**Solve linear equations for a tridiagonal matrix. Extra material for *Introduction to Chemical Engineering Computing,* 2nd ed., Bruce A. Finlayson, Wiley (2012).**

Tridiagonal matrix – MATLAB programs
lu_tri.m LU decomposition of a tridiagonal matrix
fas_tri.m solves for the right-hand side after an LU decomposition of a tridiagonal matrix

```
function lu_tri(n)
global afd bfd cfd

%    this subroutine does an lu decomposition of
%         a tridiagonal system of
%    equations of the type that often occur with the
%    finite difference method.
%         After calling lu_tri one must call fas_tri to
%         process the right-hand side.
%    input
%    a(n), b(n), c(n) in the following equation:
%    a(i)*y(i-1) + b(i)*y(i) + c(i)*y(i+1) = d(i)
%    n - the size of the system being solved.

%   output
%    afd(n), bfd(n), cfd(n) this is the lu decomposition.
%    the original afd, bfd, cfd are destroyed.

%  lower decomposition  tridiagonal matrix

for l=2:n
  s = afd(l)/bfd(l-1);
  bfd(l) = bfd(l)-s*cfd(l-1);
  afd(l) = s;
end


function fas_tri(n)
global afd bfd cfd dfd

%    this subroutine does the fore and aft
%         sweep to solve a tridiagonal system of
%    equations of the type that often occur with the
%    finite difference method.
%         One must call lu_tri one must call first.
%    input
%    afd(n), bfd(n), cfd(n) from the lu decomposition
%         solving the problem:
%    a(i)*y(i-1) + b(i)*y(i) + c(i)*y(i+1) = d(i)
```

```
%     n - the size of the system being solved.
%   output
%     dfd(n) is the solution

%  forward sweep  tridiagonal matrix
for l=2:n
   dfd(l) = dfd(l)-afd(l)*dfd(l-1);
end
%  back substitution
dfd(n) = dfd(n)/bfd(n);
for l=2:n
   k = n-l+1;
   dfd(k) = (dfd(k)-cfd(k)*dfd(k+1))/bfd(k);
end
```