j then
436             A(i,j)=2+h^2*q
437         elseif i<j & abs(i-j)^=2
438             A(i,j)=2*h*(q+1)
439         else
440             A(i,j)=h*(q+1)
441         end
442     end
443 end
444
445 for i=1:3
446     for j=1:3
447         if i==j then
448             A(i,j)=2+h^2*q
449         elseif i<j & abs(i-j)^=2
450             A(i,j)=2*h*(q+1)
451         else
452             A(i,j)=h*(q+1)
453         end
454     end
455 end
456
457 for i=1:3
458     for j=1:3
459         if i==j then
460             A(i,j)=2+h^2*q
461         elseif i<j & abs(i-j)^=2
462             A(i,j)=2*h*(q+1)
463         else
464             A(i,j)=h*(q+1)
465         end
466     end
467 end
468
469 for i=1:3
470     for j=1:3
471         if i==j then
472             A(i,j)=2+h^2*q
473         elseif i<j & abs(i-j)^=2
474             A(i,j)=2*h*(q+1)
475         else
476             A(i,j)=h*(q+1)
477         end
478     end
479 end
480
481 for i=1:3
482     for j=1:3
483         if i==j then
484             A(i,j)=2+h^2*q
485         elseif i<j & abs(i-j)^=2
486             A(i,j)=2*h*(q+1)
487         else
488             A(i,j)=h*(q+1)
489         end
490     end
491 end
492
493 for i=1:3
494     for j=1:3
495         if i==j then
496             A(i,j)=2+h^2*q
497         elseif i<j & abs(i-j)^=2
498             A(i,j)=2*h*(q+1)
499         else
500             A(i,j)=h*(q+1)
501         end
502     end
503 end
504
505 for i=1:3
506     for j=1:3
507         if i==j then
508             A(i,j)=2+h^2*q
509         elseif i<j & abs(i-j)^=2
510             A(i,j)=2*h*(q+1)
511         else
512             A(i,j)=h*(q+1)
513         end
514     end
515 end
516
517 for i=1:3
518     for j=1:3
519         if i==j then
520             A(i,j)=2+h^2*q
521         elseif i<j & abs(i-j)^=2
522             A(i,j)=2*h*(q+1)
523         else
524             A(i,j)=h*(q+1)
525         end
526     end
527 end
528
529 for i=1:3
530     for j=1:3
531         if i==j then
532             A(i,j)=2+h^2*q
533         elseif i<j & abs(i-j)^=2
534             A(i,j)=2*h*(q+1)
535         else
536             A(i,j)=h*(q+1)
537         end
538     end
539 end
540
541 for i=1:3
542     for j=1:3
543         if i==j then
544             A(i,j)=2+h^2*q
545         elseif i<j & abs(i-j)^=2
546             A(i,j)=2*h*(q+1)
547         else
548             A(i,j)=h*(q+1)
549         end
550     end
551 end
552
553 for i=1:3
554     for j=1:3
555         if i==j then
556             A(i,j)=2+h^2*q
557         elseif i<j & abs(i-j)^=2
558             A(i,j)=2*h*(q+1)
559         else
560             A(i,j)=h*(q+1)
561         end
562     end
563 end
564
565 for i=1:3
566     for j=1:3
567         if i==j then
568             A(i,j)=2+h^2*q
569         elseif i<j & abs(i-j)^=2
570             A(i,j)=2*h*(q+1)
571         else
572             A(i,j)=h*(q+1)
573         end
574     end
575 end
576
577 for i=1:3
578     for j=1:3
579         if i==j then
580             A(i,j)=2+h^2*q
581         elseif i<j & abs(i-j)^=2
582             A(i,j)=2*h*(q+1)
583         else
584             A(i,j)=h*(q+1)
585         end
586     end
587 end
588
589 for i=1:3
590     for j=1:3
591         if i==j then
592             A(i,j)=2+h^2*q
593         elseif i<j & abs(i-j)^=2
594             A(i,j)=2*h*(q+1)
595         else
596             A(i,j)=h*(q+1)
597         end
598     end
599 end
600
601 for i=1:3
602     for j=1:3
603         if i==j then
604             A(i,j)=2+h^2*q
605         elseif i<j & abs(i-j)^=2
606             A(i,j)=2*h*(q+1)
607         else
608             A(i,j)=h*(q+1)
609         end
610     end
611 end
612
613 for i=1:3
614     for j=1:3
615         if i==j then
616             A(i,j)=2+h^2*q
617         elseif i<j & abs(i-j)^=2
618             A(i,j)=2*h*(q+1)
619         else
620             A(i,j)=h*(q+1)
621         end
622     end
623 end
624
625 for i=1:3
626     for j=1:3
627         if i==j then
628             A(i,j)=2+h^2*q
629         elseif i<j & abs(i-j)^=2
630             A(i,j)=2*h*(q+1)
631         else
632             A(i,j)=h*(q+1)
633         end
634     end
635 end
636
637 for i=1:3
638     for j=1:3
639         if i==j then
640             A(i,j)=2+h^2*q
641         elseif i<j & abs(i-j)^=2
642             A(i,j)=2*h*(q+1)
643         else
644             A(i,j)=h*(q+1)
645         end
646     end
647 end
648
649 for i=1:3
650     for j=1:3
651         if i==j then
652             A(i,j)=2+h^2*q
653         elseif i<j & abs(i-j)^=2
654             A(i,j)=2*h*(q+1)
655         else
656             A(i,j)=h*(q+1)
657         end
658     end
659 end
660
661 for i=1:3
662     for j=1:3
663         if i==j then
664             A(i,j)=2+h^2*q
665         elseif i<j & abs(i-j)^=2
666             A(i,j)=2*h*(q+1)
667         else
668             A(i,j)=h*(q+1)
669         end
670     end
671 end
672
673 for i=1:3
674     for j=1:3
675         if i==j then
676             A(i,j)=2+h^2*q
677         elseif i<j & abs(i-j)^=2
678             A(i,j)=2*h*(q+1)
679         else
680             A(i,j)=h*(q+1)
681         end
682     end
683 end
684
685 for i=1:3
686     for j=1:3
687         if i==j then
688             A(i,j)=2+h^2*q
689         elseif i<j & abs(i-j)^=2
690             A(i,j)=2*h*(q+1)
691         else
692             A(i,j)=h*(q+1)
693         end
694     end
695 end
696
697 for i=1:3
698     for j=1:3
699         if i==j then
700             A(i,j)=2+h^2*q
701         elseif i<j & abs(i-j)^=2
702             A(i,j)=2*h*(q+1)
703         else
704             A(i,j)=h*(q+1)
705         end
706     end
707 end
708
709 for i=1:3
710     for j=1:3
711         if i==j then
712             A(i,j)=2+h^2*q
713         elseif i<j & abs(i-j)^=2
714             A(i,j)=2*h*(q+1)
715         else
716             A(i,j)=h*(q+1)
717         end
718     end
719 end
720
721 for i=1:3
722     for j=1:3
723         if i==j then
724             A(i,j)=2+h^2*q
725         elseif i<j & abs(i-j)^=2
726             A(i,j)=2*h*(q+1)
727         else
728             A(i,j)=h*(q+1)
729         end
730     end
731 end
732
733 for i=1:3
734     for j=1:3
735         if i==j then
736             A(i,j)=2+h^2*q
737         elseif i<j & abs(i-j)^=2
738             A(i,j)=2*h*(q+1)
739         else
740             A(i,j)=h*(q+1)
741         end
742     end
743 end
744
745 for i=1:3
746     for j=1:3
747         if i==j then
748             A(i,j)=2+h^2*q
749         elseif i<j & abs(i-j)^=2
750             A(i,j)=2*h*(q+1)
751         else
752             A(i,j)=h*(q+1)
753         end
754     end
755 end
756
757 for i=1:3
758     for j=1:3
759         if i==j then
760             A(i,j)=2+h^2*q
761         elseif i<j & abs(i-j)^=2
762             A(i,j)=2*h*(q+1)
763         else
764             A(i,j)=h*(q+1)
765         end
766     end
767 end
768
769 for i=1:3
770     for j=1:3
771         if i==j then
772             A(i,j)=2+h^2*q
773         elseif i<j & abs(i-j)^=2
774             A(i,j)=2*h*(q+1)
775         else
776             A(i,j)=h*(q+1)
777         end
778     end
779 end
780
781 for i=1:3
782     for j=1:3
783         if i==j then
784             A(i,j)=2+h^2*q
785         elseif i<j & abs(i-j)^=2
786             A(i,j)=2*h*(q+1)
787         else
788             A(i,j)=h*(q+1)
789         end
790     end
791 end
792
793 for i=1:3
794     for j=1:3
795         if i==j then
796             A(i,j)=2+h^2*q
797         elseif i<j & abs(i-j)^=2
798             A(i,j)=2*h*(q+1)
799         else
800             A(i,j)=h*(q+1)
801         end
802     end
803 end
804
805 for i=1:3
806     for j=1:3
807         if i==j then
808             A(i,j)=2+h^2*q
809         elseif i<j & abs(i-j)^=2
810             A(i,j)=2*h*(q+1)
811         else
812             A(i,j)=h*(q+1)
813         end
814     end
815 end
816
817 for i=1:3
818     for j=1:3
819         if i==j then
820             A(i,j)=2+h^2*q
821         elseif i<j & abs(i-j)^=2
822             A(i,j)=2*h*(q+1)
823         else
824             A(i,j)=h*(q+1)
825         end
826     end
827 end
828
829 for i=1:3
830     for j=1:3
831         if i==j then
832             A(i,j)=2+h^2*q
833         elseif i<j & abs(i-j)^=2
834             A(i,j)=2*h*(q+1)
835         else
836             A(i,j)=h*(q+1)
837         end
838     end
839 end
840
841 for i=1:3
842     for j=1:3
843         if i==j then
844             A(i,j)=2+h^2*q
845         elseif i<j & abs(i-j)^=2
846             A(i,j)=2*h*(q+1)
847         else
848             A(i,j)=h*(q+1)
849         end
850     end
851 end
852
853 for i=1:3
854     for j=1:3
855         if i==j then
856             A(i,j)=2+h^2*q
857         elseif i<j & abs(i-j)^=2
858             A(i,j)=2*h*(q+1)
859         else
860             A(i,j)=h*(q+1)
861         end
862     end
863 end
864
865 for i=1:3
866     for j=1:3
867         if i==j then
868             A(i,j)=2+h^2*q
869         elseif i<j & abs(i-j)^=2
870             A(i,j)=2*h*(q+1)
871         else
872             A(i,j)=h*(q+1)
873         end
874     end
875 end
876
877 for i=1:3
878     for j=1:3
879         if i==j then
880             A(i,j)=2+h^2*q
881         elseif i<j & abs(i-j)^=2
882             A(i,j)=2*h*(q+1)
883         else
884             A(i,j)=h*(q+1)
885         end
886     end
887 end
888
889 for i=1:3
890     for j=1:3
891         if i==j then
892             A(i,j)=2+h^2*q
893         elseif i<j & abs(i-j)^=2
894             A(i,j)=2*h*(q+1)
895         else
896             A(i,j)=h*(q+1)
897         end
898     end
899 end
900
901 for i=1:3
902     for j=1:3
903         if i==j then
904             A(i,j)=2+h^2*q
905         elseif i<j & abs(i-j)^=2
906             A(i,j)=2*h*(q+1)
907         else
908             A(i,j)=h*(q+1)
909         end
910     end
911 end
912
913 for i=1:3
914     for j=1:3
915         if i==j then
916             A(i,j)=2+h^2*q
917         elseif i<j & abs(i-j)^=2
918             A(i,j)=2*h*(q+1)
919         else
920             A(i,j)=h*(q+1)
921         end
922     end
923 end
924
925 for i=1:3
926     for j=1:3
927         if i==j then
928             A(i,j)=2+h^2*q
929         elseif i<j & abs(i-j)^=2
930             A(i,j)=2*h*(q+1)
931         else
932             A(i,j)=h*(q+1)
933         end
934     end
935 end
936
937 for i=1:3
938     for j=1:3
939         if i==j then
940             A(i,j)=2+h^2*q
941         elseif i<j & abs(i-j)^=2
942             A(i,j)=2*h*(q+1)
943         else
944             A(i,j)=h*(q+1)
945         end
946     end
947 end
948
949 for i=1:3
950     for j=1:3
951         if i==j then
952             A(i,j)=2+h^2*q
953         elseif i<j & abs(i-j)^=2
954             A(i,j)=2*h*(q+1)
955         else
956             A(i,j)=h*(q+1)
957         end
958     end
959 end
960
961 for i=1:3
962     for j=1:3
963         if i==j then
964             A(i,j)=2+h^2*q
965         elseif i<j & abs(i-j)^=2
966             A(i,j)=2*h*(q+1)
967         else
968             A(i,j)=h*(q+1)
969         end
970     end
971 end
972
973 for i=1:3
974     for j=1:3
975         if i==j then
976             A(i,j)=2+h^2*q
977         elseif i<j & abs(i-j)^=2
978             A(i,j)=2*h*(q+1)
979         else
980             A(i,j)=h*(q+1)
981         end
982     end
983 end
984
985 for i=1:3
986     for j=1:3
987         if i==j then
988             A(i,j)=2+h^2*q
989         elseif i<j & abs(i-j)^=2
990             A(i,j)=2*h*(q+1)
991         else
992             A(i,j)=h*(q+1)
993         end
994     end
995 end
996
997 for i=1:3
998     for j=1:3
999         if i==j then
1000             A(i,j)=2+h^2*q
1001         elseif i<j & abs(i-j)^=2
1002             A(i,j)=2*h*(q+1)
1003         else
1004             A(i,j)=h*(q+1)
1005         end
1006     end
1007 end
1008
1009 for i=1:3
1010     for j=1:3
1011         if i==j then
1012             A(i,j)=2+h^2*q
1013         elseif i<j & abs(i-j)^=2
1014             A(i,j)=2*h*(q+1)
1015         else
1016             A(i,j)=h*(q+1)
1017         end
1018     end
1019 end
1020
1021 for i=1:3
1022     for j=1:3
1023         if i==j then
1024             A(i,j)=2+h^2*q
1025         elseif i<j & abs(i-j)^=2
1026             A(i,j)=2*h*(q+1)
1027         else
1028             A(i,j)=h*(q+1)
1029         end
1030     end
1031 end
1032
1033 for i=1:3
1034     for j=1:3
1035         if i==j then
1036             A(i,j)=2+h^2*q
1037         elseif i<j & abs(i-j)^=2
1038             A(i,j)=2*h*(q+1)
1039         else
1040             A(i,j)=h*(q+1)
1041         end
1042     end
1043 end
1044
1045 for i=1:3
1046     for j=1:3
1047         if i==j then
1048             A(i,j)=2+h^2*q
1049         elseif i<j & abs(i-j)^=2
1050             A(i,j)=2*h*(q+1)
1051         else
1052             A(i,j)=h*(q+1)
1053         end
1054     end
1055 end
1056
1057 for i=1:3
1058     for j=1:3
1059         if i==j then
1060             A(i,j)=2+h^2*q
1061         elseif i<j & abs(i-j)^=2
1062             A(i,j)=2*h*(q+1)
1063         else
1064             A(i,j)=h*(q+1)
1065         end
1066     end
1067 end
1068
1069 for i=1:3
1070     for j=1:3
1071         if i==j then
1072             A(i,j)=2+h^2*q
1073         elseif i<j & abs(i-j)^=2
1074             A(i,j)=2*h*(q+1)
1075         else
1076             A(i,j)=h*(q+1)
1077         end
1078     end
1079 end
1080
1081 for i=1:3
1082     for j=1:3
1083         if i==j then
1084             A(i,j)=2+h^2*q
1085         elseif i<j & abs(i-j)^=2
1086             A(i,j)=2*h*(q+1)
1087         else
1088             A(i,j)=h*(q+1)
1089         end
1090     end
1091 end
1092
1093 for i=1:3
1094     for j=1:3
1095         if i==j then
1096             A(i,j)=2+h^2*q
1097         elseif i<j & abs(i-j)^=2
1098             A(i,j)=2*h*(q+1)
1099         else
1100             A(i,j)=h*(q+1)
1101         end
1102     end
1103 end
1104
1105 for i=1:3
1106     for j=1:3
1107         if i==j then
1108             A(i,j)=2+h^2*q
1109         elseif i<j & abs(i-j)^=2
1110             A(i,j)=2*h*(q+1)
1111         else
1112             A(i,j)=h*(q+1)
1113         end
1114     end
1115 end
1116
1117 for i=1:3
1118     for j=1:3
1119         if i==j then
1120             A(i,j)=2+h^2*q
1121         elseif i<j & abs(i-j)^=2
1122             A(i,j)=2*h*(q+1)
1123         else
1124             A(i,j)=h*(q+1)
1125         end
1126     end
1127 end
1128
1129 for i=1:3
1130     for j=1:3
1131        
```

```

25         A(i,j)=-1+h*x/2
26     elseif i>j & abs(i-j)^=2
27         A(i,j)=-1-h*x/2
28     end
29 end
30 if i==3 then
31     B(i)=-h^2*r(i+1)+(-h*x/2+1)*Y(1+2*(i-1))
32 else
33     B(i)=-h^2*r(i+1)+(h*x/2+1)*Y(1+2*(i-1))
34 end
35 B(i)=(-1)^(i+1)*B(i)
36 end
37 disp(B,"B =",A,'A = ')
38 y=inv(A)*B
39 for i=1:3
40     Y(i+1)=y(i)
41 end
42 x=0;
43 disp("The Solution is : ",B,"B =",A,'A = ')
44 printf(' x : ')
45 for i=1:5
46     printf('\t %.2f ',x)
47     x=x+h
48 end
49 x=0;printf('\n y : ');
50 for i=1:5
51     printf('\t%.3f ',Y(i))
52 end

```

Scilab code Exa 16.5 Non Linear Problem

```

1 //Example 16.5
2 //Non Linear Problem

```

```

3 //Page no. 584
4 clc;close;clear;
5
6 def(f,'y=f(x)', 'y=2/(1+x)')
7 Y=[1,0.75,0.75,0.75,0.5];h=0.25
8 A=[-2,1,0;1,-2,1;0,1,-2];A_1=inv(A)
9 disp(A_1," Inverse of A =",A,"A =")
10 printf('\nThe Solution of the system is: \n\n
    Iteration\t Y0\t\t Y1\t\t Y2\t\t Y3\t\t Y4\n
')
11 for i=0:6
12     printf('\n      %i',i)
13     for j=1:5
14         if j<4 & i~=0 then
15             Y(j+1)=y(j)
16         end
17         printf(' \t\t%.4f',Y(j))
18     end
19     x=0;
20     for j=1:3
21         x=x+h
22         if j~=2 then
23             B(j)=h^2*f(x)*Y(j+1)^2-Y(1+2*(j-1))
24         else
25             B(j)=h^2*f(x)*Y(j+1)^2
26         end
27     end
28     y=A_1*B
29 end

```

Scilab code Exa 16.6 Collocation Method

```

1 //Example 16.6
2 //Collocation Method
3 //Page no. 589
4 clc; close; clear;
5
6 h1=0.000001; h=0.25; x=0;
7 Y(1)=0; Y(5)=0;
8 deff( 'y=p(x) ', 'y=1')
9 deff( 'y=q(x) ', 'y=-2/(1+x)^2 ')
10 deff( 'y=f(x) ', 'y=(2*x-4)/(1+x)^4 ')
11 deff( 'y=fi(x,j) ', 'y=(1-x)*x^j ')
12 deff( 'y=f1(x,y) ', 'y=(-x+y)/h1' ) //function for
    differentiation
13 for i=1:4
14     x=x+h
15     for j=1:4
16         A(i,j)=p(x)*f1(f1(fi(x,j),fi(x+h1,j)),f1(fi(
            x+h1,j),fi(x+2*h1,j)))+f1(p(x),p(x+h1))*f1(
            fi(x,j),fi(x+h1,j))+q(x)*fi(x,j)
17     end
18 end
19 x=0;
20 for i=1:4
21     x=x+h
22     B(i)=f(x)
23 end
24 disp(B, 'B = ',A,"A =")
25 C=inv(A)*B
26 x=0;
27 for i=2:4
28     x=x+h;
29     for j=1:4
30         Y(i)=Y(i)+C(j)*fi(x,j)
31     end
32 end
33 disp(Y," Solution Matrix Y = ")

```

Chapter 18

Numerical Solutions of Parabolic Partial Differential Equations

Scilab code Exa 18.4 Forward Difference Method

```
1 //Example 18.4
2 //Forward Difference Method
3 //Page no. 624
4 clc;clear;close;
5
6 h=0.2;k=0.02;
7 r=k/h^2;
8 printf ('\n j \t t \t | \t i -->\t ')
9 for i=0:5
10     printf ('    %i\t ',i)
11 end
12 printf ('\n | \t | \t | \t x -->\t ')
13 for i=0:5
14     printf ('%.3f\t ',(i)/5)
15 end
```

```

16 printf( '\n
17   ')
18   for j=1:6
19     printf( '\n %i\t%.3f\t| \t\t',j-1,(j-1)/50)
20     for i=1:6
21       if i==1 | i==6 then
22         u(j,i)=0;
23       elseif j==1 then
24         u(j,i)=sin(pi*(i-1)/5)
25       else
26         u(j,i)=(u(j-1,i-1)+u(j-1,i+1))/2
27       end
28     printf( '%.3f\t',u(j,i))
29   end
29 end

```

Scilab code Exa 18.5 Bender Schmidt Method

```

1 //Example 18.5
2 //Bender Schmidt Method
3 //Page no. 625
4 clc;clear;close;
5
6 h=0.1;k=0.005;
7 r=k/h^2;
8 printf( '\n j | \t i -->\t ')
9 for i=0:10
10   printf( ' %i\t',i)
11 end
12 printf( '\n | \t x -->\t ')
13 for i=0:10
14   printf( '%.3f\t',(i)/10)

```

```

15 end
16 printf( '\n


---


    ')
17 for j=1:9
18     printf( '\n %i | \t\t',j-1)
19     for i=1:11
20         if i==1 | i==11 then
21             u(j,i)=0;
22         elseif j==1 then
23             u(j,i)=sin(%pi*(i-1)/10)
24         else
25             u(j,i)=u(j-1,i)/2+(u(j-1,i-1)+u(j-1,i+1))
26             )/4
27         end
28     printf( '%.3f\t',u(j,i))
29 end


---



```

Scilab code Exa 18.6 Crank Nicolson Method

```

1 //Example 18.6
2 //Crank Nicolson Method
3 //Page no. 631
4 clc;clear;close;
5 h=1/2;k=1/8;
6 r=k/h^2;
7 for i=1:2:3
8     for j=1:9
9         if i==1 | j==1 then
10             u(i,j)=0;
11         end
12         if i==3 then

```

```

13           u(i,j)=(j-1)/8
14       end
15   end
16 end
17 a=[3,-1,0;-1,3,-1;0,-1,3];
18 a=inv(a);
19 for j=2:9
20     u(2,j)=(u(1,j-1)+2*u(2,j-1)+u(3,j-1)+u(1,j) +
21         u(3,j))/6
22 end
23 u=u'
24 printf('\nfor h = 1/2\n')
25 printf('i\nj --> ')
26 for i=1:3
27     printf('\t%i\t',i)
28 printf('\n
29 n')
30 for i=1:9
31     printf('\n %i',i)
32     for j=1:3
33         printf('\t %.9f',u(i,j))
34     end
35 end
36
37
38
39 h=1/4;k=1/8;
40 r=k/h^2;
41 for i=1:4:5
42     for j=1:9
43         if i==1 | j==1 then
44             u(i,j)=0;
45         end
46         if i==5 then
47             u(i,j)=(j-1)/8

```

```

48         end
49     end
50 end
51 a=[3,-1,0;-1,3,-1;0,-1,3];
52 a=inv(a);
53 for j=2:9
54     b=[u(1,j-1)-u(2,j-1)+u(3,j-1)+u(1,j);u(2,j
      -1)-u(3,j-1)+u(4,j-1);u(3,j-1)-u(4,j-1)+u
      (5,j-1)+u(5,j)]
55     x=a*b
56     u(2,j)=x(1);u(3,j)=x(2);u(4,j)=x(3);
57 end
58 u=u'
59 printf('n\n\n\n\nfor h = 1/4\n')
60 printf('i\nj --> ')
61 for i=1:5
62     printf('\t%i\t',i)
63 end
64 printf('\n
      n')
65 for i=1:9
66     printf('\n %i',i)
67     for j=1:5
68         printf('\t %.9f',u(i,j))
69     end
70 end

```

Scilab code Exa 18.7 Gauss Seidel Method

```

1 //Example 18.7
2 //Gauss Seidel Method
3 //Page no. 637

```

```

4 clc;clear;close;
5 def('y=f(x)', 'y=4*x-4*x^2')
6 h=0.2;k=0.04;
7 r=k/h^2;
8 printf('\n k\t|\t i -->\t')
9 for i=0:5
10     printf(' %i\t',i)
11 end
12 printf('\n |\t|\t x -->\t')
13 for i=0:5
14     printf('%.2f\t',(i)/5)
15 end
16 printf('\n
')
17 for k=1:7
18     printf('\n %i\t|\t|\t',k-1)
19     for i=1:6
20         if i==1 | i==6 then
21             u(k,i)=0;
22         elseif k==1 then
23             u(k,i)=f((i-1)/5)
24         else
25             u(k,i)=(u(k-1,i-1)+u(k-1,i+1))/2
26         end
27         printf('%.2f\t',u(k,i))
28     end
29 end

```

Scilab code Exa 18.8 ADI Method

```

1 //Example 18.8
2 //ADI Method

```

```

3 //Page no. 642
4 clc;clear;close;
5
6 for i=1:4
7     for j=1:5
8         P(i,j)=20
9     end
10 end
11 r=1;k=0;
12 for i=1:6
13     v1(i)=20
14     u1(i)=20
15 end
16 P1
=[25,30,35,50,60;35,0,0,0,70;45,0,0,0,80;60,70,80,100,90]

17 for i=1:4
18     printf('\n')
19     for j=1:5
20         printf('%i\t',P(i,j))
21     end
22     if i==2 then
23         printf('-->')
24     end
25     printf('\t')
26     for j=1:5
27         printf('%i\t',P1(i,j))
28         if i>1 & i<4 & j>1 & j<5 then
29             P1(i,j)=P(i,j)
30         end
31     end
32 end
33 P1v=P1;P1h=P1;
34 for i=1:6
35     for j=1:6
36         if i==j then
37             Av(i,j)=1+2*r
38         elseif abs(i-j)==1 & i+j~=5 & i+j~=9

```

```

39           Av(i,j)=-r
40       end
41   end
42 end
43 for i=1:6
44     for j=1:6
45         if i==j then
46             Ah(i,j)=1+2*r
47         elseif abs(i-j)==1 & i+j~=7
48             Ah(i,j)=-r
49         end
50     end
51 end
52 n=8
53 for l=1:n
54     k=0;
55 for j=0:2
56     for i=1:2
57         if i==1 then
58             Bv(i+j+k)=r*P1h(i+1,j+1)+(1-2*r)*P1h(i
59             +1,j+2)+r*P1h(i+1,j+3)+r*P1h(i,j+1)
60         else
61             Bv(i+j+k)=r*P1h(i+1,j+1)+(1-2*r)*P1h(i
62             +1,j+2)+r*P1h(i+1,j+3)+r*P1h(i+2,j+1)
63         end
64     end
65     k=k+1;
66 end
67 k=0;
68 Bh=[r*30+(1-2*r)*v1(1)+r*v1(4)+r*35;r*35+(1-2*r)*v1
69     (3)+r*v1(5);r*v1(1)+(1-2*r)*v1(2)+r*v1(3)+r*(70);
70     r*v1(1)+(1-2*r)*v1(2)+r*(70+45);r*v1(3)+(1-2*r)*
71     v1(4)+r*80;r*v1(5)+(1-2*r)*v1(6)+r*(100+80)]
72     for i=1:6
73         v(i,l)=v1(i)
74     end
75     for i=1:6
76         u(i,l)=u1(i)

```

```

72     end
73     v1=inv(Av)*Bv
74     u1=inv(Ah)*Bh
75     k=1;
76     for i=2:3
77         for j=2:4
78             P1h(i,j)=u1(i+j+k-4)
79         end
80         k=k+2
81     end
82     k=0;
83     for j=2:4
84         for i=2:3
85             P1v(i,j)=v1(i+j+k-3)
86         end
87         k=k+1
88     end
89 end
90 printf('\n\n\n\nResults for Vertical Transverse in
Celsius :\n')
91 for i=1:7
92     printf('\n')
93     if i==1 then
94         printf('Itr -->')
95         for j=1:n
96             printf('\t %i',j-1)
97         end
98         printf('\n
')
99     else
100         printf(' v%0i',i-1)
101         for j=1:n
102             printf('\t%.2f',v(i-1,j))
103         end
104     end
105 end
106 printf('\n\n\n\nResults for Horizontal Transverse in

```

```
Celsius :\'n')
107 for i=1:7
108     printf('\'n')
109     if i==1 then
110         printf(' Itr -->')
111         for j=1:n
112             printf('\t %i',j-1)
113         end
114         printf('\'n
115     ')
116     else
117         printf(' u%0i',i-1)
118         for j=1:n
119             printf('\t%.2f',u(i-1,j))
120         end
121     end
```

Chapter 19

Numerical Solutions of Hyperbolic Partial Differential Equations

Scilab code Exa 19.3 Simple Explicit Method

```
1 //Example 19.3
2 //Simple Explicit Method
3 //Page no. 658
4 clc;clear;close;
5
6 c=-2;dt=0.07;dx=0.2;
7 r=abs(c)*dt/dx;
8 printf ('\n x\ ti\ t|\ tj -->\t ')
9 for i=0:6
10     printf (' %i\t ',i)
11 end
12 printf ('\n |\ t|\ t|\ tt -->\t ')
13 for i=0:6
14     printf ('%.3 f\t ',i*dt)
15 end
```

```

16 printf( '\n
17   ')
18   for j=1:6
19     printf( '\n %.1f\t%i\t|\t| \t\t', (j-1)*dx ,j-1)
20     for i=1:7
21       if i==1 then
22         u(j,i)=0;
23       elseif j==1 then
24         u(j,i)=1
25       else
26         u(j,i)=(1-r)*u(j,i-1)+r*u(j-1,i-1)
27       end
28     end
29   end

```

Scilab code Exa 19.4 Simple Implicit Method

```

1 //Example 19.4
2 //Simple Implicit Method
3 //Page no. 659
4 clc;clear;close;
5
6 c=-2;dt=0.07;dx=0.2;
7 r=abs(c)*dt/dx;
8 printf( '\n x\ ti\t|\t| \tj -->\t')
9 for i=0:6
10   printf( ' %i\t',i)
11 end
12 printf( '\n |\t| \t|\t| \tt -->\t')
13 for i=0:6
14   printf( '%.3f\t',i*dt)

```

```

15 end
16 printf( '\n


---


    ')
17 for j=1:6
18     printf( '\n %.1f\t%.\t| \t\t', (j-1)*dx , j-1)
19     for i=1:7
20         if i==1 then
21             u(j,i)=0;
22         elseif j==1 then
23             u(j,i)=1
24         else
25             u(j,i)=(1/(1+r))*u(j,i-1)+r*u(j-1,i)/(1+
26             r)
27         end
28         printf( '%.3f\t', u(j,i))
29     end


---


29 end

```

Scilab code Exa 19.5 Lax Wendroff Method

```

1 //Example 19.5
2 //Lax Wendroff Method
3 //Page no. 660
4 clc;clear;close;
5
6 c=-2;dt=0.07;dx=0.2;
7 r=abs(c)*dt/dx;
8 printf( '\n x\ti\t|\t\t-->\t')
9 for i=0:6
10     printf( ' %i\t', i)
11 end
12 printf( '\n |\t|\t|\t\t-->\t')

```

```

13 for i=0:6
14     printf ('%.3f\t',i*dt)
15 end
16 i=1;
17 printf ('\n
18 ')
18 for j=1:7
19     for i=1:6
20         if j==1 then
21             u(i,j)=0;
22             u(i+1,j)=0;
23         elseif i==1 then
24             u(i,j)=1
25         else
26             u(i,j)=r*(r-1)*u(i+1,j-1)/2+(1-r^2)*u(i,
27                 j-1)+r*(1+r)*u(i-1,j-1)/2
28         end
29     end
30 end
31 for i=1:6
32     printf ('\n %.1f\t|i\t|\t| t |', (i-1)*dx,i-1)
33     for j=1:7
34         printf ('%.3f\t',u(i,j))
35     end

```

Scilab code Exa 19.6 Wendroff Method

```

1 //Example 19.6
2 //Wendroff Method
3 //Page no. 661
4 clc;clear;close;

```

```

5
6 c=2;k=0.07;h=0.2;
7 a=(h+k*c)/(h-k*c)
8 printf ('\n  x\ t i\ t |\t j -->\t ')
9 for i=0:6
10   printf ('%i\t',i)
11 end
12 printf ('\n  |\t |\t |\t t -->\t ')
13 for i=0:6
14   printf ('%.3f\t',i*k)
15 end
16 printf ('\n
')
17 for i=1:6
18   printf ('\n %.1f\t%i\t|\t\t',(i-1)*h,i-1)
19   for j=1:7
20     if j==1 then
21       u(i,j)=0;
22     elseif i==1 then
23       u(i,j)=1
24     else
25       u(i,j)=u(i-1,j-1)+(u(i,j-1)-u(i-1,j))/a
26     end
27     printf ('%.3f\t',u(i,j))
28   end
29 end

```

Scilab code Exa 19.7 Leapfrog Method

```

1 //Example 19.7
2 //Leapfrog Method
3 //Page no. 662

```

```

4 clc;clear;close;
5
6 c=2;k=0.07;h=0.2;
7 r=c*k/h
8 printf( '\n  x\ t i \t | \t j -->\t ')
9 for i=0:6
10   printf(' %i\t',i)
11 end
12 printf( '\n  |\t | \t | \t t -->\t ')
13 for i=0:6
14   printf('%.3f\t',i*k)
15 end
16 printf( '\n
')
17
18 for j=1:7
19   for i=1:6
20     if j==1 | j==2 & i~=1 then
21       u(i,j)=0;
22       u(i+1,j)=0;
23     elseif i==1 then
24       u(i,j)=1
25     else
26       u(i,j)=u(i,j-2)-r*(u(i+1,j-1)-u(i-1,j-1))
27     end
28   end
29 end
30 for i=1:6
31   printf( '\n %.1f\t%i\t|\t\t', (i-1)*h, i-1)
32   for j=1:7
33     printf('%.3f\t',u(i,j))
34   end
35 end

```

Scilab code Exa 19.8 Variable Coefficients

```
1 //Example 19.8
2 //Variable Coefficients
3 //Page no. 663
4 clc;clear;close;
5
6 //simple explicit method
7 printf ('\n\nBy Simple Explicit Method:\n\n')
8 dt=0.05;dx=0.2;
9 x=0;
10 printf (' \n i \t x\t r\t|\t j -->\t ')
11 for i=0:6
12     printf (' %i\t',i)
13 end
14 printf (' \n \t\t|\t tt -->\t ')
15 for i=0:6
16     printf ('%.3f\t',i*dt)
17 end
18 printf ('\n
_____
')
19 for j=1:6
20     r=sqrt (1+2*x)*dt/dx;
21     printf (' \n %i\t%.3f\t%.3f\t|\t', (j-1),x,r)
22     for i=1:7
23         if i==1 then
24             u(j,i)=0;
25         elseif j==1 then
26             u(j,i)=1
27         else
28             u(j,i)=(1-r)*u(j,i-1)+r*u(j-1,i-1)
```

```

29         end
30         printf( '%.3f\t', u(j,i))
31
32     end
33     x=x+dx
34 end
35
36
37 // simple implicit method
38 printf( '\n\n\n By Simple Implicit Method:\n')
39 c=-2; dt=0.05; dx=0.2; x=0
40 printf( '\n i\t x\t r\t | \tj -->\t')
41 for i=0:6
42     printf( '%i\t', i)
43 end
44 printf( '\n \t\t\t | \tt -->\t')
45 for i=0:6
46     printf( '%.3f\t', i*dt)
47 end
48 printf( '\n
')
49 for j=1:6
50     r=sqrt(1+2*x)*dt/dx;
51     printf( '\n %i\t%.3f\t%.3f\t| \t\t\t', (j-1), x, r)
52     for i=1:7
53         if i==1 then
54             u(j,i)=0;
55         elseif j==1 then
56             u(j,i)=1
57         else
58             u(j,i)=(1/(1+r))*u(j,i-1)+r*u(j-1,i)/(1+
59             r)
60         end
61         printf( '%.3f\t', u(j,i))
62     end
63     x=x+dx
64 end

```

```

64
65
66 // wendroff method
67 printf( '\n\n\nBy Wendroff Method:\n' )
68 k=0.05; h=0.2;
69 x=0.1;
70 printf( '\n i\t x\t c\t a\t |\t j -->\t' )
71 for i=0:6
72     printf( '%i\t', i)
73 end
74 printf( '\n \t\t\t|\t t -->\t' )
75 for i=0:6
76     printf( '%.3f\t', i*k)
77 end
78 printf( '\n
_____
')
79 for i=1:6
80     c=sqrt(1+2*x);
81     a=(h+k*c)/(h-k*c)
82     printf( '\n %i\t%.3f\t%.3f\t%.3f\t|\t t\t', (i-1), x-
83         h/2, c, a)
84     for j=1:7
85         if j==1 then
86             u(i,j)=0;
87             u(i+1,j)=0;
88         elseif i==1 then
89             u(i,j)=1
90         else
91             u(i,j)=u(i-1,j-1)+(u(i,j-1)-u(i-1,j))/a
92         end
93         printf( '%.3f\t', u(i,j))
94     end
95     x=x+h
96 end

```

Scilab code Exa 19.9 Inhomogeneous 1st Order Hyperboolic Differential Equation

```
1 //Example 19.9
2 //Inhomogeneous 1st Order Hyperboolic Differential
   Equation
3 //Page no. 665
4 clc;clear;close;
5
6 //simple explicit method
7 printf('n\nBy Simple Explicit Method:\n')
8 c=-2;dt=0.07;dx=0.2;
9 r=abs(c)*dt/dx;
10 printf('n i\tx\t|\tj -->\t')
11 for i=0:6
12     printf(' %i\t',i)
13 end
14 printf('n |\t|\t|\t\t -->\t')
15 for i=0:6
16     printf('%.3f\t',i*dt)
17 end
18 printf('n
_____
')
19 x=0;
20 for j=1:6
21     printf('n %i\t%.1f\t|\t',j-1,x)
22     for i=1:7
23         if i==1 then
24             u(j,i)=exp(-x);
25         elseif j==1 then
26             u(j,i)=1
27         else
```

```

28         u(j,i)=(1-r)*u(j,i-1)+r*u(j-1,i-1)+dt*2*
29             x
30         end
31     printf( '%.3f\t',u(j,i))
32   end
33 x=x+dx
34
35
36 //simple implicit method
37 printf( '\n\n\nBy Simple Implicit Method:\n')
38 c=-2; dt=0.07; dx=0.2;
39 r=abs(c)*dt/dx;
40 printf( '\n i\tx\t|\tj -->\t')
41 for i=0:6
42   printf( ' %i\t',i)
43 end
44 printf( '\n |\t|\t|\tt -->\t')
45 for i=0:6
46   printf( '%.3f\t',i*dt)
47 end
48 printf( '\n
49 x=0;
50 for j=1:6
51   printf( '\n %i\t%.1f\t|\t',j-1,x)
52   for i=1:7
53     if i==1 then
54       u(j,i)=exp(-x);
55     elseif j==1 then
56       u(j,i)=1
57     else
58       u(j,i)=(1/(1+r))*u(j,i-1)+r*u(j-1,i)/(1+
59           r)+dt*2*x
60     end
61     printf( '%.3f\t',u(j,i))
62   end

```

```

62      x=x+dx
63  end
64
65
66 // wendroff method
67 printf( '\n\n\nBy Wendroff Method:\n')
68 c=2; k=0.07; h=0.2;
69 a=(h+k*c)/(h-k*c)
70 printf( '\n  x\ ti\ t|\ tj -->\t ')
71 for i=0:6
72     printf( '%i\t',i)
73 end
74 printf( '\n  |\t|\t|\tt -->\t ')
75 for i=0:6
76     printf( '%.3f\t',i*k)
77 end
78 printf( '\n
')
79 x=0;
80 for i=1:6
81     printf( '\n %.1f\t%i\t|\t',x,i-1)
82     for j=1:7
83         if j==1 then
84             u(i,j)=exp(-x);
85         elseif i==1 then
86             u(i,j)=1
87         else
88             u(i,j)=u(i-1,j-1)+(u(i,j-1)-u(i-1,j))/a
89             +(2*h*k)*(x+h/2)/(a*(h+c*k))
90         end
91         printf( '%.3f\t',u(i,j))
92     end
93     x=x+h
94 end

```

Scilab code Exa 19.10 Non Linear 1st Order Hyperboolic Differential Equation

```
1 //Example 19.10
2 //Non Linear 1st Order Hyperboolic Differential
   Equation
3 //Page no. 667
4 clc;clear;close;
5
6 c=-2;k=0.05;h=0.2;
7 r=abs(c)*k/h;
8 printf ('\n i\t x\t |\t j -->\t ')
9 for i=0:6
10    printf ('%i\t',i)
11 end
12 printf ('\n |\t |\t |\t tt -->\t ')
13 for i=0:6
14    printf ('%.3f\t',i*k)
15 end
16 i=1;
17 x=0;
18 printf ('\n
')
19 for j=1:7
20    for i=1:6
21       if j==1 then
22          u(i,j)=exp(-x);
23          u(i+1,j)=exp(-(x+h));
24       elseif i==1 then
25          u(i,j)=1
26       else
27          u(i,j)=u(i,j-1)-k*(u(i+1,j-1)^2-u(i-1,j)
```

```

-1)^2)/(4*h)+k^2*((u(i+1,j-1)+u(i,j)
-1))*(u(i+1,j-1)^2-u(i,j-1)^2)-(u(i,j
-1)+u(i-1,j-1))*(u(i,j-1)^2-u(i-1,j
-1)^2))/(8*h^2)

28     end
29     x=x+h
30     end
31 end
32 x=0;
33 for i=1:6
34     printf ('\n %i\t%.1f\t|\t',i-1,x)
35     for j=1:7
36         printf ('%.3f\t',u(i,j))
37     end
38     x=x+h
39 end

```

Scilab code Exa 19.11 Finite Difference Method

```

1 //Example 19.11
2 //Finite Difference Method
3 //Page no. 670
4 clc;clear;close;
5 deff( 'y=f(x) ', 'y=sin (%pi*x) ')
6 deff( 'y=g(x) ', 'y=0')
7 a=1;b=1;c=1;n=5;m=10;
8 h=a/n;k=b/m;r=c*k/h;
9 r1=r^2;r2=r1/2;s1=1-r1;s2=2*(1-r2)
10 printf ('\n i ')
11 for i=1:n
12     printf ('\t %i ',i)
13 end
14 printf ('\n

```

```

    _____\

n fi ')
15 for i=1:n
16     f1(i)=f(h*(i-1))
17     printf (' \t% .3 f ',f1(i))
18 end
19 printf (' \n g i ')
20 for i=1:n
21     g1(i)=g(h*(i-1))
22     printf (' \t %g ',g1(i))
23 end
24 printf (' \n \n \n i / j --> ')
25 for i=1:m
26     printf (' \t %i ',i)
27 end
28 printf (' \n

')
29 for j=1:m
30     for i=1:n
31         if i==1 | i==n then
32             u(i,j)=0;
33         elseif j==1
34             u(i,j)=f1(i)
35         elseif j==2
36             u(i,j)=s1*f1(i)+k*g1(i)+r2*(f1(i+1)+f1(i
-1))
37         else
38             u(i,j)=s2*u(i,j-1)+r1*u(i-1,j-1)+u(i+1,j
-1)-u(i,j-2)
39         end
40     end
41 end
42 end
43 for i=1:n
44     printf (' \n %i \t ',i)
45     for j=1:m
46         printf (' \t% .3 f ',u(i,j))

```

```
47      end  
48 end
```

Scilab code Exa 19.12 Hyperbolic Partial Differential Equations

```
1 //Example 19.12  
2 //Hyperbolic Partial Differential Equations  
3 //Page no. 673  
4 clc;clear;close;  
5 deff( 'y=f(x)', 'y=12*x' )  
6 Ua(1)=0.25;  
7 Ua(2)=0.75  
8 A=[1,-2;1,2];  
9 x1=inv(A)*Ua;  
10 printf('Xb = %g and Tb = %g',x1(1),x1(2))  
11 A=[2,-1;2,1];  
12 B=[-7.5;-8.5];  
13 x2=inv(A)*B;  
14 printf('\n\n Pb = %g and Qb = %g',x2(1),x2(2))  
15 x1(1)=x1(1)-Ua(1)  
16 du=x1'*x2  
17 printf('\n\n dU = %g',du)  
18 Ub=f(Ua(1))+du;  
19 printf('\n\n Modified Ub = %g',Ub)
```

Scilab code Exa 19.13 Hyperbolic Differential Equations in 2D or 3D

```
1 //Example 19.13  
2 //Hyperbolic Differential Equations in 2D or 3D
```

```

3 //Page no. 675
4 clc;clear;close;
5
6 def(f,'y=f(x,y)', 'y=x*(2-x)*y*(2-y)')
7 c2=3;k=0.4;h=0.4;c2=3;s2=0.5
8 for l=0:11
9   if l==0 then
10     printf('\n t = %i\n n i \t x\t|\t j -->\t',l)
11   for i=0:5
12     printf(' %i\t',i)
13   end
14   printf('\n |\t |\t|\t y -->\t')
15   for i=0:5
16     printf('%.3f\t',i*k)
17   end
18   x=0;
19   printf('\n
20
21   for i=1:6
22     y=0;
23     printf('\n %i\t%.3f\t|\t|\t|\t',i-1,x)
24     for j=1:6
25       if i==1 | i==6 then
26         u(i,j)=0;
27       elseif j==1 | j==6 then
28         u(i,j)=0
29       else
30         u(i,j)=f(x,y)
31       end
32       printf('%.3f\t',u(i,j))
33     y=y+k;
34   end
35 end
36 u2=u;
37 else
38   printf('\n\n t = %i\n n i \t x\t|\t j -->\t',l)

```

```

39      for i=0:5
40          printf( '%i\t',i)
41      end
42      printf( '\n |\t | \t| \ty -->\t ')
43      for i=0:5
44          printf( '%.3f\t',i*k)
45      end
46      x=0;
47      printf( '\n
48          ')
48      for i=1:6
49          y=0;
50          printf( '\n %i\t%.3f\t|\t\t',i-1,x)
51          for j=1:6
52              if i==1 | i==6 then
53                  u(i,j)=0;
54              elseif j==1 | j==6 then
55                  u(i,j)=0
56              elseif l==1
57                  u(i,j)=s2*(u1(i+1,j)+u1(i-1,j)+u1(i,
58                                  j+1)+u1(i,j-1)-4*u1(i,j))+2*u1(i,
59                                  j)
60              else
61                  u(i,j)=s2*(u1(i+1,j)+u1(i-1,j)+u1(i,
62                                  j+1)+u1(i,j-1)-4*u1(i,j))+2*u1(i,
63                                  j)-u2(i,j)
64              end
65              printf( '%.4f\t',u(i,j))
66          end
67          y=y+k;
68      end
69      x=x+h
70      end
71  if l>1 then
72      u2=u1
73  end
74  u1=u;

```

71 **end**

Chapter 20

Numerical Solutions of Elliptical Partial Differential Equations

Scilab code Exa 20.1 Direct Method

```
1 //Example 20.1
2 //Direct Method
3 //Page no. 682
4 clc;clear;close;
5 h=1/3;
6 A=[-4,1,1,0;1,-4,0,1;1,0,-4,1;0,1,1,-4]
7 x=0;
8 for i=1:4
9     x=x+h
10    if i==4 then
11        B(i,1)=0
12    else
13        B(i,1)=-1*sin(x*pi)^2
14    end
15 end
```

```
16 disp(A, 'A =')
17 disp(B, 'B =')
18 U=inv(A)*B
19 disp(U, 'U =')
```

Scilab code Exa 20.2 Five Point Formula

```
1 //Example 20.2
2 //Five Point Formula
3 //Page no. 683
4 clc;clear;close;
5
6 A=[-4,1,1,0;1,0,-4,1;1,-4,0,1;0,1,1,-4];
7 B=[-25;-150;-25;-150];
8 u1=inv(A)*B;
9 j=0;k=1
10 for i=1:4
11     j=j+1;
12     printf('nu%i%i = %g\n',k,j,u1(i))
13     if i==2 then
14         j=0;k=2
15     end
16 end
17 printf('\n\n U = \n')
18 for i=1:4
19     printf('\n')
20     for j=1:4
21         if j==1 then
22             u(i,j)=0
23         elseif j==4
24             u(i,j)=100
25         elseif (i==1 | i==4) & j==2
26             u(i,j)=25
```

```

27     elseif i==1 | i==4
28         u(i,j)=u(i,j-1)*2
29     else
30         u(i,j)=u1((j-1)*2-i+2)
31     end
32     printf( '\t%g\t' ,u(i,j))
33 end
34 end

```

Scilab code Exa 20.3 Finite Difference Method

```

1 //Example 20.3
2 //Finite Difference Method
3 //Page no. 685
4 clc;clear;close;
5
6 printf('Itr\t\t U11\t\t U21\t\t U12\t\t U22\n

```

```

n')
7 for i=1:4
8     for j=1:4
9         if j==1 then
10             u(i,j)=0
11         elseif j==4
12             u(i,j)=100
13         elseif (i==1 | i==4) & j==2
14             u(i,j)=25
15         elseif i==1 | i==4
16             u(i,j)=u(i,j-1)*2
17         else
18             u(i,j)=0
19         end
20     end

```

```

21 end
22 for k=0:17
23     printf(' %i\t\t%.3f\t%.3f\t%.3f\t%.3f\n',
24         k,u(3,2),u(3,3),u(2,2),u(2,3))
25     for i=3:-1:2
26         for j=2:3
27             u1(i,j)=(u(i,j+1)+u(i,j-1)+u(i-1,j)+u(i
28                 +1,j))/4
29         end
30     end
31     for i=3:-1:2
32         for j=2:3
33             u(i,j)=u1(i,j)
34         end
35     end
36 disp(u,'U = ')

```

Scilab code Exa 20.4 Seven Point Formula

```

1 //Example 20.4
2 //Seven Point Formula
3 //Page no. 686
4 clc;clear;close;
5 printf('Itr\t\t U111\t\t U211\t\t U121\t\t U221\n

```

```

n')
6 for i=1:4
7     for j=1:4
8         for k=3:-1:1
9             if k==3 then
10                 u(i,j,k)=100
11             elseif (i==1 | i==4 | j==1 | j==4) & k

```

```

                ==2
12             u(i,j,k)=300
13         elseif k==2
14             u(i,j,k)=0
15         elseif (i==1 | i==4 | j==1 | j==4) & k
16             ==1
17                 u(i,j,k)=500
18             else
19                 u(i,j,k)=700
20             end
21         end
22     end
23 k=2
24 for l=0:14
25     printf(' %i\t\t%.3f\t\t%.3f\t\t%.3f\t\t%.3f\n',
26         l,u(3,2,2),u(3,3,2),u(2,2,2),u(2,3,2))
27     for i=3:-1:2
28         for j=2:3
29             u1(i,j)=(u(i,j+1,k)+u(i,j-1,k)+u(i-1,j,k)
30             )+u(i+1,j,k)+u(i,j,k+1)+u(i,j,k-1))/6
31         end
32     end
33     for i=3:-1:2
34         for j=2:3
35             u(i,j,2)=u1(i,j)
36         end

```

Scilab code Exa 20.5 Nine Point Formula

1 //Example 20.5

```

2 //Nine Point Formula
3 //Page no. 688
4 clc;clear;close;
5
6 printf('I tr\t\t U11\t\t U12\t\t U21\t\t U22\n
          n')
7 for i=1:4
8     for j=1:4
9         if j==1 then
10             u(i,j)=0
11         elseif j==4
12             u(i,j)=100
13         elseif (i==1 | i==4) & j==2
14             u(i,j)=25
15         elseif i==1 | i==4
16             u(i,j)=u(i,j-1)*2
17         else
18             u(i,j)=0
19         end
20     end
21 end
22 for k=0:17
23     printf(' %i\t%.3f\t%.3f\t%.3f\t%.3f\n',
24         k,u(3,2),u(2,2),u(3,3),u(2,3))
25     for i=3:-1:2
26         for j=2:3
27             u1(i,j)=(u(i+1,j-1)+u(i-1,j-1)+u(i+1,j
28                 +1)+u(i-1,j+1)+4*(u(i,j+1)+u(i,j-1)+u
29                     (i-1,j)+u(i+1,j)))/20
30         end
31     for i=3:-1:2
32         for j=2:3
33             u(i,j)=u1(i,j)
34         end
35     end

```

```
35 disp(u, 'The Solution of the System is = ')
```

Scilab code Exa 20.6 Five Point Formula

```
1 //Example 20.6
2 //Five Point Formula
3 //Page no. 689
4 clc;clear;close;
5
6 h=0.25;k=0.25;y=1;x=0;
7 deff( 'x=f(y)', 'x=y^3' )
8
9 for i=1:5
10    x=0;
11    printf( '\n%g\t| ',y)
12    for j=1:5
13        if (i==1 | i==5)
14            u(i,j)=f(x)
15        elseif j==5
16            u(i,j)=f(x)
17        else
18            u(i,j)=0
19        end
20        x=x+k;
21        printf( '%f\t',u(i,j))
22    end
23    y=y-h
24 end
25 printf( '\n\t
26 n')
27 x=0;
28 for j=1:5
```

```

28         printf( '\t    %g\t',x)
29         x=x+k
30     end
31 printf( '\n\n\n  Itr\t U11\t U12\t U13\t U21\t U22\t
32                                     U23\t U31\t U32\t U33\n
33
34
35         printf( '  %i\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f
36             \t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f
37             \t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f
38             \t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f
39             u1(i,j)=(u(i,j+1)+u(i,j-1)+u(i-1,j)+u(i
40                 +1,j)-h^2*y)/4
41         end
42     end
43     for i=4:-1:2
44         for j=2:4
45             u(i,j)=u1(i,j)
46         end
47     end

```

Scilab code Exa 20.7 Laplace Distribution

```

1 //Example 20.7
2 //Laplace Distribution
3 //Page no. 694
4 clc;clear;close;

```

```

5
6 dr=3;r0=4;dth=%pi/4;
7 def ('y=f(u1,u2,u3,u4)', 'y=(u1+u3+(dr*(u3-u1))/(2*r0
     )+(u2+u4)*(dr/(r0*dth))^2)/(2*(1+(dr/(r0*dth))^2)
     ') //laplace distribution
8 for i=1:8
9     U(i)=0;
10 end
11 printf('I tr\t U1\t U2\t U3\t U4\t U5\t U6\t
   U7\t U8\n
   ')
12 for l=0:15
13     printf('\n%i',l)
14     for i=1:8
15         if i==1 then
16             u1(i)=f(100,U(8),40,U(i+1))
17         elseif i==8
18             u1(i)=f(100,U(i-1),40,U(1))
19         else
20             u1(i)=f(100,U(i-1),40,U(i+1))
21         end
22
23     end
24     for i=1:8
25         U(i)=u1(i)
26         printf('\t%.3f',U(i))
27     end
28 end

```

Scilab code Exa 20.8 Spherical Coordinate System

1 //Example 20.8

```

2 //Spherical Coordinate System
3 //Page no. 697
4 clc;clear;close;
5 deff( 'y=cot(x)', 'y=1/tan(x)')
6 dr=5;r0=50;dth=%pi/4;dfi=%pi/4;N=-10;Z=60;Nb=0;Zt
    =70;
7 deff( 'y=f(u1,u2,u3,u4,u5,u6,th0)', 'y=((u1+u3)/dr^2+(u3-u1)/(r0*dr)+(u2+u4)/(r0*dth)^2+(u2*cot(th0)/(r0^2*dth)+(u5+u6)/(r0*sin(th0)*dfi)^2))/(2/dr^2+2/(r0*dth)^2+cot(th0)/(r0^2*dth)+2/(r0*sin(th0)*dfi)^2)' )           //laplace distribution in spherical coordinate
8 T1=40;T2=20;H1=35;H2=10;B1=5;B2=0;t1=240;t2=180;b1
    =100;b2=80;h1=210;h2=150
9 printf( '\n
_____
n')
10 s=["T","H","B","t","h","b"];
11 for i=1:8
12     if i<4 | i>6 then
13         T(1,i)=T1;
14         H(1,i)=H1;
15         B(1,i)=B1;
16         b(1,i)=b1;
17         t(1,i)=t1;
18         h(1,i)=h1;
19     else
20         T(1,i)=T2;
21         H(1,i)=H2;
22         b(1,i)=b2;
23         B(1,i)=B2;
24         t(1,i)=t2;
25         h(1,i)=h2;
26     end
27 end
28 h(1)=0;h(2)=0;
29 A1=[T;H;B;t;h;b]
30 for i=1:6

```

```

31     if i==1 then
32         printf('Temperature Distribution in Outer
33             Sphere\n')
34         printf('
35             _____
36             n')
37         end
38         if i==4 then
39             printf('\nTemperature Distribution in Inner
40                 Sphere\n')
41             printf('
42                 _____
43                 n')
44             end
45             printf('\nPoint : ')
46             for j=1:8
47                 printf('\t%s%i',s(i),j)
48             end
49             printf('\nTemperature : ')
50             for j=1:8
51                 if (j==1 | j==2) & i==5 then
52                     printf('\t%s,"?")'
53                 else
54                     printf('\t%i',A1(i,j))
55                 end
56             end
57             printf('\n
58         )
59     end
60     th0=10^-30
61     Uh1=f(1000,A1(5,8),A1(2,1),A1(5,2),A1(6,1),A1(4,1),
62         th0)
63     disp(Uh1,'Uh1 = ')
64     th0=%pi/4;
65     Uh2=f(1000,Uh1,A1(2,2),A1(5,3),A1(6,2),A1(4,2),th0)
66     disp(Uh2,'Uh2 = ')

```

Chapter 21

Advances in Numerical Methods Using Parallel Computing Paradigm

Scilab code Exa 21.1 Parallel Bisection Method

```
1 //Example 21.1
2 //Parallel Bisection Method
3 //Page no. 721
```

ltr	a	b	h	x0 y0	x1 y1	x2 y2	x3 y3	x4 y4	x5 y5
1	0	1	0.2	0.000000 -1.000000	0.200000 -0.940067	0.400000 -0.761061	0.600000 -0.465336	0.800000 -0.056707	1.000000 0.459698
2	0.8	1	0.04	0.800000 -0.056707	0.840000 0.038137	0.880000 0.137249	0.920000 0.240580	0.960000 0.348080	1.000000 0.459698
3	0.8	0.84	0.008	0.800000 -0.056707	0.808000 -0.038082	0.816000 -0.019284	0.824000 -0.000315	0.832000 0.018625	0.840000 0.038137
4	0.824	0.832	0.002	0.824000 -0.000315	0.825600 0.003499	0.827200 0.007321	0.828800 0.011149	0.830400 0.014984	0.832000 0.018825
5	0.824	0.8256	0.0003	0.824000 -0.000315	0.824320 0.000447	0.824640 0.001210	0.824960 0.001973	0.825280 0.002736	0.825600 0.003499
6	0.824	0.82432	6e-005	0.824000 -0.000315	0.824064 -0.000163	0.824128 -0.000010	0.824192 0.000142	0.824256 0.000295	0.824320 0.000447

Figure 21.1: Parallel Bisection Method

Scilab code Exa 21.2 Lagrange Interpolation in Parallel Computing

```
1 //Example 21.2
2 //Lagrange Interpolation in Parallel Computing
3 //Page no. 723
4 clc;close;clear;
5
6 xi=[-1,0,2,5];
7 yi=[9,5,3,15];
8 s= ["x=1", "n=4", "Data:", "(-1,9)", "(0,5)", "(2,3)", "(5,15)"]
9 for i=1:4
10     printf(' \tProcessor\t ')
11 end
12 printf(' \n')
13 for i=1:4
14     printf(' \t    N%ii\t \t ',i)
15 end
16 printf(' \n')
17 for i=1:7
18     for j=1:4
19         printf('           %s\t \t ',s(i))
20     end
21     printf(' \n')
22 end
23
24 x=1; T=0;
25 for k=0:3
26     p=yi(k+1)
27     for j=0:3
28         if(j ~= k)
29             p=p*((x-xi(j+1))/(xi(k+1)-xi(j+1)))
```

```

30         end
31     end
32     T=T+p;
33     printf( '\nT(%i) = %g',k+1,p)
34 end
35 printf( '\n\nT = %g',T)

```

Scilab code Exa 21.3 Trapezoidal Rule and Simpsons Rule in Parallel Computing

```

1 //Example 21.3
2 //Trapezoidal Rule and Simpsons Rule in Parallel
   Computing
3 //Page no. 726
4 clc;close;clear;
5 n=8;a=0;b=8;
6 h=(b-a)/n
7 def(f,'y=f(x)', 'y=1/(1+x)')
8 for i=0:8
9     x(i+1)=i;
10    y(i+1)=f(x(i+1))
11 end
12 printf('xi\t')
13 for i=1:9
14     printf('%i\t',x(i))
15 end
16 printf('\n yi\t')
17 for i=1:9
18     printf('1/%i\t',i)
19 end
20
21 // trapezoidal rule
22 S=0;
23 for i=1:9

```

```

24     if(i==1 | i==9)
25         S=S+y(i)
26     else
27         S=S+2*y(i)
28     end
29 end
30 S=S*h/2
31 printf ('\n\nTrapezoidal Rule Sum = %g',S)
32
33 //Simpsons 1/3rd Rule
34 S=0;
35 for i=1:9
36     if(i==1 | i==9)
37         S=S+y(i)
38     elseif(((i)/2)-fix((i)/2)==0)
39         S=S+4*y(i)
40     else
41         S=S+2*y(i)
42     end
43 end
44 S=S*h/3
45 printf ('\n\nSimpsons 1/3rd Rule Sum = %g',S)

```

Scilab code Exa 21.4 Parallel Gauss Seidel Method

```

1 //Example 21.4
2 //Parallel Gauss-Seidel Method
3 //Page no. 730
4 clc;close;clear;
5
6 A=[3,2;6,2];
7 B=[2;3];
8 x(1)=1/4;

```

```

9 x(2)=1/5;
10 e=0.002;
11 old(1)=x(1);
12 old(2)=x(2);
13 new(1)=old(1);
14 new(2)=old(2);
15 printf ('\t\tProcess 1\t\tProcess 2\n Itr\t\told\
    tnew1\t\told2\tnew2\n\n')
16 printf ('%i\t\t%g\t%g\t\t%g\t\t%g\n',0,old(1),new(1),
    old(2),new(2))
17 for i=1:4
18     printf ('%i',i)
19     for j=1:2
20         k=0;
21         for l=1:j-1
22             k=k-(A(j,l)*old(l));
23         end
24         m=0;
25         for l=j+1:2
26             m=m-(A(j,l)*old(l));
27         end
28         new(j)=(B(j)+k+m)/A(j,j)
29         printf ('\t\t%.5g\t%.5g',old(j),new(j))
30     end
31     printf ('\n')
32     old(1)=new(1)
33     old(2)=new(2)
34 end

```

Scilab code Exa 21.5 Poissons Partial Differential Equation

```

1 //Example 21.5
2 //Poissons Partial Differential Equation

```

```

3 //Page no. 733
4 clc;clear;close;
5
6 s=[”st”, ”nd”, ”rd”]
7 for i=4:20
8     s(i)=”th”
9 end
10 h=0.25; deff( ’y=f(x)’ , ’y=x^3’); y=1; x=0;
11 for i=1:6
12
13
14 if i~=6 then
15     printf( ’%g\t|’ , y)
16     y=y-h;
17     x=0;
18     for j=1:5
19         if i==1 | i==5 | j==5 then
20             P(i,j)=f(x)
21         else
22             P(i,j)=0
23         end
24         printf( ’%f\t’ , P(i,j))
25         x=x+h;
26     end
27 else
28     printf( ’
29             n\t’ )
30     x=0;
31     for j=1:5
32         printf( ’      %g\t\t’ , x)
33         x=x+h
34     end
35     printf( ’\n’ )
36 end
37
38 printf( ’\n\n\n’ )

```

```

39
40 for l=0:17
41     y=1;
42     if l~=0 then
43         printf('After the %i%s Iteration : \n
44             n  %i',l,s(l),1)
45     for i=1:6
46         if i~=6 then
47             printf('\t%g',y)
48             y=y-h
49             for j=1:5
50                 printf('\t%.3f',P(i,j))
51             end
52         else
53             x=0;
54             printf('\t')
55             for j=1:5
56                 printf('\t%g',x)
57                 x=x+h
58             end
59             printf('\n')
60     end
61     printf(
62             n )
63 end
64 y=0;
65 for i=4:-1:2
66     y=y+h
67     for j=2:4
68         P1(i,j)=(P(i,j+1)+P(i,j-1)+P(i-1,j)+P(i
69             +1,j)-h^2*y)/4
70     end
71     for i=4:-1:2
72         for j=2:4

```

```
72          P(i,j)=P1(i,j)
73      end
74  end
75 end
```

Chapter 22

Numerical Methods Using Neural Networks

Scilab code Exa 22.1 MLP Algorithm

```
1 //Example 22.1
2 //MLP Algorithm
3 //Page no. 748
4 clc;clear;close;
5 deff( 'y=f(x)' , 'y=1/(1+exp(-x))' )
6 Wih=[0.1,-0.3;0.3,0.4];
7 Who=[0.4;0.5]
8 i=[0.2,0.6];
9 t=0.7;
10 a=10;
11 for k=1:3
12     printf( '\n\n\nAfter Iteration %i :\n\n' ,k)
13     disp(Wih, 'Wih = ')
14     disp(Who, 'Who = ')
15 a1=i*Wih;
16 disp(a1, 'a = ')
17 h=[f(a1(1)),f(a1(2))]
```

```

18 disp(h, 'h = ')
19 b1=h*Who
20 disp(b1, 'b1 =')
21 o=f(b1)
22 disp(o, 'o = ')
23 d=o*(1-o)*(t-o)
24 disp(d, 'd =')
25 for j=1:2
26     e(1,j)=h(j)*(1-h(j))*d*Who(j)
27 end
28 disp(e, 'e =')
29 dWho=a*h'*d;
30 disp(dWho, 'dWho =')
31 Who=Who+dWho;
32 dWih=a*i'*e;
33 disp(dWih, 'dWih =')
34 Wih=Wih+dWih;
35 end

```

Scilab code Exa 22.2 MLP

```

1 //Example 22.2
2 //MLP
3 //Page no. 758
4 clc;clear;close;
5
6 deff('y=f(x)', 'y=x^3-x^2+x-1')
7 printf('Input\t\Desired\t\Network\t>Error\n\t\t
        Output\t\t Output\n
        \n')
8 in
      =[0.7572, 0.7601, 0.7620, 1.4831, 1.4874, 1.4900, 2.0913, 2.0934, 2.1006,];

```

```

9 n
= [-0.3941, -0.3896, -0.3867, 1.6054, 1.6259, 1.6391, 5.8762, 5.8969, 5.96

10 for i=1:18
11     printf(' %.4f\t%.4f\t%.4f\t%.4f\n', in(i), f
12         (in(i)), n(i), n(i)-f(in(i)))
13 end

```

Scilab code Exa 22.3 Bisection Method

```

1 //Example 22.3
2 //Bisection Method
3 //Page no. 764
4 clc;clear;close;
5
6 def(f, 'y=f(x)', 'y=x^3-x^2+x-1')
7 printf('N01\tn02\tn11\tn12\tn21\tnet31\tO31\tN41\
8 tN42\n
n')
8 N01
=[0, 1, 0.5, 0.75, 0.875, 0.938, 0.969, 0.984, 0.992, 0.996, 0.998, 0.999, 1, 1.031, 1.062, 1.093, 1.124, 1.155, 1.186, 1.217, 1.248, 1.279, 1.31, 1.341, 1.372, 1.403, 1.434, 1.465, 1.496, 1.527, 1.558, 1.589, 1.62, 1.651, 1.682, 1.713, 1.744, 1.775, 1.806, 1.837, 1.868, 1.899, 1.93, 1.961, 1.992, 2.023, 2.054, 2.085, 2.116, 2.147, 2.178, 2.209, 2.24, 2.271, 2.302, 2.333, 2.364, 2.395, 2.426, 2.457, 2.488, 2.519, 2.55, 2.581, 2.612, 2.643, 2.674, 2.705, 2.736, 2.767, 2.798, 2.829, 2.86, 2.891, 2.922, 2.953, 2.984, 3.015, 3.046, 3.077, 3.108, 3.139, 3.17, 3.201, 3.232, 3.263, 3.294, 3.325, 3.356, 3.387, 3.418, 3.449, 3.48, 3.511, 3.542, 3.573, 3.604, 3.635, 3.666, 3.697, 3.728, 3.759, 3.79, 3.821, 3.852, 3.883, 3.914, 3.945, 3.976, 4.007, 4.038, 4.069, 4.1, 4.131, 4.162, 4.193, 4.224, 4.255, 4.286, 4.317, 4.348, 4.379, 4.41, 4.441, 4.472, 4.503, 4.534, 4.565, 4.596, 4.627, 4.658, 4.689, 4.72, 4.751, 4.782, 4.813, 4.844, 4.875, 4.906, 4.937, 4.968, 4.999, 5.03, 5.061, 5.092, 5.123, 5.154, 5.185, 5.216, 5.247, 5.278, 5.309, 5.34, 5.371, 5.402, 5.433, 5.464, 5.495, 5.526, 5.557, 5.588, 5.619, 5.65, 5.681, 5.712, 5.743, 5.774, 5.805, 5.836, 5.867, 5.898, 5.929, 5.96, 5.991, 6.022, 6.053, 6.084, 6.115, 6.146, 6.177, 6.208, 6.239, 6.27, 6.301, 6.332, 6.363, 6.394, 6.425, 6.456, 6.487, 6.518, 6.549, 6.58, 6.611, 6.642, 6.673, 6.704, 6.735, 6.766, 6.797, 6.828, 6.859, 6.89, 6.921, 6.952, 6.983, 7.014, 7.045, 7.076, 7.107, 7.138, 7.169, 7.2, 7.231, 7.262, 7.293, 7.324, 7.355, 7.386, 7.417, 7.448, 7.479, 7.51, 7.541, 7.572, 7.603, 7.634, 7.665, 7.696, 7.727, 7.758, 7.789, 7.82, 7.851, 7.882, 7.913, 7.944, 7.975, 8.006, 8.037, 8.068, 8.1, 8.131, 8.162, 8.193, 8.224, 8.255, 8.286, 8.317, 8.348, 8.379, 8.41, 8.441, 8.472, 8.503, 8.534, 8.565, 8.596, 8.627, 8.658, 8.689, 8.72, 8.751, 8.782, 8.813, 8.844, 8.875, 8.906, 8.937, 8.968, 9.0, 9.031, 9.062, 9.093, 9.124, 9.155, 9.186, 9.217, 9.248, 9.279, 9.31, 9.341, 9.372, 9.403, 9.434, 9.465, 9.496, 9.527, 9.558, 9.589, 9.62, 9.651, 9.682, 9.713, 9.744, 9.775, 9.806, 9.837, 9.868, 9.9, 9.939, 9.96, 9.981, 10.0, 10.021, 10.042, 10.063, 10.084, 10.105, 10.126, 10.147, 10.168, 10.189, 10.21, 10.231, 10.252, 10.273, 10.294, 10.315, 10.336, 10.357, 10.378, 10.399, 10.42, 10.441, 10.462, 10.483, 10.504, 10.525, 10.546, 10.567, 10.588, 10.609, 10.63, 10.651, 10.672, 10.693, 10.714, 10.735, 10.756, 10.777, 10.798, 10.819, 10.84, 10.861, 10.882, 10.903, 10.924, 10.945, 10.966, 10.987, 11.0, 11.021, 11.042, 11.063, 11.084, 11.105, 11.126, 11.147, 11.168, 11.189, 11.21, 11.231, 11.252, 11.273, 11.294, 11.315, 11.336, 11.357, 11.378, 11.399, 11.42, 11.441, 11.462, 11.483, 11.504, 11.525, 11.546, 11.567, 11.588, 11.609, 11.63, 11.651, 11.672, 11.693, 11.714, 11.735, 11.756, 11.777, 11.798, 11.819, 11.84, 11.861, 11.882, 11.903, 11.924, 11.945, 11.966, 11.987, 12.0, 12.021, 12.042, 12.063, 12.084, 12.105, 12.126, 12.147, 12.168, 12.189, 12.21, 12.231, 12.252, 12.273, 12.294, 12.315, 12.336, 12.357, 12.378, 12.399, 12.42, 12.441, 12.462, 12.483, 12.504, 12.525, 12.546, 12.567, 12.588, 12.609, 12.63, 12.651, 12.672, 12.693, 12.714, 12.735, 12.756, 12.777, 12.798, 12.819, 12.84, 12.861, 12.882, 12.903, 12.924, 12.945, 12.966, 12.987, 13.0, 13.021, 13.042, 13.063, 13.084, 13.105, 13.126, 13.147, 13.168, 13.189, 13.21, 13.231, 13.252, 13.273, 13.294, 13.315, 13.336, 13.357, 13.378, 13.399, 13.42, 13.441, 13.462, 13.483, 13.504, 13.525, 13.546, 13.567, 13.588, 13.609, 13.63, 13.651, 13.672, 13.693, 13.714, 13.735, 13.756, 13.777, 13.798, 13.819, 13.84, 13.861, 13.882, 13.903, 13.924, 13.945, 13.966, 13.987, 14.0, 14.021, 14.042, 14.063, 14.084, 14.105, 14.126, 14.147, 14.168, 14.189, 14.21, 14.231, 14.252, 14.273, 14.294, 14.315, 14.336, 14.357, 14.378, 14.399, 14.42, 14.441, 14.462, 14.483, 14.504, 14.525, 14.546, 14.567, 14.588, 14.609, 14.63, 14.651, 14.672, 14.693, 14.714, 14.735, 14.756, 14.777, 14.798, 14.819, 14.84, 14.861, 14.882, 14.903, 14.924, 14.945, 14.966, 14.987, 15.0, 15.021, 15.042, 15.063, 15.084, 15.105, 15.126, 15.147, 15.168, 15.189, 15.21, 15.231, 15.252, 15.273, 15.294, 15.315, 15.336, 15.357, 15.378, 15.399, 15.42, 15.441, 15.462, 15.483, 15.504, 15.525, 15.546, 15.567, 15.588, 15.609, 15.63, 15.651, 15.672, 15.693, 15.714, 15.735, 15.756, 15.777, 15.798, 15.819, 15.84, 15.861, 15.882, 15.903, 15.924, 15.945, 15.966, 15.987, 16.0, 16.021, 16.042, 16.063, 16.084, 16.105, 16.126, 16.147, 16.168, 16.189, 16.21, 16.231, 16.252, 16.273, 16.294, 16.315, 16.336, 16.357, 16.378, 16.399, 16.42, 16.441, 16.462, 16.483, 16.504, 16.525, 16.546, 16.567, 16.588, 16.609, 16.63, 16.651, 16.672, 16.693, 16.714, 16.735, 16.756, 16.777, 16.798, 16.819, 16.84, 16.861, 16.882, 16.903, 16.924, 16.945, 16.966, 16.987, 17.0, 17.021, 17.042, 17.063, 17.084, 17.105, 17.126, 17.147, 17.168, 17.189, 17.21, 17.231, 17.252, 17.273, 17.294, 17.315, 17.336, 17.357, 17.378, 17.399, 17.42, 17.441, 17.462, 17.483, 17.504, 17.525, 17.546, 17.567, 17.588, 17.609, 17.63, 17.651, 17.672, 17.693, 17.714, 17.735, 17.756, 17.777, 17.798, 17.819, 17.84, 17.861, 17.882, 17.903, 17.924, 17.945, 17.966, 17.987, 18.0, 18.021, 18.042, 18.063, 18.084, 18.105, 18.126, 18.147, 18.168, 18.189, 18.21, 18.231, 18.252, 18.273, 18.294, 18.315, 18.336, 18.357, 18.378, 18.399, 18.42, 18.441, 18.462, 18.483, 18.504, 18.525, 18.546, 18.567, 18.588, 18.609, 18.63, 18.651, 18.672, 18.693, 18.714, 18.735, 18.756, 18.777, 18.798, 18.819, 18.84, 18.861, 18.882, 18.903, 18.924, 18.945, 18.966, 18.987, 19.0, 19.021, 19.042, 19.063, 19.084, 19.105, 19.126, 19.147, 19.168, 19.189, 19.21, 19.231, 19.252, 19.273, 19.294, 19.315, 19.336, 19.357, 19.378, 19.399, 19.42, 19.441, 19.462, 19.483, 19.504, 19.525, 19.546, 19.567, 19.588, 19.609, 19.63, 19.651, 19.672, 19.693, 19.714, 19.735, 19.756, 19.777, 19.798, 19.819, 19.84, 19.861, 19.882, 19.903, 19.924, 19.945, 19.966, 19.987, 20.0, 20.021, 20.042, 20.063, 20.084, 20.105, 20.126, 20.147, 20.168, 20.189, 20.21, 20.231, 20.252, 20.273, 20.294, 20.315, 20.336, 20.357, 20.378, 20.399, 20.42, 20.441, 20.462, 20.483, 20.504, 20.525, 20.546, 20.567, 20.588, 20.609, 20.63, 20.651, 20.672, 20.693, 20.714, 20.735, 20.756, 20.777, 20.798, 20.819, 20.84, 20.861, 20.882, 20.903, 20.924, 20.945, 20.966, 20.987, 21.0, 21.021, 21.042, 21.063, 21.084, 21.105, 21.126, 21.147, 21.168, 21.189, 21.21, 21.231, 21.252, 21.273, 21.294, 21.315, 21.336, 21.357, 21.378, 21.399, 21.42, 21.441, 21.462, 21.483, 21.504, 21.525, 21.546, 21.567, 21.588, 21.609, 21.63, 21.651, 21.672, 21.693, 21.714, 21.735, 21.756, 21.777, 21.798, 21.819, 21.84, 21.861, 21.882, 21.903, 21.924, 21.945, 21.966, 21.987, 22.0, 22.021, 22.042, 22.063, 22.084, 22.105, 22.126, 22.147, 22.168, 22.189, 22.21, 22.231, 22.252, 22.273, 22.294, 22.315, 22.336, 22.357, 22.378, 22.399, 22.42, 22.441, 22.462, 22.483, 22.504, 22.525, 22.546, 22.567, 22.588, 22.609, 22.63, 22.651, 22.672, 22.693, 22.714, 22.735, 22.756, 22.777, 22.798, 22.819, 22.84, 22.861, 22.882, 22.903, 22.924, 22.945, 22.966, 22.987, 23.0, 23.021, 23.042, 23.063, 23.084, 23.105, 23.126, 23.147, 23.168, 23.189, 23.21, 23.231, 23.252, 23.273, 23.294, 23.315, 23.336, 23.357, 23.378, 23.399, 23.42, 23.441, 23.462, 23.483, 23.504, 23.525, 23.546, 23.567, 23.588, 23.609, 23.63, 23.651, 23.672, 23.693, 23.714, 23.735, 23.756, 23.777, 23.798, 23.819, 23.84, 23.861, 23.882, 23.903, 23.924, 23.945, 23.966, 23.987, 24.0, 24.021, 24.042, 24.063, 24.084, 24.105, 24.126, 24.147, 24.168, 24.189, 24.21, 24.231, 24.252, 24.273, 24.294, 24.315, 24.336, 24.357, 24.378, 24.399, 24.42, 24.441, 24.462, 24.483, 24.504, 24.525, 24.546, 24.567, 24.588, 24.609, 24.63, 24.651, 24.672, 24.693, 24.714, 24.735, 24.756, 24.777, 24.798, 24.819, 24.84, 24.861, 24.882, 24.903, 24.924, 24.945, 24.966, 24.987, 25.0, 25.021, 25.042, 25.063, 25.084, 25.105, 25.126, 25.147, 25.168, 25.189, 25.21, 25.231, 25.252, 25.273, 25.294, 25.315, 25.336, 25.357, 25.378, 25.399, 25.42, 25.441, 25.462, 25.483, 25.504, 25.525, 25.546, 25.567, 25.588, 25.609, 25.63, 25.651, 25.672, 25.693, 25.714, 25.735, 25.756, 25.777, 25.798, 25.819, 25.84, 25.861, 25.882, 25.903, 25.924, 25.945, 25.966, 25.987, 26.0, 26.021, 26.042, 26.063, 26.084, 26.105, 26.126, 26.147, 26.168, 26.189, 26.21, 26.231, 26.252, 26.273, 26.294, 26.315, 26.336, 26.357, 26.378, 26.399, 26.42, 26.441, 26.462, 26.483, 26.504, 26.525, 26.546, 26.567, 26.588, 26.609, 26.63, 26.651, 26.672, 26.693, 26.714, 26.735, 26.756, 26.777, 26.798, 26.819, 26.84, 26.861, 26.882, 26.903, 26.924, 26.945, 26.966, 26.987, 27.0, 27.021, 27.042, 27.063, 27.084, 27.105, 27.126, 27.147, 27.168, 27.189, 27.21, 27.231, 27.252, 27.273, 27.294, 27.315, 27.336, 27.357, 27.378, 27.399, 27.42, 27.441, 27.462, 27.483, 27.504, 27.525, 27.546, 27.567, 27.588, 27.609, 27.63, 27.651, 27.672, 27.693, 27.714, 27.735, 27.756, 27.777, 27.798, 27.819, 27.84, 27.861, 27.882, 27.903, 27.924, 27.945, 27.966, 27.987, 28.0, 28.021, 28.042, 28.063, 28.084, 28.105, 28.126, 28.147, 28.168, 28.189, 28.21, 28.231, 28.252, 28.273, 28.294, 28.315, 28.336, 28.357, 28.378, 28.399, 28.42, 28.441, 28.462, 28.483, 28.504, 28.525, 28.546, 28.567, 28.588, 28.609, 28.63, 28.651, 28.672, 28.693, 28.714, 28.735, 28.756, 28.777, 28.798, 28.819, 28.84, 28.861, 28.882, 28.903, 28.924, 28.945, 28.966, 28.987, 29.0, 29.021, 29.042, 29.063, 29.084, 29.105, 29.126, 29.147, 29.168, 29.189, 29.21, 29.231, 29.252, 29.273, 29.294, 29.315, 29.336, 29.357, 29.378, 29.399, 29.42, 29.441, 29.462, 29.483, 29.504, 29.525, 29.546, 29.567, 29.588, 29.609, 29.63, 29.651, 29.672, 29.693, 29.714, 29.735, 29.756, 29.777, 29.798, 29.819, 29.84, 29.861, 29.882, 29.903, 29.924, 29.945, 29.966, 29.987, 30.0, 30.021, 30.042, 30.063, 30.084, 30.105, 30.126, 30.147, 30.168, 30.189, 30.21, 30.231, 30.252, 30.273, 30.294, 30.315, 30.336, 30.357, 30.378, 30.399, 30.42, 30.441, 30.462, 30.483, 30.504, 30.525, 30.546, 30.567, 30.588, 30.609, 30.63, 30.651, 30.672, 30.693, 30.714, 30.735, 30.756, 30.777, 30.798, 30.819, 30.84, 30.861, 30.882, 30.903, 30.924, 30.945, 30.966, 30.987, 31.0, 31.021, 31.042, 31.063, 31.084, 31.105, 31.126, 31.147, 31.168, 31.189, 31.21, 31.231, 31.252, 31.273, 31.294, 31.315, 31.336, 31.357, 31.378, 31.399, 31.42, 31.441, 31.462, 31.483, 31.504, 31.525, 31.546, 31.567, 31.588, 31.609, 31.63, 31.651, 31.672, 31.693, 31.714, 31.735, 31.756, 31.777, 31.798, 31.819, 31.84, 31.861, 31.882, 31.903, 31.924, 31.945, 31.966, 31.987, 32.0, 32.021, 32.042, 32.063, 32.084, 32.105, 32.126, 32.147, 32.168, 32.189, 32.21, 32.231, 32.252, 32.273, 32.294, 32.315, 32.336, 32.357, 32.378, 32.399, 32.42, 32.441, 32.462, 32.483, 32.504, 32.525, 32.546, 32.567, 32.588, 32.609, 32.63, 32.651, 32.672, 32.693, 32.714, 32.735, 32.756, 32.777, 32.798, 32.819, 32.84, 32.861, 32.882, 32.903, 32.924, 32.945, 32.966, 32.987, 33.0, 33.021, 33.042, 33.063, 33.084, 33.105, 33.126, 33.147, 33.168, 33.189, 33.21, 33.231, 33.252, 33.273, 33.294, 33.315, 33.336, 33.357, 33.378, 33.399, 33.42, 33.441, 33.462, 33.483, 33.504, 33.525, 33.546, 33.567, 33.588, 33.609, 33.63, 33.651, 33.672, 33.693, 3
```

```

18     031(i)=0;
19   end
20   N41(i)=(1-031(i))*(N01(i))+031(i)*N01(i+1)
21   N42(i)=(1-031(i))*N01(i+1)+031(i)*N02(i)
22   if i==2 then
23     printf('%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\n',0,N02(i),f(N01(i)),
24     N01(i+1),f(N01(i+1)),net31(i),031(i),N41(
25     i),N42(i))
26   else
27     printf('%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\t%.3f\n',N01(i),N02(i),f(N01(i)),N01(i
28     +1),f(N01(i+1)),net31(i),031(i),N41(i),N42(i)
29   )
30 end
31
32 end
33
34 printf('\n\nTherefore the solution is %.3f',N42(13))

```

Scilab code Exa 22.4 Hopfield Neural Network

```

1 //Example 22.4
2 //Hopfield Neural Network
3 //Page no. 766
4 clc;clear;close;
5
6 A=[1,2,1;-1,1,1;1,0,-1];
7 disp(inv(A),'Inverse of A =',A,'A =')
8 for i=1:3
9   for j=1:3
10    k=0;
11    for l=1:3
12      k=k+A(i,l)*A(j,l)

```

```
13      end
14      T(i,j)=k;
15  end
16 end
17 disp(T, 'T =')
```

Scilab code Exa 22.5 RBF Network

```
1 //Example 22.5
2 //RBF Network
3 //Page no. 773
4 clc;clear;close;
5
6 def(f,'y=f(x)', 'y=10*sin(x)')
7 printf('Input\tDesired\tNetwork\tError\n\t\t
          Output\t\tOutput\n
          _____\n')
8 in
     =[0.7053,0.7060,0.7097,1.5056,1.5103,1.5377,2.2481,2.2514,2.2599,-
9 n
     =[6.4828,6.4883,6.5164,9.9786,9.9816,9.9944,7.7926,7.7718,7.7180,-
10 for i=1:18
11     printf(' %.4f\t%.4f\t%.4f\t%.4f\n',in(i),f
           (in(i)),n(i),f(in(i))-n(i))
12 end
```

Scilab code Exa 22.7 First Order ODE