Computer Implementation 1.9 *(Matlab)* *Computing reactions (p. 42)*

The manipulations for imposing essential boundary conditions and final solution for nodal unknowns can easily be achieved using *Matlab* by defining the following NodalSoln function.

MatlabFiles\Chap1\NodalSoln.m

```
function [d, rf] = NodalSoln(K, R, debc, ebcVals)
% [nd, rf] = NodalSoln(K, R, debc, ebcVals)
% Computes nodal solution and reactions
% K = global coefficient matrix
% R = global right hand side vector
% debc = list of degrees of freedom with specified values
% ebcVals = specified values
dof = length(R);
df = setdiff(1:dof, debc);
Kf = K(df, df);
Rf = R(df) - K(df, debc)*ebcVals;
dfVals = Kf\Rf;
d = zeros(dof,1);
d(debc) = ebcVals;
d(df) = dfVals;
rf = K(debc,:)*d - R(debc);
```

Two lists are established to contain degrees of freedom where essential boundary conditions are to be imposed and their corresponding values. The list of degrees of freedom to be retained are obtained by taking complement of the list of all degrees of freedom with the debc. Using the list of remaining degrees of freedom the final system of equations can now be established easily. The solution for nodal unknowns can now be obtained by using the `\` operation in *Matlab*. The complete vector of nodal values, in the original order established by the node numbering, is obtained by combining these values with those specified as essential boundary conditions. This function is used in almost all Matlab implementations presented in later chapters.

MatlabFiles\Chap1\EssentialBCEx.m

```
% Incorporate essential boundary conditions
K=[1000,10,20,30,40,50; 10,2000,21,31,41,51;
    20,21,3000,32,42,52; 30,31,32,4000,43,53;
    40,41,42,43,5000,54; 50,51,52,53,54,6000];
R=[100; 110; 120; 130; 140; 150];
[d, reactions] = NodalSoln(K, R, [1, 4, 5], [5; -7; 0])
```

```
>> EssentialBCEx

d =

    5.0000
    0.1366
    0.0796
   -7.0000
         0
    0.0433


reactions =

  1.0e+004 *
```

```
 0.4695
-2.7971
-0.0230
```