Matrix Applications with SCILAB

By

Gilberto E. Urroz, Ph.D., P.E.

Distributed by

infoClearinghouse.com

A "zip" file containing all of the programs in this document (and other SCILAB documents at InfoClearinghouse.com) can be downloaded at the following site:

http://www.engineering.usu.edu/cee/faculty/gurro/Software_Calculators/Scilab_Docs/ScilabBookFunctions.zip

The author's SCILAB web page can be accessed at:

http://www.engineering.usu.edu/cee/faculty/gurro/Scilab.html

Please report any errors in this document to: gurro@cc.usu.edu

Matrix applications **2**
Electric circuits 2
Structural mechanics 4
Dimensionless numbers in fluid mechanics 7
Stress at a point in a solid in equilibrium 10
Principal stresses at a point 15
Multiple linear fitting 17
Polynomial fitting 19
Selecting the best fitting 24

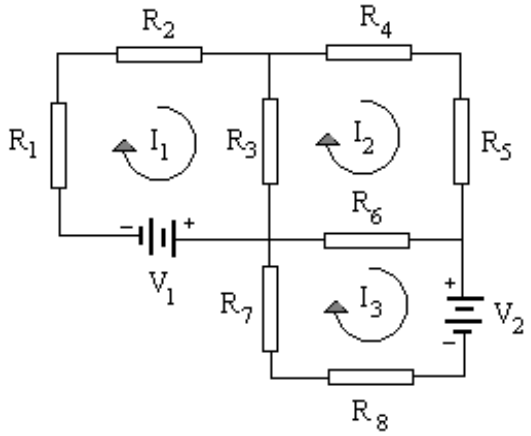
Exercises **26**

Matrix applications

In this section we explore some applications of matrices in the physical sciences.

Electric circuits

Consider the simple electrical circuit shown in the figure below.



Given the values of the electric resistance, $R_1 = R_3 = R_5 = R_7 = 1.5\ k\Omega$, $R_2 = R_4 = R_6 = R_8 = 800\ \Omega$, and the known steady voltages $V_1 = 12$ V, $V_2 = 24$ V. We are asked to determine the electrical currents I_1, I_2, and I_3, associated with the circulation loops shown in the figure.

The circulation loops shown pre-determine for us a preferred direction in each loop to write Kirchoff law of voltage in a closed loop. Basically, we start at a node in the circuit and move around a given loop subtracting voltages $R \cdot I$ if the current is in the same direction as the loop direction, or adding voltages if the current and the loop directions are opposite. When encountering a voltage source, the voltage from the source is added or subtracted according to the orientation of the voltage source with respect to the loop circulation direction. We stop back at the same node were we started to complete the voltage equation for a given loop.

For the case shown in the figure we can write:

I_1: $-R_1 \cdot I_1 - R_2 \cdot I_1 - R_3 \cdot (I_1 - I_2) - V_1 = 0$
I_2: $-R_4 \cdot I_2 - R_5 \cdot I_2 - R_6 \cdot (I_2 - I_3) - R_3 \cdot (I_2 - I_1) = 0$
I_3: $-V_2 - R_8 \cdot I_3 - R_7 \cdot I_3 - R_6 \cdot I_3 = 0$

Replacing the values of the resistances and voltage sources:

I_1: $-1500 \cdot I_1 - 800 \cdot I_1 - 1500 \cdot (I_1 - I_2) - 12 = 0$
I_2: $-800 \cdot I_2 - 1500 \cdot I_2 - 800 \cdot (I_2 - I_3) - 1500 \cdot (I_2 - I_1) = 0$
I_3: $-24 - 800 \cdot I_3 - 1500 \cdot I_3 - 800 \cdot (I_3 - I_2) = 0$

Algebraic manipulation of the equations reduce them to the following system of linear equations:

$$\begin{aligned} -3800 I_1 + 1500 I_2 &= 12 \\ 1500 I_1 - 4600 I_2 + 800 I_3 &= 0 \\ 800 I_2 - 3100 I_3 &= 24 \end{aligned}$$

This system can be written as a matricial system:

$$\mathbf{A} = \begin{bmatrix} -3800 & 1500 & 0 \\ 1500 & -4600 & 800 \\ 0 & 800 & -3100 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 12 \\ 0 \\ 24 \end{bmatrix}.$$

A solution can be obtained by using left division, an inverse matrix, or function *linsolve.* The three methods are illustrated below:

```
 -->A=[-3800,1500,0;1500,-4600,800;0,800,-3100]
  A  =

! - 3800.    1500.    0.    !
!   1500.  - 4600.    800.  !
!   0.       800.   - 3100. !

-->b = [12;0;24]
 b  =

!   12. !
!   0.  !
!   24. !
```

Using left division:

```
-->x = A\b
 x  =

! -  .0042929 !
! -  .0028753 !
! -  .0084840 !
```

Using the inverse of matrix A:

```
-->x = inv(A)*b
 x  =

! -  .0042929 !
! -  .0028753 !
! -  .0084840 !
```

Using *linsolve*:

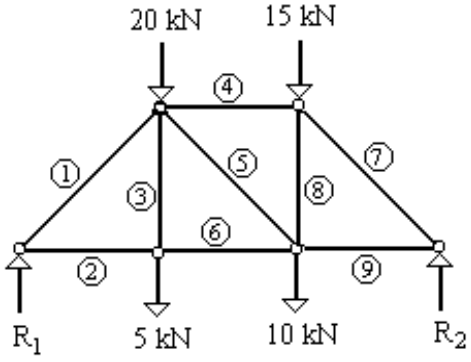
```
-->c = -b
 c  =

! - 12. !
!   0.  !
! - 24. !

-->x = linsolve(A,c)
 x  =

! -  .0042929 !
! -  .0028753 !
! -  .0084840 !
```


Regardless of the method used to obtain the solution, the final results are:

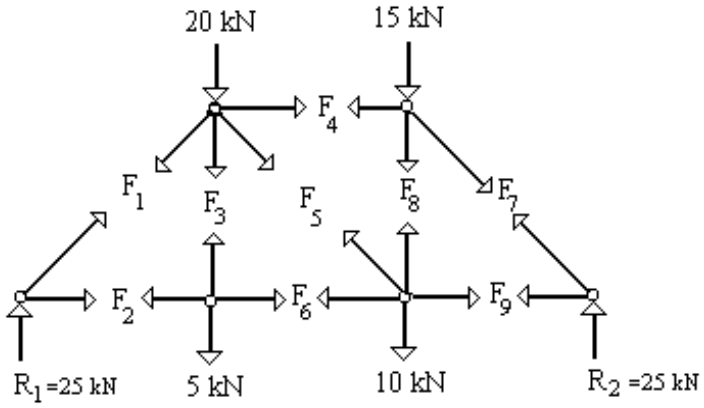
I_1 = -0.0042929 A = -4.2929 mA, I_2 = -0.0028753 A = -2.8753 mA, I_3 = -0.0084840 A = 8.484 mA.

Structural mechanics

Consider the truss structure shown in the figure below. Horizontal and vertical bars are of length 1.0 m, and diagonal bars 1.4142 m. All acute angles in the truss are 45°.



By isolating each node, as shown in the figure below, we can write the following equations for node equilibrium (i.e., ΣFx = 0, ΣFy = 0):



$$F_2 + F_1 \cos 45^o = 0,$$
$$25 + F_1 \sin 45^o = 0,$$
$$-F_2 + F_6 = 0,$$
$$-5 + F_3 = 0,$$
$$F_4 - F_1 \cos 45^o + F_5 \cos 45^o = 0,$$
$$-20 - F_3 - F_1 \cos 45^o - F_5 \cos 45^o = 0,$$
$$-F_4 + F_7 \cos 45^o = 0,$$

$$-15 - F_8 - F_7 \cos 45^o = 0,$$
$$-F_6 + F_9 - F_5 \cos 45^o = 0,$$
$$-10 + F_8 + F_5 \cos 45^o = 0,$$
$$-F_9 - F_7 \cos 45^o = 0,$$
$$25 + F_7 \cos 45^o = 0.$$

With $\sin 45^o = \cos 45^o = 0.866$, then we have:

$$\begin{aligned}
&0.866\,F_1 + F_2 = 0\\
&0.866\,F_1 = -25\\
&-F_2 + F_6 = 0\\
&F_3 = 5\\
&-0.866F_1 + F_4 + 0.866F_5 = 0\\
&-0.866F_1 - F_3 - 0.866F_5 = 20\\
&-F_4 + 0.866F_7 = 0\\
&-0.866F_7 - F_8 = 15\\
&-0.866F_5 - F_6 + F_9 = 0\\
&0.866F_5 + F_8 = 10\\
&-0.866F_7 - F_9 = 0\\
&0.866\,F_7 = -25
\end{aligned}$$

We have a total of 12 equations with 9 unknowns. The system is over-determined, so we choose, arbitrarily, the first 9 equations:

$$\begin{aligned}
&0.866\,F_1 + F_2 = 0\\
&0.866\,F_1 = -25\\
&-F_2 + F_6 = 0\\
&F_3 = 5\\
&-0.866F_1 + F_4 + 0.866F_5 = 0\\
&-0.866F_1 - F_3 - 0.866F_5 = 20\\
&-F_4 + 0.866F_7 = 0\\
&-0.866F_7 - F_8 = 15\\
&-0.866F_5 - F_6 + F9 = 0
\end{aligned}$$

Writing the system as a matrix equation:

$$\begin{bmatrix}
0.866 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0.866 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
-0.866 & 0 & 0 & 1 & 0.866 & 0 & 0 & 0 & 0\\
-0.866 & 0 & -1 & 0 & -0.866 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & -1 & 0 & 0 & 0.866 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & -0.866 & -1 & 0\\
0 & 0 & 0 & 0 & -0.866 & -1 & 0 & 0 & 1
\end{bmatrix}
\cdot
\begin{bmatrix} F_1\\ F_2\\ F_3\\ F_4\\ F_5\\ F_6\\ F_7\\ F_8\\ F_9 \end{bmatrix}
=
\begin{bmatrix} 0\\ -25\\ 0\\ 5\\ 0\\ 20\\ 0\\ 15\\ 0 \end{bmatrix}$$

The coefficient matrix for this problem is a *sparse matrix*. To solve this problem using SCILAB we need to load vectors containing the indices and the values of the non-zero elements of the matrix A, i.e.,

```
-->index =
[1,1;1,2;2,1;3,2;3,6;4,3;5,1;5,4;5,5;6,1;6,3;6,5;7,4;7,7;8,7;8,8;9,5;9,6;9,9];

-->values = [0.866,1,0.866,-1,1,1,-0.866,1,0.866,-0.866,-1,-0.866,-1,0.866,-
0.866,-1,-0.866,-1,1];

-->dim = [9,9];
```

To check that the dimensions of the matrix *index* and vector *values* are compatible use the function *size*

```
-->size(index), size(values)
 ans  =

!   19.    2. !
 ans  =

!   1.    19. !
```

Next, we create the matrix of coefficients **A** as a sparse matrix:

```
-->A = sparse(index,values,dim);
```

The full matrix can be seen by using:

```
-->full(A)
 ans  =

!    .866    1.    0.    0.    0.       0.    0.       0.    0. !
!    .866    0.    0.    0.    0.       0.    0.       0.    0. !
!   0.     - 1.    0.    0.    0.       1.    0.       0.    0. !
!   0.       0.    1.    0.    0.       0.    0.       0.    0. !
! -  .866    0.    0.    1.     .866    0.    0.       0.    0. !
! -  .866    0.  - 1.    0.  -  .866    0.    0.       0.    0. !
!   0.       0.    0.  - 1.    0.       0.     .866    0.    0. !
!   0.       0.    0.    0.    0.       0.  -  .866  - 1.    0. !
!   0.       0.    0.    0.  -  .866  - 1.    0.       0.    1. !
```

Next, we define the right-hand side vector:

```
-->b = full(sparse(indexb,valuesb,dimb));

-->b
 b  =

!   0.  !
! - 25. !
!   0.  !
!   5.  !
!   0.  !
!   20. !
!   0.  !
!   15. !
!   0.  !
```


The solution to the system is:

```
-->x = lusolve(A,b)
 x  =

! - 28.86836 !
!   25.      !
!   5.       !
! - 25.      !
!   0.       !
!   25.      !
! - 28.86836 !
!   10.      !
!   25.      !
```

i.e.,

$$F_1 = -28.87 \text{ kN}, F_2 = 25 \text{ kN}, F_3 = 5 \text{ kN},$$
$$F_4 = -25 \text{ kN}, F_5 = 0 \text{ kN}, F_6 = 25 \text{ kN},$$
$$F_7 = -28.87 \text{ kN}, F_8 = 10 \text{ kN}, F_9 = 25 \text{ kN}.$$

Dimensionless numbers in fluid mechanics

Dimensional analysis is a technique used in fluid mechanics, and other sciences, to reduce the number of variables involved in an experiment by creating *dimensionless numbers* that combine the original set of variables. In order to obtain these dimensionless numbers, we make use of the principle of *dimensional homogeneity*, which basically states that an equation derived from conservation laws and material properties should have the same dimensions on both sides of the equation. For example, the equation for the distance traveled by a projectile dropped from rest at a certain elevation above the ground is given by $d = \frac{1}{2} gt^2$, where $g = 9.806 \text{ m/s}^2$, is the acceleration of gravity, and t is the time in seconds. The distance d is given in meters. Instead of dealing with units, we refer to three (sometimes more) fundamental dimensions: length (L), time (T), and mass (M). We use brackets to refer to the dimensions of a quantity, thus, [d] = L, g = [LT^{-2}], and t = [T]. Replacing dimensions in the formula for d we have:

$$[d] = [1/2][g][t]^2 = 1 \cdot LT^{-2} \cdot T^2 = L,$$

as expected. Thus, we say that the equation $d = \frac{1}{2} gt^2$ is dimensionally homogeneous.

Suppose that we have an experiment that involves the following variables (showed with their dimensions attached):

D = a diameter (L)
V = a flow velocity (LT^{-1})
ν = kinematic viscosity of the fluid (L^2T^{-1})
ρ = density of the fluid (ML^{-3})
E = bulk density of the fluid ($ML^{-1}T^{-2}$)
σ = surface tension of the fluid (MT^{-2})
Δp = a characteristic pressure drop in the flow ($ML^{-1}T^{-2}$)
g = acceleration of gravity (LT^{-2})

There are m = 8 variables which need n = 3 dimensions to be expressed (i.e., L, T, and M). Buckingham's Π theorem indicates that you can form r = m – n = 8 – 3 = 5 dimensionless parameters. The technique consists in selecting one geometric variable, in this case we have no choice but to select D, the only variable that represents geometry alone; a kinematic variable, V (you can also choose ν), i.e., a variable involving length and time; and, finally, a dynamic variable, say ρ, i.e., a variable involving length, time, and mass. These three variables, D, V and ρ, become repeating variables, i.e., variables that will participate in each of the dimensionless parameters to be formed. Each dimensionless parameter ,or Π number, is formed by multiplying the repeating variables raised to a certain unknown power and multiplying one of the remaining variables. For example, we can form for this case the following Π parameters:

$$\Pi_1 = \rho^x \cdot D^y \cdot V^z \cdot \nu,$$

$$\Pi_2 = \rho^x \cdot D^y \cdot V^z \cdot E,$$

$$\Pi_3 = \rho^x \cdot D^y \cdot V^z \cdot \sigma,$$

$$\Pi_4 = \rho^x \cdot D^y \cdot V^z \cdot \Delta p,$$

$$\Pi_5 = \rho^x \cdot D^y \cdot V^z \cdot g.$$

Since the Π numbers are dimensionless, we can write $[\Pi_i] = L^0 \cdot T^0 \cdot M^0$, for i = 1,2, 3, 4, 5. Replacing the dimensions of the variables involved in each dimensionless parameters we can write, for example, for Π_1:

$$L^0 \cdot T^0 \cdot M^0 = (ML^{-3})^x \cdot (L)^y \cdot (LT^{-1})^z \cdot (L^2T^{-1}) = (L)^{-3x+y+z+2} \cdot (T)^{-z-1} \cdot (M)^x,$$

From which we get the following equations:

$$\begin{aligned} -3x+y+z+2 &= 0 \\ -z-1 &= 0 \\ x &= 0 \end{aligned}$$

Or,

$$\begin{bmatrix} -3 & 1 & 1 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \\ 0 \end{bmatrix}$$

If we replace the dimensions of the non-repeating variables in the remaining Π parameters, we can expand the matrix equation shown above to read:

$$\begin{bmatrix} -3 & 1 & 1 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -2 & 1 & 0 & 1 & -1 \\ 1 & 2 & 2 & 2 & 2 \\ 0 & -1 & -1 & -1 & 0 \end{bmatrix}$$

So, the independent vector **b** has become a matrix **B**, and we can write the matrix equation **A**·**X** = **B**. The columns of **B** are the negatives of the exponents of the dimensions, L, T, and M, in that order, of each of the non-repeating variables as shown in the Π parameters that we set up.

To solve for the variables x,y,z for each parameter using SCILAB, we propose using function *gausselimd*:

```
-->A = [-3,1,1;0,0,-1;1,0,0]
 A  =

! - 3.    1.    1. !
!   0.    0.  - 1. !
!   1.    0.    0. !

-->B = [-2,1,0,1,-1;1,2,2,2,2;0,-1,-1,-1,0]
 B  =

! - 2.    1.    0.    1.  - 1. !
!   1.    2.    2.    2.    2. !
!   0.  - 1.  - 1.  - 1.    0. !

-->getf('gausselimd')

-->[x,detA]=gausselimd(A,B)
 detA  =

    1.
 x  =

!   0.  - 1.  - 1.  - 1.    0. !
! - 1.    0.  - 1.    0.    1. !
! - 1.  - 2.  - 2.  - 2.  - 2. !
```

The result is the matrix

$$\mathbf{X} = \begin{pmatrix} 0 & -1 & -1 & -1 & 0 \\ -1 & 0 & -1 & 0 & 1 \\ -1 & -2 & -2 & -2 & -2 \end{pmatrix},$$

each column representing the values of x,y,z, for the repeating variables in each of the dimensionless parameters, thus we have:

$$\Pi_1 = \rho^0 \cdot D^{-1} \cdot V^{-1} \cdot \nu = \nu / DV,$$

$$\Pi_2 = \rho^{-1} \cdot D^0 \cdot V^{-2} \cdot E = E / \rho V^2,$$

$$\Pi_3 = \rho^{-1} \cdot D^{-1} \cdot V^{-2} \cdot \sigma = \sigma / \rho DV^2$$

$$\Pi_4 = \rho^{-1} \cdot D^0 \cdot V^{-2} \cdot \Delta p = \Delta p / \rho V^2,$$

$$\Pi_5 = \rho^0 \cdot D^1 \cdot V^{-2} \cdot g = gD / V^2.$$

Note: if you don't want to use function *gausselimd* you can use, for example, left-division:

```
-->A\B
 ans  =

!   0.  - 1.  - 1.  - 1.    0. !
! - 1.    0.  - 1.    0.    1. !
```



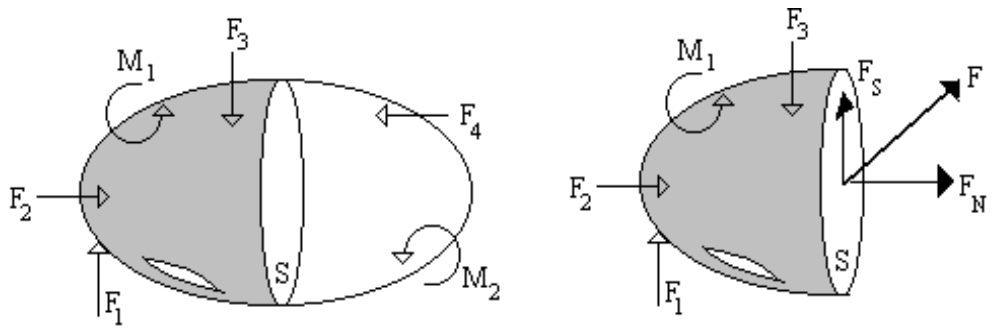
```
! - 1.  - 2.  - 2.  - 2.  - 2. !
```

Or, a Gauss-Jordan elimination with function *rref:*

```
-->A_aug = [A B]
 A_aug  =

! - 3.    1.    1.  - 2.    1.    0.    1.  - 1. !
!   0.    0.  - 1.    1.    2.    2.    2.    2. !
!   1.    0.    0.    0.  - 1.  - 1.  - 1.    0. !

-->rref(A_aug)
 ans  =

!   1.    0.    0.    0.  - 1.  - 1.  - 1.    0. !
!   0.    1.    0.  - 1.    0.  - 1.    0.    1. !
!   0.    0.    1.  - 1.  - 2.  - 2.  - 2.  - 2. !
```

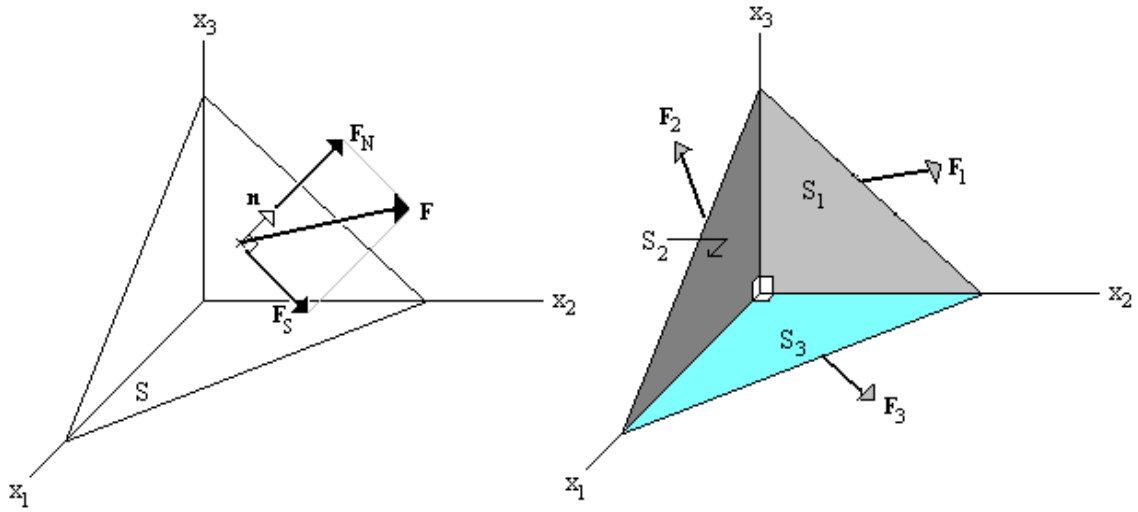
Stress at a point in a solid in equilibrium

Consider a solid body in equilibrium under the action of a system of forces and moments, as illustrated in the figure below. If we were to make an imaginary cut through the solid body, so that we can separate it into two parts at section S.



The effect of the part that we remove to the right of the cut surface S is replaced by the force **F**, which in turn can be decomposed into a normal component, $\mathbf{F}_N$, and a shear or tangential component $\mathbf{F}_S$.

Suppose now that we isolate a small particle off this solid body, and we do it by cutting the body with four planes so that we can draw the particle as shown in the left-hand side of the figure below.



Three of the planes are chose to be perpendicular to each other so that they help us identify a Cartesian coordinate system (x_1, x_2, x_3) as shown above. The surface S, limiting the particle from above, has a normal unit vector $\mathbf{n} = [\cos\alpha_1, \cos\alpha_2, \cos\alpha_3]$, where $\cos\alpha_1$, $\cos\alpha_2$, and $\cos\alpha_3$ are the direction cosines of **n**. The other three surfaces limiting the particle are S_1, S_2, and S_3, where the sub-index indicates the axis that is normal to the surface. The effect of the solid body on this particle is represented by the forces **F**, $\mathbf{F}_1$, $\mathbf{F}_2$, and $\mathbf{F}_3$, acting, respectively, upon surfaces S, S_1, S_2, and S_3. Let the areas corresponding to each surface S, S_1, S_2, and S_3 be given by A, A_1, A_2, and A_3. It is possible to show, from the geometry of the figure, that

$$A_1 = A\cdot\cos\alpha_1,\ A_2 = A\cdot\cos\alpha_2, \text{ and } A_3 = A\cdot\cos\alpha_3.$$

The force **F** on surface S can be decomposed into a normal component,

$$\mathbf{F}_N = F_N\cdot\mathbf{n} = F_N\cdot[\cos\alpha_1, \cos\alpha_2, \cos\alpha_3] = F_N\cdot(\cos\alpha_1\cdot\mathbf{e}_1 + \cos\alpha_2\cdot\mathbf{e}_2 + \cos\alpha_3\cdot\mathbf{e}_3) = F_N\cdot\cos\alpha_j\cdot\mathbf{e}_j,\ (*)$$

(*) using Einstein's repeated index convention.

and a shear component,

$$\mathbf{F}_S = \mathbf{F} - \mathbf{F}_N,$$

as shown in the figure above. The vectors are the unit vectors corresponding to the three coordinate directions.

The forces on surfaces S_1, S_2, and S_3 can be written in terms of the stress components, σ_{ij}, shown in the figure below, as

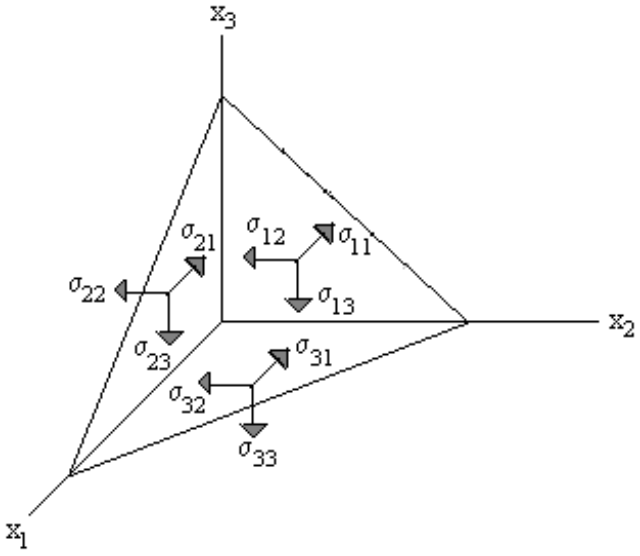
$$\mathbf{F}_i = [-\sigma_{i1}, -\sigma_{i2}, -\sigma_{i3}]\cdot A_i = (-\sigma_{i1}\cdot\mathbf{e}_1 - \sigma_{i2}\cdot\mathbf{e}_2 - \sigma_{i3}\cdot\mathbf{e}_3)\cdot A_i = (-\sigma_{i1}\cdot\mathbf{e}_1 - \sigma_{i2}\cdot\mathbf{e}_2 - \sigma_{i3}\cdot\mathbf{e}_3)\cdot A\cdot\cos\alpha_i\ (i = 1,2,3).$$

Using Einstein's repeated index convention we can write

$$\mathbf{F}_i = [-\sigma_{i1}, -\sigma_{i2}, -\sigma_{i3}] \cdot A_i = -\sigma_{ij} \cdot \mathbf{e}_j \cdot A_i = -\sigma_{ij} \cdot \mathbf{e}_j \cdot A \cdot \cos \alpha_i \ (i = 1,2,3).$$

The sub-indices identifying each stress components are chosen so that the first sub-index represents the sub-index of the axis normal to the surface of interest, and the second represents the direction along which the stress acts.

Stresses with the same sub-index, σ_{ii} (i = 1,2,3) act normal to the appropriate surface and are known as *normal stresses*. The other two components on each of the surfaces S_1, S_2, and S_3, are known as *shear stresses*, i.e., σ_{ij}, $i \neq j$. The direction of action as shown in the figure below is the conventional way to represent the stresses, namely, the stresses are positive when acting in the negative coordinate directions, so that the resulting forces have a negative sign, as shown in the equation above.



The stress components illustrated in the figure above can be written as a matrix known as the *stress tensor*,

$$\mathbf{T} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}.$$

The set up of the Cartesian coordinate system and the stresses in the particle under consideration can be used to define the stress condition at a point in the limit when the dimensions of the particle tend to zero. Under such conditions you can prove that the stress tensor is symmetric, i.e., $\sigma_{ij} = \sigma_{ji}$. Therefore, to define completely the state of stress at a point we need only to know the three normal stresses and three of the shear stresses.

For the equilibrium of force on the particle we can write

$$\mathbf{F} + \Sigma\mathbf{F}_i = \mathbf{F} + \Sigma(-\sigma_{ij}\cdot\mathbf{e}_j\cdot A\cdot\cos\alpha_i) = 0, \text{ or } \mathbf{F} = \sigma_{ij}\cdot\mathbf{e}_j\cdot A\cdot\cos\alpha_i$$

[using Einstein's convention, with both i and j repeated]

If we let

$$\mathbf{F} = \boldsymbol{\sigma}\cdot A,$$

where s is the stress vector on surface S, and replace this value in the previous equation we get

$$\boldsymbol{\sigma} = \sigma_{ij}\cdot\cos\alpha_i\cdot\mathbf{e}_j = \cos\alpha_i\cdot\mathbf{e}_j\cdot\sigma_{ij} = \mathbf{n}\cdot\mathbf{T}$$

To find the magnitude of the normal component of the stress vector, i.e., the projection of the stress $\boldsymbol{\sigma}$ along the unit normal vector $\mathbf{n}$, we use

$$\sigma_n = \boldsymbol{\sigma}\bullet\mathbf{n} \;/\; |\mathbf{n}| = \boldsymbol{\sigma}\bullet\mathbf{n} = (\sigma_{ij}\cdot\cos\alpha_i\cdot\mathbf{e}_j)\bullet(\cos\alpha_k\cdot\mathbf{e}_k) = \sigma_{ij}\cdot\cos\alpha_i\cdot\cos\alpha_k\cdot(\mathbf{e}_j\bullet\mathbf{e}_k).$$

We can prove that for the unit vectors in the Cartesian coordinate system,

$$\mathbf{e}_j\bullet\mathbf{e}_k = \delta_{jk},$$

where δ_{jk} is Dirac's delta function. Thus, the normal component of the stress on surface S is

$$\sigma_n = \sigma_{ij}\cdot\cos\alpha_i\cdot\cos\alpha_k\cdot\delta_{jk}$$

Since the product indicated in this expression is zero if $j\neq k$, then the only terms surviving are those for which j = k, i.e.,

$$\sigma_n = \sigma_{ij}\cdot\cos\alpha_i\cdot\cos\alpha_j = \cos\alpha_j\cdot\sigma_{ij}\cdot\cos\alpha_i.$$

You can prove that this latter result can be written in vector and matrix notation as the quadratic form

$$\sigma_n = \mathbf{n}\cdot\mathbf{T}\cdot\mathbf{n}^T,$$

where

$$\mathbf{n} = \cos\alpha_j\cdot\mathbf{e}_j.$$

Thus, the normal stress magnitude can be written as a quadratic form for any normal unit vector $\mathbf{n} = n_j\cdot\mathbf{e}_j$, written as a row vectors, with $n_j = \cos\alpha_j$, j = 1, 2, 3. Also, the normal stress as a vector will be written as

$$\boldsymbol{\sigma}_n = \sigma_n\cdot\mathbf{n} = (\mathbf{n}\cdot\mathbf{T}\cdot\mathbf{n}^T)\cdot\mathbf{n}$$

The normal force is given by

$$\mathbf{F}_N = \boldsymbol{\sigma}_n\cdot A = (\sigma_n\cdot A)\cdot\mathbf{n}.$$

The shear force can be written in terms of shear stress on surface S, $\mathbf{F}_S = \mathbf{F} - \mathbf{F}_N = \boldsymbol{\sigma}_S\cdot A$, so that

$$\boldsymbol{\sigma}_S = \boldsymbol{\sigma} - \boldsymbol{\sigma}_n.$$

Example:

Let the stress at a point be given by

$$\mathbf{T} = \begin{bmatrix} 25 & -10 & 20 \\ -10 & 30 & 15 \\ 20 & 15 & 40 \end{bmatrix} \cdot Pa$$

Determine the total stress $\boldsymbol{\sigma}$, the normal stress $\boldsymbol{\sigma}_n$, and shear stress $\boldsymbol{\sigma}_S$, if the surface S has a normal unit vector **n** = [0.5 0.25 0.8292]. What are the total force **F**, the normal force $\mathbf{F}_n$, and the shear force $\mathbf{F}_S$, if the surface S has an area of 0.00001 m^2

Solution:

To calculate the total stress we use

$$\vec{\sigma} = n \cdot T \cdot \vec{e} = \begin{bmatrix} 25 & -10 & 20 \\ -10 & 30 & 15 \\ 20 & 15 & 40 \end{bmatrix} \cdot \begin{bmatrix} 0.5 \\ 0.25 \\ 0.8292 \end{bmatrix}$$

This result can be obtained by using SCILAB as follows:

```
-->T = [25,-10,20;-10,30,15;20,15,40], n = [0.5,0.25,0.829]
 T  =

!   25.  - 10.    20. !
! - 10.    30.    15. !
!   20.    15.    40. !
 n  =

!    .5     .25     .829 !

-->sigma = T*n'
 sigma  =

!   26.58  !
!   14.935 !
!   46.91  !
```

To calculate the normal stress, use:

$$\vec{\sigma} = n \cdot T \cdot \vec{e} = \begin{bmatrix} 25 & -10 & 20 \\ -10 & 30 & 15 \\ 20 & 15 & 40 \end{bmatrix} \cdot \begin{bmatrix} 0.5 \\ 0.25 \\ 0.8292 \end{bmatrix}$$

Using SCILAB:

```
-->sigma_n = n*T*n'
 sigma_n  =
```



```
    55.91214
```

The result is $\boxed{\sigma_n = 55.93 \text{ Pa.}}$

The shear stress is given by

$$\boldsymbol{\sigma}_S = \boldsymbol{\sigma} - \boldsymbol{\sigma}_n,$$

or, using SCILAB:

```
-->sigma_s = sigma - sigma_n*n'
 sigma_s  =

! - 1.37607   !
!    .956965  !
!    .5588359 !
```

The forces can be calculated by multiplying the stresses times the area of the surface S, i.e., $\mathbf{F} = \boldsymbol{\sigma} \cdot A$, $\mathbf{F}_n = \boldsymbol{\sigma}_n \cdot A$, and $\mathbf{F}_t = \boldsymbol{\sigma}_t \cdot A$. Using Maple, the forces are calculated as:

```
-->A = 0.00001
 A  =

     .00001

-->Fn = sigma_n*n'*A
 Fn  =

!    .0002796 !
!    .0001398 !
!    .0004635 !

-->Ft = sigma_s*A
 Ft  =

! -  .0000138 !
!    .0000096 !
!    .0000056 !

-->F = sigma*A
 F  =

!    .0002658 !
!    .0001494 !
!    .0004691 !
```

In paper, these forces are written as

$$\mathbf{F}_n = (2.79\mathbf{i} + 1.39\mathbf{j} + 4.63\mathbf{k}) \times 10^{-4} \text{ N},$$
$$\mathbf{F}_s = (-1.38\mathbf{i} + 0.96\mathbf{j} + 0.54\mathbf{k}) \times 10^{-5} \text{ N},$$
$$\mathbf{F} = (2.66\mathbf{i} + 1.49\mathbf{j} + 4.69\mathbf{k}) \times 10^{-4} \text{ N}.$$

Principal stresses at a point

Given the stress tensor **T** representing the state of stress at a point P in a Cartesian coordinate system (x_1, x_2, x_3), suppose that you want to find the normal vector, or vectors, **n** for which the stress is only in the normal direction. In other words, we are trying to find **n** and σ_n such that

$$\mathbf{T}\cdot\mathbf{n} = \sigma_n\cdot\mathbf{n}.$$

This equation is the eigenvalue equation for the matrix T with eigenvalues σ_n and eigenvectors **n**.

Recall that this equation can be written also as

$$(\mathbf{T} - \sigma_n\cdot\mathbf{I})\cdot\mathbf{n} = 0,$$

which has non-trivial solution if

$$det(\mathbf{T} - \sigma_n\cdot\mathbf{I}) = 0.$$

For the previous example, we can write

$$\det(\mathbf{T}-\sigma_n\cdot\mathbf{I}) = \begin{vmatrix} 25-\sigma_n & -10 & 20 \\ -10 & 30-\sigma_n & 15 \\ 20 & 15 & 40-\sigma_n \end{vmatrix} = 0\cdot$$

To obtain the eigenvalues and eigenvectors of **T** we use function *eigenvectors* in SCILAB:

```
-->getf('eigenvectors')

-->[x,sigm] = eigenvectors(T)
 sigm  =

!   1.1203785    37.800525    56.079096 !
 x  =

! -  .6673642  -  .6009826     .4398237 !
! -  .5119885     .7991229     .3150721 !
!    .5408260     .0149168     .8410022 !
```

In paper we would write:

$\mathbf{n}_1$ = [-0.667 -0.512 0.541], $(\sigma_n)_1$ = 1.12,
$\mathbf{n}_2$ = [-0.601 0.800 0.015], $(\sigma_n)_2$ = 37.80,
$\mathbf{n}_3$ =[0.440 0.315 0.841], $(\sigma_n)_3$ = 56.08.

The three normal stresses found are known as the *principal stresses* at the point. The eigenvalues represent the normal vectors to the surfaces where those principal stresses act. These directions are known as the *principal axes*.

Multiple linear fitting

Consider a data set of the form

x_1	x_2	x_3	...	x_n	y
x_{11}	x_{21}	x_{31}	...	x_{n1}	y_1
x_{12}	x_{22}	x_{32}	...	x_{n2}	y_2
x_{13}	x_{32}	x_{33}	...	x_{n3}	y_3
.	.	.		.	.
.	.	.	.	.	.
$x_{1,m-1}$	$x_{2,m-1}$	$x_{3,m-1}$	...	$x_{n,m-1}$	y_{m-1}
$x_{1,m}$	$x_{2,m}$	$x_{3,m}$	...	$x_{n,m}$	y_m

Suppose that we search for a data fitting of the form

$$y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3 + \ldots + b_n \cdot x_n.$$

You can obtain the least-square approximation to the values of the coefficients

$$\mathbf{b} = [b_0 \quad b_1 \quad b_2 \quad b_3 \ldots b_n],$$

by putting together the matrix **X**

$$\begin{bmatrix} 1 & x_{11} & x_{21} & x_{31} & \ldots & x_{n1} \\ 1 & x_{12} & x_{22} & x_{32} & \ldots & x_{n2} \\ 1 & x_{13} & x_{32} & x_{33} & \ldots & x_{n3} \\ . & . & . & . & & . \\ . & . & . & . & . & . \\ 1 & x_{1,m} & x_{2,m} & x_{3,m} & \ldots & x_{n,m} \end{bmatrix}$$

Then, the vector of coefficients is obtained from

$$\mathbf{b} = (\mathbf{X}^T \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^T \cdot \mathbf{y},$$

where **y** is the vector

$$\mathbf{y} = [y_1 \, y_2 \ldots y_m]^T.$$

For *example*, use the following data to obtain the multiple linear fitting

$$y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3,$$

x_1	x_2	x_3	y
1.20	3.10	2.00	5.70
2.50	3.10	2.50	8.20

3.50	4.50	2.50	5.00
4.00	4.50	3.00	8.20
6.00	5.00	3.50	9.50

With SCILAB you can proceed as follows:

First, enter the vectors x_1, x_2, x_3, and y, as row vectors:

```
->x1 = [1.2,2.5,3.5,4.0,6.0]
 x1  =

!   1.2    2.5    3.5    4.    6. !

-->x2 = [3.1,3.1,4.5,4.5,5.0]
 x2  =

!   3.1    3.1    4.5    4.5    5. !

-->x3 = [2.0,2.5,2.5,3.0,3.5]
 x3  =

!   2.    2.5    2.5    3.    3.5 !

-->y = [5.7,8.2,5.0,8.2,9.5]
 y  =

!   5.7    8.2    5.    8.2    9.5 !
```

Next, we form matrix X and replace y by its transpose:

```
-->X = [ones(5,1) x1' x2' x3']
 X  =

!   1.    1.2    3.1    2.  !
!   1.    2.5    3.1    2.5 !
!   1.    3.5    4.5    2.5 !
!   1.    4.     4.5    3.  !
!   1.    6.     5.     3.5 !
```

The vector of coefficients for the multiple linear equation is calculated as:

```
-->b =inv(X'*X)*X'*y
 b  =

! - 2.1649851 !
! -  .7144632 !
! - 1.7850398 !
!   7.0941849 !
```

Thus, the multiple-linear regression equation is:

$$y^{\wedge} = -2.1649851 - 0.7144632 \cdot x_1 - 1.7850398 \cdot x_2 + 7.0941849 \cdot x_3.$$

This function can be used to evaluate y for values of x given as $[x_1,x_2,x_3]$. For example, for $[x_1,x_2,x_3] = [3,4,2]$, construct a vector xx = [1,3,4,2], and multiply xx times b, to obtain y(xx):

```
-->xx = [1,3,4,2]
 xx  =
```



```
!   1.    3.    4.    2. !

-->xx*b
 ans  =

    2.739836
```

The fitted values of y corresponding to the values of x_1, x_2, and x_3 from the table are obtained from **y** = **X**·**b**:

```
-->X*b
 ans  =

!   5.6324056 !
!   8.2506958 !
!   5.0371769 !
!   8.2270378 !
!   9.4526839 !
```

Compare these fitted values with the original data as shown in the table below:

x_1	x_2	x_3	y	y-fitted
1.20	3.10	2.00	5.70	5.63
2.50	3.10	2.50	8.20	8.25
3.50	4.50	2.50	5.00	5.04
4.00	4.50	3.00	8.20	8.23
6.00	5.00	3.50	9.50	9.45

Polynomial fitting

Consider the x-y data set

x	**y**
x_1	y_1
x_2	y_2
x_3	y_3
.	.
.	.
x_{n-1}	y_{n-1}
x_n	y_n

Suppose that we want to fit a polynomial or order p to this data set. In other words, we seek a fitting of the form

$$y = b_0 + b_1 \cdot x + b_2 \cdot x^2 + b_3 \cdot x^3 + \ldots + b_p \cdot x^p.$$

You can obtain the least-square approximation to the values of the coefficients

$$\mathbf{b} = [b_0 \quad b_1 \quad b_2 \quad b_3 \ldots b_p],$$

by putting together the matrix **X**

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & \ldots & x_1^{p-2} & x_1^{p-1} \\ 1 & x_2 & x_2^2 & x_2^3 & \ldots & x_2^{p-2} & x_2^{p-1} \\ 1 & x_3 & x_3^2 & x_3^3 & \ldots & x_3^{p-2} & x_3^{p-1} \\ . & . & . & . & & . & . \\ . & . & . & . & . & . & . \\ 1 & x_n & x_n^2 & x_n^3 & \ldots & x_n^{p-2} & x_n^{p-1} \end{bmatrix}$$

Then, the vector of coefficients is obtained from $\mathbf{b} = (\mathbf{X}^T \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^T \cdot \mathbf{y}$, where **y** is the vector $\mathbf{y} = [y_1 \; y_2 \ldots y_n]^T$.

Earlier on, in this chapter, we defined the Vandermonde matrix corresponding to a vector $\mathbf{x} = [x_1 \; x_2 \ldots x_n]$ as

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & \ldots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & x_2^3 & \ldots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & x_3^3 & \ldots & x_3^{n-1} \\ . & . & . & . & & . \\ . & . & . & . & . & . \\ 1 & x_n & x_n^2 & x_n^3 & \ldots & x_n^{n-1} \end{bmatrix}$$

Notice that this matrix is similar to the matrix **X** of interest to the polynomial fitting, but having only *n*, rather than (*p+1*) columns.

We can take advantage of the VANDERMONDE function to create the matrix **X** if we observe the following rules:

If $p = n-1$, $\mathbf{X} = \mathbf{V}_n$.
If $p < n-1$, then we need to remove columns *p+2, …, n-1, n* from matrix $\mathbf{V}_n$ to form matrix **X**.
If $p > n-1$, then we need to add columns *n+1, …, p-1, p+1*, to matrix $\mathbf{V}_n$ to form matrix **X**.

After **X** is ready, and having the vector **y** available, the calculation of the coefficient vector **b** is the same as in multiple linear fitting (the previous matrix application).

Because we can fit a polynomial of any degree to our data, we need to be able to evaluate the fitting by checking on a couple of parameters, namely, the sum of squared errors (SSE) and the correlation coefficient, r. These parameters are defined as follows:

Given the vectors **x** and **y** of data to be fit to the polynomial equation, we form the matrix **X** and use it to calculate a vector of polynomial coefficients **b**. We can calculate a *vector of fitted data*, **y**', by using

$$\mathbf{y}' = \mathbf{X} \cdot \mathbf{b}.$$

An *error vector* is calculated by

$$\mathbf{e} = \mathbf{y} - \mathbf{y}'.$$

The *sum of square errors* is equal to the square of the magnitude of the error vector, i.e.,

$$SSE = |\mathbf{e}|^2 = \mathbf{e} \bullet \mathbf{e} = \Sigma\, e_i^2 = \Sigma\, (y_i - y'_i)^2.$$

To calculate the correlation coefficient we need to calculate first what is known as the *sum of squared totals*, SST, defined as

$$SST = \Sigma\, (y_i - \bar{y})^2,$$

where $\bar{y}$ is the *mean value* of the original y values, i.e.,

$$\bar{y} = (\Sigma y_i)/n.$$

In terms of SSE and SST, the *correlation coefficient* is defined by

$$r = \sqrt{1 - \frac{SSE}{SST}}.$$

This value is constrained to the range $-1 < r < 1$. The closer r is to +1 or –1, the better the data fitting.

The following function, *polyfit*, takes as input the vectors x and y and the polynomial order *p* and returns the coefficients of the polynomial fitting (vector b), the sum of square errors (SSE), and the correlation coefficient (r):

```
function [SSE,r,b] = polyfit(xx,yy,p)

//Calculates the polynomial fitting
//y^ = b(1) + b(2)*x + b(3)*x^2 + ... + b(p)*x^p
//given data sets xx, yy, and the polynomial
//degree p.
//Vectors xx and yy are row vectors.

[n m] = size(xx');

getf('vandermonde');

V = vandermonde(xx);   //Get Vandermonde matrix for xx

//Get matrix X for solution
if p == n-1 then
   X = V;
elseif p<n-1 then
   X = V(1:n,1:p+1);
else
  X = V;
  for k = n+1:p+1
      X = [X xx'^k]
  end
end;

//Calculating coefficients b, SSE, and r
b=inv(X'*X)*X'*yy';
yfit = X*b;
err = yy'-yfit;
SSE = err'*err;
ybar = sum(yy)/n;
ybarv = ybar*ones(n,1);
SST = sum((yy'-ybarv)^2);
r = sqrt(1-SSE/SST);
```



```
//end function
```

As an *example*, use the following data to obtain a polynomial fitting with p = 2, 3, 4, 5, 6.

x	y
2.30	179.72
3.20	562.30
4.50	1969.11
1.65	65.87
9.32	31220.89
1.18	32.81
6.24	6731.48
3.45	737.41
9.89	39248.46
1.22	33.45

First, we enter:

```
--> x =[2.30,3.20,4.50,1.65,9.32,1.18,6.24,3.45,9.89,1.22];
-->
y=[179.72,562.30,1969.11,65.87,31220.89,32.81,6731.48,737.41,39248.46,33.45];
```

To fit the data to polynomials of order p = 2, 3, 4, 5, 6, 7, and 8 we use the following calls to function *polyfit*.

```
-->getf('polyfit')

-->[SSE,r,b] = polyfit(x,y,2)
 b  =

!   4527.7303 !
! - 3958.5178 !
!   742.23219 !
 r  =

     .9971908
 SSE  =

    10731140.

-->[SSE,r,b] = polyfit(x,y,3)
 b  =

! - 998.0541  !
!   1303.2053 !
! - 505.27432 !
!   79.229744 !
 r  =

     .9999768
 SSE  =

    88619.368

-->[SSE,r,b] = polyfit(x,y,4)
 b  =

!   20.917344 !
! - 2.6108313 !
```



```
! - 1.5247295 !
!   6.0491773 !
!   3.5068553 !
 r  =

    1.
 SSE  =

      7.4827578

-->[SSE,r,b] = polyfit(x,y,5)
 b  =

!   19.083718 !
!    .1745033 !
! - 2.9383508 !
!   6.3611564 !
!   3.475986  !
!    .0011220 !
 r  =

    1.
 SSE  =

    7.4140764

-->[SSE,r,b] = polyfit(x,y,6)
 b  =

! - 16.807588 !
!   67.398517 !
! - 48.814654 !
!   21.163051 !
!   1.0603971 !
!    .1930681 !
! -  .0058903 !
 r  =

    1.
 SSE  =

    3.8884213

-->[SSE,r,b] = polyfit(x,y,7)
 warning
 matrix is close to singular or badly scaled.
 results may be inaccurate. rcond =   1.1558E-19

 b  =

!   117.79067 !
! - 237.32895 !
!   218.31856 !
! - 96.918027 !
!   29.689084 !
! - 3.6422545 !
!    .25902   !
! -  .0073389 !
 r  =

    1.
 SSE  =
```



```
    1.2829472

-->[SSE,r,b] = polyfit(x,y,8)
 warning
 matrix is close to singular or badly scaled.
 results may be inaccurate. rcond =   1.7245E-23

 b  =

!   68.081558 !
! - 100.44092 !
!   65.29768  !
! - 6.3024667 !
! - 1.3844292 !
!   2.6919754 !
! -  .4920537 !
!    .0401628 !
! -  .0012344 !
 r  =

     .9999909
 SSE  =

    34695.662
```

Selecting the best fitting

The following table summarizes the values of r and SSE found for the different polynomial orders:

p	r	SSE
2	*0.9971908*	*10731140*
3	*0.9999768*	*88619.37*
4	*1*	*7.482758*
5	*1*	*7.414076*
6	*1*	*3.888421*
7	*1*	*1.282947*
8	*0.9999909*	*34695.66*

While the correlation coefficient is very close to 1.0 for all values of p, the values of SSE vary widely. The smallest value of SSE corresponds to p = 7. However, a warning is reported for values of p = 7 and 8, indicating that the results may be inaccurate. Thus, we eliminate from the analysis values for p = 7 and 8.

Discarding those values, the best fitting in terms of the minimum value of SSE is p = 6, however, there is very little difference in the values of SSE for values of p = 4, 5, or 6 (at least when compared to those values for p = 2, 3, or 8). Thus, any of the polynomial degrees p = 4, 5, or 6, will produce a good fitting of the original data.

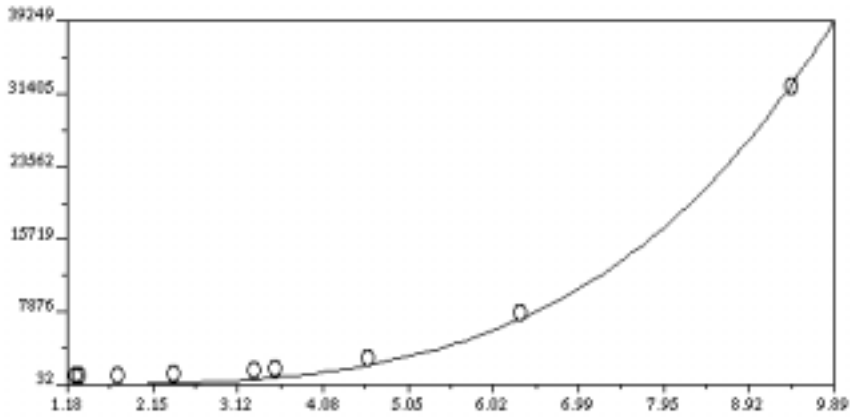
To visualize the original data and the fitted data, we can use the following function *plotpoly*, which calls on function *polyfit*. Function *plotpoly* requires the user to provide the (row) vectors x and y, as well as the polynomial degree *p*. During execution, *plotpoly* requests from the user the number of the graphics window where the plot will be produced. The function

returns the plot of the original data points and the polynomial fitting. A listing of the function follows:

```
function plotpoly(xx,yy,p)

//Plots original data and polynomial fitting
//for degree p

[m n] = size(xx);
xmin = min(xx); xmax = max(xx);
xs = [xmin:(xmax-xmin)/100:xmax];
[mm nn] = size(xs);

getf('polyfit');
[SSE,r,b] = polyfit(xx,yy,p);


XX = ones(1,nn);
for j = 1:p
    XX = [XX;xs^j];
end;

yfit = b'*XX;
ymin = min(yfit); ymax = max(yfit);

nwindow = input('Enter the graphic window number:');
xset('window',nwindow);
xset('mark',-9,3);

//plot2d(xs,yfit);
//plot2d(xx,yy,-9);

plot2d(xx',yy',-9,'010','x',[xmin ymin xmax ymax])
plot2d(xs',yfit',1,'011','x',[xmin ymin xmax ymax])


//end function
```

Calling the function for p = 5, for example, produces the following:

```
-->getf('plotpoly')

-->plotpoly(x,y,5)

 Enter the graphic window number:
-->  2
```

Exercises

[1]. Using SCILAB *rand* function to generate a matrix $\mathbf{A}_{2x2}$ (`-->A = rand(2,2)`)and a matrix $\mathbf{B}_{2x2}$ (`-->B = rand(2,2)`). Then, calculate the following:

(a)$\mathbf{A}^T$	(b)$\mathbf{A}^{-1}$	(c)$\mathbf{B}^T$	(d)$\mathbf{B}^{-1}$	(e)$\mathbf{A}+\mathbf{B}$	(f)$\mathbf{A}-\mathbf{B}$
(g) $2\mathbf{A}$	(h) $-5\mathbf{B}$	(i) $2\mathbf{A}-5\mathbf{B}$	(j) $\mathbf{A}^T - \mathbf{B}^T$	(k) $3\mathbf{A}^{-1}+5\mathbf{B}^T$	(l) $\mathbf{A}\cdot\mathbf{B}$
(m)$\mathbf{B}\cdot\mathbf{A}$	(n)$\mathbf{A}^{-1}\cdot\mathbf{A}$	(o)$\mathbf{A}^{-1}\cdot\mathbf{A}^T$	(p) $\mathbf{B}^{-1}\cdot\mathbf{A}$	(q)$\mathbf{A}^{-1}\cdot\mathbf{B}^T$	(r) $\mathbf{A}\cdot\mathbf{B}\cdot\mathbf{B}^T$
(s) $\mathbf{B}^T\cdot\mathbf{B}$	(t) $\mathbf{B}^T\cdot\mathbf{B}+\mathbf{A}^{-1}\cdot\mathbf{A}^T$	(u) $\mathbf{A}^{-1}\cdot\mathbf{A}^T\cdot\mathbf{B}^T$	(v) norm(A,2)	(w)det(A)	(x)trace(A)
(y) rank(B)	(z) cond(A)				

[2]. Using *SCILAB rand* function generate a matrix $\mathbf{A}_{3x3}$ and a matrix $\mathbf{B}_{3x3}$. Then, calculate the following:

(a)$\mathbf{A}^T$	(b)$\mathbf{A}^{-1}$	(c)$\mathbf{B}^T$	(d)$\mathbf{B}^{-1}$	(e)$\mathbf{A}+\mathbf{B}$	(f)$\mathbf{A}-\mathbf{B}$
(g) $2\mathbf{A}$	(h) $-5\mathbf{B}$	(i) $2\mathbf{A}-5\mathbf{B}$	(j) $\mathbf{A}^T - \mathbf{B}^T$	(k) $3\mathbf{A}^{-1}+5\mathbf{B}^T$	(l) $\mathbf{A}\cdot\mathbf{B}$
(m)$\mathbf{B}\cdot\mathbf{A}$	(n)$\mathbf{A}^{-1}\cdot\mathbf{A}$	(o)$\mathbf{A}^{-1}\cdot\mathbf{A}^T$	(p) $\mathbf{B}^{-1}\cdot\mathbf{A}$	(q)$\mathbf{A}^{-1}\cdot\mathbf{B}^T$	(r) $\mathbf{A}\cdot\mathbf{B}\cdot\mathbf{B}^T$
(s) $\mathbf{B}^T\cdot\mathbf{B}$	(t) $\mathbf{B}^T\cdot\mathbf{B}+\mathbf{A}^{-1}\cdot\mathbf{A}^T$	(u) $\mathbf{A}^{-1}\cdot\mathbf{A}^T\cdot\mathbf{B}^T$	(v) norm(A,2)	(w)det(A)	(x)trace(A)
(y) rank(B)	(z) cond(A)				

[3]. Using *SCILAB rand* function generate a matrix $\mathbf{A}_{3x2}$ and a matrix $\mathbf{B}_{3x2}$. Then, if possible, calculate the following:

(a)$\mathbf{A}^T$	(b)$\mathbf{A}^{-1}$	(c)$\mathbf{B}^T$	(d)$\mathbf{B}^{-1}$	(e)$\mathbf{A}+\mathbf{B}$	(f)$\mathbf{A}-\mathbf{B}$
(g) $2\mathbf{A}$	(h) $-5\mathbf{B}$	(i) $2\mathbf{A}-5\mathbf{B}$	(j) $\mathbf{A}^T - \mathbf{B}^T$	(k) $3\mathbf{A}^{-1}+5\mathbf{B}^T$	(l) $\mathbf{A}\cdot\mathbf{B}$
(m)$\mathbf{B}\cdot\mathbf{A}$	(n)$\mathbf{A}^{-1}\cdot\mathbf{A}$	(o)$\mathbf{A}^{-1}\cdot\mathbf{A}^T$	(p) $\mathbf{B}^{-1}\cdot\mathbf{A}$	(q)$\mathbf{A}^{-1}\cdot\mathbf{B}^T$	(r) $\mathbf{A}\cdot\mathbf{B}\cdot\mathbf{B}^T$
(s) $\mathbf{B}^T\cdot\mathbf{B}$	(t) $\mathbf{B}^T\cdot\mathbf{B}+\mathbf{A}^{-1}\cdot\mathbf{A}^T$	(u) $\mathbf{A}^{-1}\cdot\mathbf{A}^T\cdot\mathbf{B}^T$	(v) norm(A,2)	(w)norm(A,∞)	(x)rank(A)

[4]. Using *SCILAB rand* function generate a matrix $\mathbf{A}_{3x2}$ and a matrix $\mathbf{B}_{2x3}$. Then, if possible, calculate the following:

(a)$\mathbf{A}^T$	(b)$\mathbf{A}^{-1}$	(c)$\mathbf{B}^T$	(d)$\mathbf{B}^{-1}$	(e)$\mathbf{A}+\mathbf{B}$	(f)$\mathbf{A}-\mathbf{B}$
(g) $2\mathbf{A}$	(h) $-5\mathbf{B}$	(i) $2\mathbf{A}-5\mathbf{B}$	(j) $\mathbf{A}^T - \mathbf{B}^T$	(k) $3\mathbf{A}^{-1}+5\mathbf{B}^T$	(l) $\mathbf{A}\cdot\mathbf{B}$
(m)$\mathbf{B}\cdot\mathbf{A}$	(n)$\mathbf{A}^{-1}\cdot\mathbf{A}$	(o)$\mathbf{A}^{-1}\cdot\mathbf{A}^T$	(p) $\mathbf{B}^{-1}\cdot\mathbf{A}$	(q)$\mathbf{A}^{-1}\cdot\mathbf{B}^T$	(r) $\mathbf{A}\cdot\mathbf{B}\cdot\mathbf{B}^T$
(s) $\mathbf{B}^T\cdot\mathbf{B}$	(t) $\mathbf{B}^T\cdot\mathbf{B}+\mathbf{A}^{-1}\cdot\mathbf{A}^T$	(u) $\mathbf{A}^{-1}\cdot\mathbf{A}^T\cdot\mathbf{B}^T$	(v) norm(A,2)	(w)norm(A,∞)	(x)rank(A)

[5]. Using *SCILAB rand* function generate matrices $\mathbf{A}_{2x2}$, and $\mathbf{B}_{3x3}$. Then, obtain symmetric matrices A' and B', and anti-symmetric matrices **A**" and **B**" such that **A** = **A**'+**A**", and **B** = **B**'+**B**". Verify the results.

[6]. Generate the Vandermonde matrix, V, corresponding to the following vectors:
(a) [2,3,-1] (b) [5,5,-2,4] (c) [1,1,2,3,9] (d) [1,2,3,4,5,6]

[7]. For the matrices generated in [6] determine:
(a) determinant (b) rank (c) condition number (e)inverse

[8]. Generate the Hilbert matrix, H, of dimensions (a) 2×2, (b)3×3, (c) 4×4, and (d)5×5.

[9]. For the matrices generated in [6] determine:
(a) determinant (b) rank (c) condition number (e)inverse

[10]. Consider the system of linear equations given by:

$$\begin{bmatrix} X+2Y-Z \\ 2X+2Y-Z-3W \\ 2X+5Y+Z+W \\ X+2Z+2W \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \\ 9 \\ 1 \end{bmatrix}$$

(a) Solve the system of linear equations using Cramer's rule
(b) solve the system of linear equations using matrices and the function *linsolve*
(c) solve the system of linear equations using $x = A^{-1}b$
(d) solve the system of linear equations using Gaussian elimination and back substitution
(e) solve the system of linear equations using Gauss-Jordan elimination
(f) solve the system of linear equations using left division, i.e., x = A\b

[11]. Consider the system of linear equations:

$$2x+4y=2$$
$$x+y=2$$
$$2x+3y=2$$

(a) Sketch the lines represented by the equations in the x-y plane with -5<x<5, -5<y<5. Is there a unique solution for the system?
(b) Obtain a "solution" to the system by using SCILAB function *leastsq*. Sketch the solution point together with the lines.
(c) Determine the error involved in this "solution".

[12]. Consider the system of linear equations:

$$5x - 2y + 3z = 10$$
$$x - 3y + 4z = 20$$

(a) Obtain a solution to the system using the function *linsolve*
(b) Obtain a solution to the system using the function *leastsq*
(c) What is the rank of the matrix of coefficients for this linear system

[13]. Consider the following systems of linear equations:

X +2Y+3Z = 28,	2X +4Y+6Z = 18,	2X +4Y+6Z = -4,
3X -2Y+ Z = 4,	3X -2Y+ Z = -10,	3X -2Y+ Z = 4,
4X +2Y -Z = 10,	4X +2Y -Z = 38,	4X +2Y -Z = 24.

(a) Solve the multiple linear system by using matrices and the function *linsolve*
(b) Solve the multiple linear system by using an augmented matrix and Gauss-Jordan elimination
(c) Solve the multiple linear system by using the inverse matrix of coefficients
(d) Obtain the inverse of the matrix of coefficients by using the appropriate augmented matrix and Gauss-Jordan elimination. Verify this solution by using the function *inv*

[14]. Given the matrix

$$A := \begin{bmatrix} -1 & 0 & 5 & 4 \\ 0 & 3 & -2 & 2 \\ 5 & -2 & 4 & 1 \\ 4 & 2 & 1 & 3 \end{bmatrix}$$

corresponding to the eigenvalue problem: **Ax** = λ**x**,

(a) Obtain the characteristic matrix for the eigenvalue problem
(b) Obtain the characteristic polynomial corresponding to matrix **A**
(c) Plot the characteristic polynomial in the range -6 < λ < 10
(d) Solve the characteristic polynomial to obtain the eigenvalues, λ, of matrix A
(e) Obtain the eigenvalue, λ, of matrix **A** using the function *spec*
(f) Obtain the eigenvectors, **x**, of matrix **A** using the user-defined function *eigenvectors*

[15]. Given the matrices

$$A := \begin{bmatrix} -1 & 3 & 5 \\ 3 & 4 & 2 \\ 5 & 2 & 3 \end{bmatrix} \qquad B := \begin{bmatrix} 2 & -3 & 2 \\ -3 & 4 & 2 \\ 2 & 2 & 10 \end{bmatrix}$$

corresponding to the generalized eigenvalue problem **Ax** = λ**Bx**,

(a) Obtain the characteristic matrix for the eigenvalue problem
(b) Obtain the characteristic polynomial corresponding to matrix **A**
(g) Plot the characteristic polynomial in the range -6 < λ < 10
(h) Solve the characteristic polynomial to obtain the eigenvalues, λ, of matrix A
(i) Obtain the generalized eigenvalues, λ, of matrix **A** using the function *geigenvectors*
(j) Obtain the eigenvectors, **x**, of matrix **A** using the function *geigenvectors*

[16]. Determine the matrices L, U, and P corresponding to the LU decomposition of

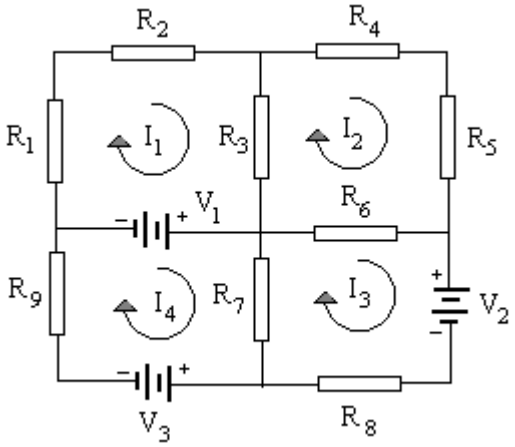
(a) matrix A in problem [14]
(b) matrix A in problem [15]
(c) matrix B in problem [15].

[17]. Determine the matrices Q and R corresponding to the QR decomposition of
(a) matrix A in problem [14]
(b) matrix A in problem [15]
(c) matrix B in problem [15].

[18]. Determine the matrices **U** and **V** of left and right vectors and the vector of singular values **s** corresponding to
(a) matrix A in problem [14]
(b) matrix A in problem [15]
(c) matrix B in problem [15].

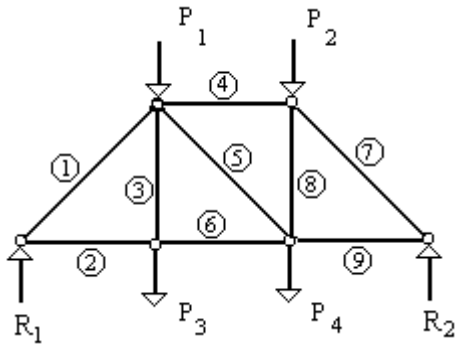
[19]. Expand the quadratic form $f(\mathbf{x})= \mathbf{x}^T\mathbf{A}\mathbf{x}$ for $\mathbf{x} = [X,Y,Z]^T$, where matrix **A** represents
(a) matrix A in problem [14]
(b) matrix A in problem [15]
(c) matrix B in problem [15].

[20]. For the electric circuit shown below



Determine the electrical currents I_1, I_2, I_3, and I_4, associated with the circulation loops shown in the figure, if
(a) $R_1 = R_3 = R_5 = R_7 = R_9 = 1.5\ k\Omega$, $R_2 = R_4 = R_6 = R_8 = 800\ \Omega$, $V_1 = 12$ V, $V_2 = 24$ V, $V_3 = 6$ V
(b) $R_1 = R_3 = R_5 = R_7 = R_9 = R_2 = R_4 = R_6 = R_8 = 1.2\ k\Omega$, $V_1 = 12$ V, $V_2 = V_3 = 6$ V
(c) $R_1 = R_3 = R_5 = R_7 = R_9 = 2.2\ k\Omega$, $R_2 = R_4 = R_6 = R_8 = 1.2\ k\Omega$, $V_1 = V_2 = V_3 = 18$ V
(d) $R_1 = R_3 = R_5 = R_7 = R_9 = 0.5\ k\Omega$, $R_2 = R_4 = R_6 = R_8 = 0.8\ k\Omega$, $V_1 = 6$ V, $V_2 = 12$ V, $V_3 = 6$ V

[21]. The truss shown in the figure below is such that all horizontal and vertical bars are of length 1.0 m, diagonal bars of length 1.4142 m, and all acute angles in the truss are 45^o.



Determine the axial forces in the truss elements if

(a) P_1 = 100 kN, P_2 = 200 kN, P_3 = 200 kN, P_4 = 100 kN.
(b) P_1 = P_2 = P_3 = P_4 = 200 kN.
(c) P_1 = 50 kN, P_2 = 150 kN, P_3 = 50 kN, P_4 = 150 kN.
(d) P_1 = 50 kN, P_2 = 150 kN, P_3 = 200 kN, P_4 = 250 kN.

To determine the reactions use the equations of moments taken about the points of application of R_1 and R_2, respectively:

$$- 3R_2 + 1\cdot(P_1 + P_3) + 2\cdot (P_2 + P_4) = 0$$
$$3R_1 - 1\cdot (P_2 + P_4) - 2\cdot (P_1 + P_3) = 0$$

[22]. Obtain dimensionless numbers to describe a fluid mechanics experiment that involves the following variables:

H = a characteristic water depth(L)
Q = a flow rate (LT^{-3})
μ = dynamic viscosity of a fluid ($ML^{-1}T^{-1}$)
γ = specific weight of a fluid ($ML^{-2}T^{-2}$)
P_0 = a characteristic pressure in the flow ($ML^{-1}T^{-2}$)
g = acceleration of gravity (LT^{-2})

Let , H, Q and γ, be the repeating variables in the dimensionless numbers.

[23]. The state of stress at a point within a solid in equilibrium is given by the stress tensor

$$T := \begin{bmatrix} 12 & -22 & 40 \\ -22 & -10 & 5 \\ 40 & 5 & 15 \end{bmatrix}$$

where the components of T represent stresses in kPa. For a plane passing through the point of interest with a normal vector given by **n** = [5, -5, 2], determine:

(a) the total stress on the plane
(b) the normal stress on the plane
(c) the shear stress on the plane

(d) the total, normal, and shear forces on the plane if the area of the plane is 0.0005 m^2.

[24]. For the stress tensor given in problem [23] determine:

(a) the principal stresses
(b) the vectors corresponding to the principal axes

[25]. The table below shows sediment load data (y, in kg/min) obtained in a laboratory flume under controlled conditions. The sediment load, y, is known to be a function of the water discharge, x_1 (lt/sec), of the mean sediment diameter, x_2 (cm), and of the flume slope, x_3 (10^{-3} m/m). Using matrices determine a multiple linear fitting of the form

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3$$

for the data provided in the table.

x_1	x_2	x_3	y
1.20	0.50	3.5	27.35
1.40	0.75	4.5	29.86
1.60	1.00	5.5	35.15
1.80	1.50	3.5	33.45
2.00	2.00	4.5	38.98
2.20	2.50	5.5	43.35
2.40	0.50	3.5	30.72
2.60	0.75	4.5	34.13
2.80	1.00	5.5	38.45
3.00	1.50	3.5	37.12
3.20	2.00	4.5	42.83
3.40	2.50	5.5	47.12

[26]. The table below shows the water discharge, y(10^3 cubic feet per second), measured at a gage station in a large river as a function of time, x(days), during a 40-day period in the early spring season. A plot of y vs. x is known as a hydrograph.

x	y
1.5	101.42
5.0	176.73
8.5	311.22
12.0	389.61
15.5	546.24
19.0	638.14
22.5	716.99
26.0	743.60
29.5	737.96
33.0	623.16
36.5	492.53

40.0	15.84

(a) Determine a polynomial fitting for this hydrograph with p = 2, 3, 4, 5, and 6, where the polynomial fitting is of the form

$$y = b_0 + b_1x + b_2x^2 + \ldots + b_px^p$$

(b) Select the best polynomial fitting for the hydrograph based on the values of the correlation coefficient and of the sum of squared errors, SSE.
(c) Plot the original hydrograph data and the fitted polynomial in the same set of axes.
(d) The area under the curve for 0 < x < 40 days, represents the total volume of water passing through the mouth of the river in that period. Using the fitted polynomial and the function *int* (*int*egral) estimate the volume of water passing through the gage station in the 40-day period of interest.

REFERENCES (for all SCILAB documents at InfoClearinghouse.com)

Abramowitz, M. and I.A. Stegun (editors), 1965,"*Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*," Dover Publications, Inc., New York.

Arora, J.S., 1985, "*Introduction to Optimum Design*," Class notes, The University of Iowa, Iowa City, Iowa.

Asian Institute of Technology, 1969, "*Hydraulic Laboratory Manual*," AIT - Bangkok, Thailand.

Berge, P., Y. Pomeau, and C. Vidal, 1984,"*Order within chaos - Towards a deterministic approach to turbulence*," John Wiley & Sons, New York.

Bras, R.L. and I. Rodriguez-Iturbe, 1985,"*Random Functions and Hydrology*," Addison-Wesley Publishing Company, Reading, Massachussetts.

Brogan, W.L., 1974,"*Modern Control Theory*," QPI series, Quantum Publisher Incorporated, New York.

Browne, M., 1999, "*Schaum's Outline of Theory and Problems of Physics for Engineering and Science*," Schaum's outlines, McGraw-Hill, New York.

Farlow, Stanley J., 1982, "*Partial Differential Equations for Scientists and Engineers*," Dover Publications Inc., New York.

Friedman, B., 1956 (reissued 1990), "*Principles and Techniques of Applied Mathematics*," Dover Publications Inc., New York.

Gomez, C. (editor), 1999, "*Engineering and Scientific Computing with Scilab*," Birkhäuser, Boston.

Gullberg, J., 1997, "*Mathematics - From the Birth of Numbers*," W. W. Norton & Company, New York.

Harman, T.L., J. Dabney, and N. Richert, 2000, "*Advanced Engineering Mathematics with MATLAB® - Second edition*," Brooks/Cole - Thompson Learning, Australia.

Harris, J.W., and H. Stocker, 1998, "*Handbook of Mathematics and Computational Science*," Springer, New York.

Hsu, H.P., 1984, "*Applied Fourier Analysis*," Harcourt Brace Jovanovich College Outline Series, Harcourt Brace Jovanovich, Publishers, San Diego.

Journel, A.G., 1989, "*Fundamentals of Geostatistics in Five Lessons*," Short Course Presented at the 28th International Geological Congress, Washington, D.C., American Geophysical Union, Washington, D.C.

Julien, P.Y., 1998,"*Erosion and Sedimentation*," Cambridge University Press, Cambridge CB2 2RU, U.K.

Keener, J.P., 1988, "*Principles of Applied Mathematics - Transformation and Approximation*," Addison-Wesley Publishing Company, Redwood City, California.

Kitanidis, P.K., 1997,"*Introduction to Geostatistics - Applications in Hydogeology*," Cambridge University Press, Cambridge CB2 2RU, U.K.

Koch, G.S., Jr., and R. F. Link, 1971, "*Statistical Analysis of Geological Data - Volumes I and II*," Dover Publications, Inc., New York.

Korn, G.A. and T.M. Korn, 1968, "*Mathematical Handbook for Scientists and Engineers*," Dover Publications, Inc., New York.

Kottegoda, N. T., and R. Rosso, 1997, "*Probability, Statistics, and Reliability for Civil and Environmental Engineers*," The Mc-Graw Hill Companies, Inc., New York.

Kreysig, E., 1983, "*Advanced Engineering Mathematics - Fifth Edition*," John Wiley & Sons, New York.

Lindfield, G. and J. Penny, 2000, "*Numerical Methods Using Matlab®,"* Prentice Hall, Upper Saddle River, New Jersey.

Magrab, E.B., S. Azarm, B. Balachandran, J. Duncan, K. Herold, and G. Walsh, 2000, "*An Engineer's Guide to MATLAB®*", Prentice Hall, Upper Saddle River, N.J., U.S.A.

McCuen, R.H., 1989,"*Hydrologic Analysis and Design - second edition*," Prentice Hall, Upper Saddle River, New Jersey.

Middleton, G.V., 2000, "*Data Analysis in the Earth Sciences Using Matlab®,"* Prentice Hall, Upper Saddle River, New Jersey.

Montgomery, D.C., G.C. Runger, and N.F. Hubele, 1998, "*Engineering Statistics*," John Wiley & Sons, Inc.

Newland, D.E., 1993, "*An Introduction to Random Vibrations, Spectral & Wavelet Analysis - Third Edition*," Longman Scientific and Technical, New York.

Nicols, G., 1995, "*Introduction to Nonlinear Science*," Cambridge University Press, Cambridge CB2 2RU, U.K.

Parker, T.S. and L.O. Chua, , "*Practical Numerical Algorithms for Chaotic Systems*," 1989, Springer-Verlag, New York.

Peitgen, H-O. and D. Saupe (editors), 1988, "*The Science of Fractal Images*," Springer-Verlag, New York.

Peitgen, H-O., H. Jürgens, and D. Saupe, 1992, "*Chaos and Fractals - New Frontiers of Science*," Springer-Verlag, New York.

Press, W.H., B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, 1989, "*Numerical Recipes - The Art of Scientific Computing (FORTRAN version)*," Cambridge University Press, Cambridge CB2 2RU, U.K.

Raghunath, H.M., 1985, "*Hydrology - Principles, Analysis and Design*," Wiley Eastern Limited, New Delhi, India.

Recktenwald, G., 2000, "*Numerical Methods with Matlab - Implementation and Application*," Prentice Hall, Upper Saddle River, N.J., U.S.A.

Rothenberg, R.I., 1991, "*Probability and Statistics*," Harcourt Brace Jovanovich College Outline Series, Harcourt Brace Jovanovich, Publishers, San Diego, CA.

Sagan, H., 1961,"*Boundary and Eigenvalue Problems in Mathematical Physics*," Dover Publications, Inc., New York.

Spanos, A., 1999,"*Probability Theory and Statistical Inference - Econometric Modeling with Observational Data*," Cambridge University Press, Cambridge CB2 2RU, U.K.

Spiegel, M. R., 1971 (second printing, 1999), "*Schaum's Outline of Theory and Problems of Advanced Mathematics for Engineers and Scientists*," Schaum's Outline Series, McGraw-Hill, New York.

Tanis, E.A., 1987, "*Statistics II - Estimation and Tests of Hypotheses*," Harcourt Brace Jovanovich College Outline Series, Harcourt Brace Jovanovich, Publishers, Fort Worth, TX.

Tinker, M. and R. Lambourne, 2000, "*Further Mathematics for the Physical Sciences*," John Wiley & Sons, LTD., Chichester, U.K.

Tolstov, G.P., 1962, "*Fourier Series*," (Translated from the Russian by R. A. Silverman), Dover Publications, New York.

Tveito, A. and R. Winther, 1998, "*Introduction to Partial Differential Equations - A Computational Approach*," Texts in Applied Mathematics 29, Springer, New York.

Urroz, G., 2000, "*Science and Engineering Mathematics with the HP 49 G - Volumes I & II*", www.greatunpublished.com, Charleston, S.C.

Urroz, G., 2001, "*Applied Engineering Mathematics with Maple*", www.greatunpublished.com, Charleston, S.C.

Winnick, J., , "*Chemical Engineering Thermodynamics - An Introduction to Thermodynamics for Undergraduate Engineering Students*," John Wiley & Sons, Inc., New York.

