```
% Program 8.7 Implicit Newton solver for Burgers equation (Fluid Flow)
% *** Nonlinear PDE:  Diffusion Eqn ***
%     ===> Burgers Eqn (Fluid Flow Model) with Dirichlet Boundary
 Conditions
%            ***===> Computer Problem 8.4.1(b) <===***
% input: space interval [xl,xr], time interval [tb,te],
%        number of space steps M, number of time steps N
% output: solution w
% Example usage:
%       1.    clear all; clearvars; close all; clc;
%       2.    w=burgers_01b(0,1,0,1,50,50)
function w=burgers_01(xl,xr,tb,te,M,N)
% alpha, beta, Diffusion Coefficient
%alf=5;bet=4;D=.05;
alf=5;bet=4;D=1;
%f=@(x) 2*D*bet*pi*sin(pi*x)./(alf+bet*cos(pi*x));
f=@(x) sin(2*pi*x);
l=@(t) 0*t;
r=@(t) 0*t;
% Space Step
%h=(xr-xl)/M;
h=0.02;

% Time Step Size
%k=(te-tb)/N;
k=0.02;

% Square-Matrix Size
m=M+1;
% Time Step
n=N;
sigma=D*k/(h*h);
% Initial Conditions set by Function call "f")
w(:,1)=f(xl+(0:M)*h)';
%{
    w =

             0
        0.0220
        0.0441
        0.0666
        0.0897
        0.1135
        0.1383
        0.1643
        0.1916
        0.2206
        0.2513
        0.2837
        0.3175
        0.3517
        0.3838
```

```
        0.4092
        0.4187
        0.3973
        0.3247
        0.1874
        0.0000
    %}
w1=w;
% Time-Step Loop
for j=1:n
  % Space-Step Loop
  for it=1:3 % Multivariate Newton iteration
    % matrix DF1:  collects derivatives of degree 1 terms of F
    DF1=zeros(m,m);     % 21x21 matrix
    % matrix DF2:  collects derivatives of degree 2 terms of F
    DF2=zeros(m,m);     % 21x21 matrix
    %{
        D = diag(v)  :  returns a square diagonal with elements of
 vector v
                           on main diagonal
        D = diag(v,k):  places elements of vector v on kth diagonal
                            (1) k=0:  main diagonal
                            (2) k>0:  above main diagonal
                            (3) k<0:  below main diagonal
    %}
    %                                                      k>0
    DF1=diag(1+2*sigma*ones(m,1))+diag(-sigma*ones(m-1,1),1);
    % DF1 = [21x21 matrix]:  static values
    % DF1 =
    %       3 -1  0  0  0
    %       0  3 -1  0  0
    %       0  0  3 -1  0
    %       0  0  0  3 -1
    %                             k<0
    DF1=DF1+diag(-sigma*ones(m-1,1),-1);
    % DF1 = [21x21 matrix]:  static values
    % DF1 =
    %       3 -1  0  0  0
    %      -1  3 -1  0  0
    %       0 -1  3 -1  0
    %       0  0 -1  3 -1
    DF2=diag([0;k*w1(3:m)/(2*h);0])-diag([0;k*w1(1:(m-2))/(2*h);0]);
    % DF2 = [21x21 matrix]:  non-static values
    % DF2 =
    %       0      0      0      0
    %       0 0.0199      0      0
    %       0      0 0.0201      0
    %       0      0      0 0.0203
    DF2=DF2+diag([0;k*w1(2:m-1)/(2*h)],1)-diag([k*w1(2:m-1)/
(2*h);0],-1);
    % DF2 = [21x21 matrix]:  non-static values
    % DF2 =
    %       0       0       0       0       0
    %      -0.0099  0.0199  0.0099  0       0
```

```
%        0       -0.0199  0.0201  0.0199  0
%        0        0      -0.0300  0.0203  0.0300
%        0        0        0     -0.0403  0.0207
%        0        0        0       0     -0.0507
DF=DF1+DF2;
% DF = [21x21 matrix]:  non-static values
% DF =
%        3.0000 -1.0000  0       0       0
%       -1.0099  3.0199 -0.9901  0       0
%        0      -1.0199  3.0201 -0.9801  0
%        0       0      -1.0300  3.0203 -0.9700
%        0       0       0      -1.0403  3.0207
%        0       0       0       0      -1.0507
F=-w(:,j)+(DF1+DF2/2)*w1; % Using Lemma 8.11 (P441)
DF(1,:)=[1 zeros(1,m-1)]; % Dirichlet conditions for DF
DF(m,:)=[zeros(1,m-1) 1];
% DF =
%        1.0000  0       0       0       0
%       -1.0099  3.0199 -0.9901  0       0
%        0      -1.0199  3.0201 -0.9801  0
%        0       0      -1.0300  3.0203 -0.9700
%        0       0       0      -1.0403  3.0207
% Dirichlet conditions for F
F(1)=w1(1)-l(j);
%{
  F =
         0
    0.0003
    0.0006
    0.0010
    0.0013
    0.0018
    0.0023
    0.0030
    0.0038
    0.0049
    0.0062
    0.0080
    0.0104
    0.0136
    0.0178
    0.0230
    0.0285
    0.0324
    0.0308
    0.0195
   -0.1874
%}
F(m)=w1(m)-r(j);
%{
  F =
    0
    0.0003
    0.0006
```

```
        0.0010
        0.0013
        0.0018
        0.0023
        0.0030
        0.0038
        0.0049
        0.0062
        0.0080
        0.0104
        0.0136
        0.0178
        0.0230
        0.0285
        0.0324
        0.0308
        0.0195
        0.0000
    %}
    % Multivariate Newton
    w1=w1-DF\F;
    %{
        w1 =

       -0.0000
        0.0217
        0.0435
        0.0657
        0.0883
        0.1117
        0.1360
        0.1613
        0.1879
        0.2159
        0.2453
        0.2760
        0.3076
        0.3388
        0.3673
        0.3882
        0.3932
        0.3686
        0.2976
        0.1700
        0
    %}
  end
  % Each pass adds another column
  w(:,j+1)=w1;
  %{
    w =
        0         -0.0000
        0.0220     0.0217
        0.0441     0.0435
```

```
        0.0666    0.0657
        0.0897    0.0883
        0.1135    0.1117
        0.1383    0.1360
        0.1643    0.1613
        0.1916    0.1879
        0.2206    0.2159
        0.2513    0.2453
        0.2837    0.2760
        0.3175    0.3075
        0.3517    0.3388
        0.3838    0.3672
        0.4092    0.3882
        0.4187    0.3931
        0.3973    0.3685
        0.3247    0.2976
        0.1874    0.1701
        0.0000         0
  %}
end
x=xl+(0:M)*h;

t=tb+(0:n)*k;

mesh(x,t,w');
%w_transpose = w';
%{
    w_transpose =
        0         0.0220    0.0441    0.0666    0.0897    0.1135
 0.1383
       -0.0000    0.0217    0.0435    0.0657    0.0883    0.1117
 0.1360
        0.0000    0.0214    0.0429    0.0647    0.0870    0.1100
 0.1337
       -0.0000    0.0211    0.0423    0.0638    0.0857    0.1082
 0.1315
       -0.0000    0.0208    0.0417    0.0628    0.0844    0.1065
 0.1292
        0.0000    0.0205    0.0411    0.0619    0.0831    0.1048
 0.1270
%}

title({'Nonlinear PDE:  Diffusion Eqn';
       'Burgers Eqn (Fluid Flow Model) with Dirichlet Boundary
 Conditions';
       '***===> Exercise 8.4.1(b):  f(x)=sin2\pix; D=1; h=k=0.02
 <===***'});
xlabel('x');
ylabel('t');
zlabel('Fluid Flow');

Not enough input arguments.

Error in burgers_01b (line 28)
```

```
m=M+1;
```

Published with MATLAB® R2018b