Hands-on Session: Linear Algebra and Ordinary Differential Equations with SCILAB

Scilab and Its Applications to Global Optimization and Fractional Differential Equations

SGGS IE & T, Nanded, April 23-25, 2010

By

Vishwesh Anant Vyawahare

Systems and Control Engineering
Indian Institute Of Technology Bombay

April 24, 2010

We will be doing...

- Solving $Ax = b$
- Eigenvalues and Eigenvectors
- Matrix Decompositions
- Solving Linear ODE
- Solving Non-linear ODE
- Solving System of ODEs

$Ax = b$

- Consider

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} -1 \\ 2 \\ 4 \end{bmatrix}$$

  Enter these matrices.

- Two ways to solve in Scilab:
  1. `x = inv(A)*b`
  2. $x = A \backslash b$

- Check `b - Ax = 0`

Playing with Hilbert Matrix

- Hilbert matrix is generated using the relation:

$$A(i,j) = \frac{1}{i+j-1}$$

- Generate a 4×4 Hilbert matrix using `for` command.
- Check if A is singular.
- Scilab directly gives the Inverse of Hilbert matrix.
- Type the command `B = testmatrix('hilb',4)`.
- Now type `A1 = inv(B)`.

Eigenvalues and Eigenvectors

- Type `spec(A)`
- All eigenvalues are *positive*. Why?
- We can also get eigenvectors using the command:
  `[evects,evals] = spec(A)`
- Check the relation: $Ax = \lambda x$.

Rank and Kernel

- Type `rank(A)`
- We find the kernel of A.
- Kernel of a matrix is a set of those vectors x for which $Ax = 0$.
- Can you guess what will be the kernel of A? (hint: A is non-singular!!)
- Type `kernel(A)`.

Norm and Condition number of Hilbert Matrix

- Condition number of a matrix is the measure of its suitability in the numerical analysis.
- It is given as $\kappa(A) = \frac{\lambda_{max}}{\lambda_{min}}$
- Extract the largest and smallest values from the matrix `evals`. Take the ratio.
- Condition number is also given as: $\kappa(A) = \|A\|_2 \|A^{-1}\|_2$
- Norm is evaluated in Scilab using the command `norm(A)`.

LU Decomposition

- Any non-singular matrix having all principal minors non-zero can be decomposed into two triangular matrices.
- Our matrix A has these properties.
- Type `[L,U] = lu(A)`
- Note: L is lower-triangular and U is upper-triangular.

QR Decomposition

- $A = QR$, where Q is orthogonal matrix and R is upper-triangular.
- Type `[Q,R] = qr(A)`
- Check the orthonormality of $Q \Rightarrow$ Columns of Q have norm=1 and they are orthogonal to each other.
- Extract first and second columns of Q using the commands `Q1 = Q(:,1)` and `Q2 = Q(:,2)`.
- Type `norm(Q1)` and `norm(Q2)`.
- Find their inner product (dot product). Do element-wise multiplication and add the elements of the resulting matrix.
- Type `cm = Q1.*Q2`.
- Now type `ip = sum(cm)`.

Cholesky Decomposition

- For a positive-definite matrix, cholesky decomposition splits a matrix into two matrices $A = L^T L$.
- Our matrix A is positive-definite.
- Type `L = chol(A)`
- L is upper-triangular with strictly positive diagonal entries.

Ordinary Differential Equations: Linear

- Scilab solves ODEs using numerical methods.
- Consider a simple ODE:

$$\frac{dy}{dt} = -y, \quad y(0) = 1.$$

- We know the solution, $y(t) = e^{-t}$.
- Scilab code for defining an ode is as follows:

```
function ydot = f(t,y)
ydot = -y
endfunction
```

- Now we define the initial condition, initial time, and time-range for which the solution is required.

```
y0 = 1; t0 = 0; t = 0:0.1:5;
```

- Solve the ode using the command

```
y=ode(y0,t0,t,f);
```

Ordinary Differential Equations: Non-linear

- Scilab also solves non-linear ODEs.
- Consider an ODE modeling the dynamics of spread of a disease:

$$\frac{dy}{dt} = ky(1-y), \quad y(0) = 1/10000.$$

- Solve this ODE for $k = 0.2$ and $0 \leq t \leq 100$. Define `k=0.2`.
- Scilab code is:

```
function ydot = f(t,y)
ydot = k*y*(1-y)
endfunction
```

- Now we define the initial condition, initial time, and time-range for which the solution is required.

```
y0 = 1/10000; t0 = 0; t = 0:1:100;
```

- Solve the ode using the command

```
y=ode(y0,t0,t,f);
```

System of Non-linear ODEs: Lorenz Attractor

- The Lorenz attractor was introduced by Edward Lorenz in 1963.
- Model of the convection rolls arising in the atmosphere.
- Example of **deterministic** system showing **chaotic** behaviour.
- System of three non-linear ODEs:

$$\frac{d}{dt}y_1(t) = \sigma(y_2(t) - y_1(t))$$
$$\frac{d}{dt}y_2(t) = y_1(t)(r - y_3(t))y_2(t)$$
$$\frac{d}{dt}y_3(t) = y_1(t)y_2(t) - by_3(t)$$

- We solve the system for the parameters values $\sigma = 10$, $b = \frac{8}{3}$, and $r = 0.2$.
- Initial conditions are $y_1(0) = -6.2$, $y_2(0) = -7$, and $y_3(0) = -23$.