

```

1 function fe2dx_n_fast_test ( )
2 %*****80
3 %
4 %% FE2DX_N_FAST_TEST tests the FE2DX_N_FAST code.
5 %
6 % Discussion:
7 %
8 %     This function sets all parameter values and initial condition information
9 %     necessary to execute the "fast" version of the fe2dx_n algorithm.
10 %
11 % Licensing:
12 %
13 %     Copyright (C) 2014 Marcus R. Garvie.
14 %     See 'mycopyright.txt' for details.
15 %
16 % Modified:
17 %
18 %     28 April 2014
19 %
20 % Author:
21 %
22 %     Marcus R. Garvie.
23 %
24 % Reference:
25 %
26 %     Marcus R Garvie, John Burkardt, Jeff Morgan,
27 %     Simple Finite Element Methods for Approximating Predator-Prey Dynamics
28 %     in Two Dimensions using MATLAB,
29 %     Submitted to Bulletin of Mathematical Biology, 2014.
30 %
31 timestamp ( );
32 fprintf ( 1, '\n' );
33 fprintf ( 1, 'FE2DX_N_FAST_TEST:\n' );
34 fprintf ( 1, '  Test the FE2DX_N_FAST function\n' );
35 fprintf ( 1, '  which applies Neumann boundary conditions as it\n' );
36 fprintf ( 1, '  approximates a solution to a predator-prey system.\n' );
37 %
38 % Set the parameters.
39 %
40 alpha = 0.4;
41 beta = 2.0;
42 gamma = 0.6;
43 delta = 1.0;
44 %
45 % Use T=150.0 for normal run.
46 % Use T=0.50 for a "quick" run that might take 15 minutes of computing.
47 %
48 T = 0.50;
49 delt = 1.0 / 384.0;
50 t = tic;
51 fe2dx_n_fast ( alpha, beta, gamma, delta, T, delt, @u0f, @v0f, @guf, @gvf );
52 t = toc ( t );
53 fprintf ( 1, '  Execution took %10.2g minutes \n', t / 60.0 );
54 %
55 % Terminate.

```

```

56 %
57 fprintf ( 1, '\n' );
58 fprintf ( 1, 'FE2DX_N_FAST_TEST:\n' );
59 fprintf ( 1, ' Normal end of execution.\n' );
60 fprintf ( 1, '\n' );
61 timestamp ( );
62 return
63 end
64 function value = u0f ( x, y )
65 %*****80
66 %
67 %% U0F evaluates the initial condition for U.
68 %
69 % Licensing:
70 %
71 % Copyright (C) 2014 Marcus R. Garvie.
72 % See 'mycopyright.txt' for details.
73 %
74 % Modified:
75 %
76 % 26 April 2014
77 %
78 % Author:
79 %
80 % Marcus R. Garvie.
81 %
82 % Parameters:
83 %
84 % Input, real X, Y, a location in the region.
85 %
86 % Output, real VALUE, the initial condition for U at (X,Y).
87 %
88 value = 6.0 / 35.0 - 2.0E-07 * ( x - 0.1 * y - 225.0 ) * ( x - 0.1 * y - 675.0 );
89 return
90 return
91 end
92 function value = v0f ( x, y )
93 %*****80
94 %
95 %% V0F evaluates the initial condition for V.
96 %
97 % Licensing:
98 %
99 % Copyright (C) 2014 Marcus R. Garvie.
100 % See 'mycopyright.txt' for details.
101 %
102 % Modified:
103 %
104 % 26 April 2014
105 %
106 % Author:
107 %
108 % Marcus R. Garvie.
109 %
110 % Parameters:
111 %
112 % Input, real X, Y, a location in the region.

```

```

113 %
114 %     Output, real VALUE, the initial condition for V at (X,Y).
115 %
116     value = 116.0 / 245.0 - 3.0E-05 * ( x - 450.0 ) - 1.2E-04 * ( y - 150.0 );
117     return
118 end
119 function value = guf ( x, y, t )
120 %*****80
121 %
122 %% GUF evaluates the Neumann boundary condition for U.
123 %
124 %     Licensing:
125 %
126 %     Copyright (C) 2014 Marcus R. Garvie.
127 %     See 'mycopyright.txt' for details.
128 %
129 %     Modified:
130 %
131 %     28 April 2014
132 %
133 %     Author:
134 %
135 %     Marcus R. Garvie.
136 %
137 %     Parameters:
138 %
139 %     Input, real X, Y, a location on the boundary.
140 %
141 %     Input, real T, the time.
142 %
143 %     Output, real VALUE, the prescribed value of dU/dn at (X,Y,T).
144 %
145     value = 0.0;
146     return
147 end
148 function value = gvf ( x, y, t )
149 %*****80
150 %
151 %% GVF evaluates the Neumann boundary condition for V.
152 %
153 %     Licensing:
154 %
155 %     Copyright (C) 2014 Marcus R. Garvie.
156 %     See 'mycopyright.txt' for details.
157 %
158 %     Modified:
159 %
160 %     28 April 2014
161 %
162 %     Author:
163 %
164 %     Marcus R. Garvie.
165 %
166 %     Parameters:
167 %
168 %     Input, real X, Y, a location on the boundary.
169 %

```

```
170 %    Input, real T, the time.
171 %
172 %    Output, real VALUE, the prescribed value of dV/dn at (X,Y,T).
173 %
174     value = 0.0;
175     return
176 end
```
