```
function F=fluxevaluate(fh,kh,U,uL,uR,method)


% function F=fluxevaluate(fh,kh,U,uL,uR,method)
%
% fh: function handle to the flux function
% kh: k/h factor
% U:  solution at time t = t_n
% uL: left boundary condition u(-L)=uL
% uR: right boundary condotion u(+L)=uR
% method: one of 'first-order-upwind', 'law-wendroff',
%                'richtmyer', 'maccormack'

N=length(U);
F=zeros(N+1,1);

%
%
%    F(1)  F(2)   F(3)       F(i-1)  F(i)   F(i+1)            F(N)   F(N+1)
%    |------|------|--- //  ---|-------|------|------|--- // ---|-----|
%       1      2                  i-1      i     i+1               N
%   -L                                                            L
%

switch lower(method)
case 'first-order-upwind'

  % F(i) = F^L_i, the left numerical flux
  for i=2:N
     if ( U(i)>0 ) F(i)=feval(fh,U(i-1)); end;
     if ( U(i)<0 ) F(i)=feval(fh,U(i)); end;
  end

  % BCs

  % left
  if ( U(1)>0 ) F(1)=feval(fh,uL); end;
  if ( U(1)<0 ) F(1)=feval(fh,U(1)); end;

  % right
  if ( U(N)>0 ) F(N+1)=feval(fh,U(N)); end;
  if ( U(N)<0 ) F(N+1)=feval(fh,uR); end;


case 'lax-wendroff'
     U = [uL; U; uR];             % set ghost cells
     N_ = N+2;

     F(1) = 1;
    %F(N_) = 1;

    for i = 2:N_-1
        a = (U(i) + U(i+1)) / 2;
        b = (U(i-1) + U(i))/2;
        F(i) = 0.5*feval(fh,U(i)) + 0.5*feval(fh,U(i-1)) + 0.5*kh*a*feval(fh,U(i)) - 0.5*kh*b*feval(fh,U(i-1));
    end

    if ( U(1)>0 ) F(1)=feval(fh,uL); end;
    if ( U(1)<0 ) F(1)=feval(fh,U(1)); end;

    % right
    if ( U(N)>0 ) F(N+1)=feval(fh,U(N)); end;
    if ( U(N)<0 ) F(N+1)=feval(fh,uR); end;


case 'richtmyer'
```

```
    U = [uL; U; uR];
% BCs

% left
    if ( U(1)>0 ) F(1)=feval(fh,uL); end;
    if ( U(1)<0 ) F(1)=feval(fh,U(1)); end;

% right
    if ( U(N)>0 ) F(N+1)=feval(fh,U(N)); end;
    if ( U(N)<0 ) F(N+1)=feval(fh,uR); end;

    for i = 2 : N+1
        a = (U(i+1) + U(i))/2; % U (i+0.5)
        b = (U(i-1) + U(i))/2; % U (i-0.5)


        if U(i) > 0
            F(i) = a - kh * (feval(fh,U(i+1))-feval(fh,U(i))) / 2;
            F(i) = feval(fh,F(i));
        end
        if U(i) < 0
            F(i) = b - kh * (feval(fh,U(i))-feval(fh,U(i-1))) / 2;
            F(i) = feval(fh,F(i));
        end
    end


case 'maccormack'
    U = [uL; U; uR];
% BCs

% left
    if ( U(1)>0 ) F(1)=feval(fh,uL); end;
    if ( U(1)<0 ) F(1)=feval(fh,U(1)); end;

% right
    if ( U(N)>0 ) F(N+1)=feval(fh,U(N)); end;
    if ( U(N)<0 ) F(N+1)=feval(fh,uR); end;

    for i = 2 : N + 1
        if U(i) > 0 % =
            temp = U(i) - kh*(feval(fh,U(i+1))-feval(fh,U(i)));
            F(i) = (feval(fh,U(i+1)) + feval(fh,temp))/2;
        end
        if U(i) < 0 %
            temp = U(i-1) - kh*(feval(fh,U(i)) - feval(fh,U(i-1)));
            F(i) = (feval(fh,U(i)) + feval(fh,temp))/2;
        end
    end

otherwise
  error('method is unknown');

end
```

```
Not enough input arguments.
Error in fluxevaluate (line 14)
N=length(U);
```

Published with MATLAB® R2017a