```
function [y,f]=method(x,h,y,f)
% [y,f]=method(x,h,y,f) computes one step of a general linear
 multistep method.
% y = matrix whose (j+1)st column is y_{n-p+j), j=0,...,p
% f = matrix whose (j+1)st column is f(x_{n-p+j},y_{n-p+j}), j=0,...,p
% On output, the (J+1)st columns of y and f are y_{n+1-p+j) and
% f(x{n+1-p+j},y_{n+1-p+j}), respectively.
% Here x_{n-p+j} = x + (j-p)*h.
% To use a different method, change the column vectors alpha and beta.

%Specify the parameters alpha and beta in column vectors.
%Note that since MATLAB does not allow for 0 indices, you must set
%alpha(j+1) = alpha_j, beta(j+1) = beta_j, j=0,...,p+1,

alpha=[-3/4; -1/2; 1/4; 1];      % Example: y_{n+1}-y_{n} =
beta=(1/8)*[5; 0; 19; 0]; % (h/3)*[3*f(x_{n},y_{n}) -
 2*f(x_{n-1},y_{n-1})]


p = max(size(alpha)) - 2;
a1 = -alpha(1:p+1)/alpha(p+2);
b1 = h*beta(1:p+1)/alpha(p+2);

tmp = y*a1+ f*b1; %Computes sum_{j=0}^p  [-alpha_j y_{n-p+j) +
                  %             +h*beta_j*f(x_{n-p+j},y_{n-p+j})]/
alpha(p+2)

if (beta(p+2) == 0) %method is explicit.
  y1 = tmp;

else  % method implicit. Use fixed point iteration to solve the
 equation
     % y1 = tmp + h*beta(p+2)*f(x+h,y1)/alpha(p+2), with tmp as
 above.

  tol = 1.e-5; itmax = 100; %specify tolerance and maximum # of
 iterations
  bp2 = h*beta(p+2)/alpha(p+2);xh = x+h;%auxiliary variables
  y0 = y(:,p+1); %starting vector for iteration
  t1 = 2*tol; t2=0;iter = 0;%initialize parameters for stopping
 criterion.

  while ((t1 > tol*t2) & (iter < itmax)) %iteration loop
    y1 = tmp + bp2*fun(xh,y0);
    t1 = norm(y1-y0); t2 = norm(y1) + norm(y0); %evaluate stopping
 criterion
    iter = iter+1;
    y0 = y1;
  end

  if (iter == itmax) %print warning if iteration did not converge.
    disp('  ');
```

```
    disp('Slow or no convergence in fixed point iteration.')
    disp('  x    rel. err.    tolerance     iterations ')
    disp([x    t1/t2     tol   iter ])
  end
end

y(:,1:p) = y(:,2:p+1);y(:,p+1)=y1; %update y
f(:,1:p) = f(:,2:p+1);f(:,p+1)=fun(x+h,y1); %update f
```

*Published with MATLAB® R2016b*