

```

1 function fe2d_nd_fast_test ( )
2 %*****80
3 %
4 %% FE2D_ND_FAST_TEST tests the FE2D_ND_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 fe2d_nd 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, 'FE2D_ND_FAST_TEST:\n' );
34 fprintf ( 1, '  Test the FE2D_ND_FAST function, which\n' );
35 fprintf ( 1, '  applies Neumann and Dirichlet 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 = 150.0;
49 T = 0.50;
50 delt = 1.0 / 384.0;
51 t = tic;
52 fe2d_nd_fast ( alpha, beta, gamma, delta, T, delt, @u0f, @v0f, @gluf, ...
53   @glvf, @g2uf, @g2vf );
54 t = toc ( t );
55 fprintf ( 1, '  Execution took %10.2g minutes \n', t / 60.0 );

```

```

56 %
57 %   Terminate.
58 %
59 fprintf ( 1, '\n' );
60 fprintf ( 1, 'FE2D_ND_FAST_TEST:\n' );
61 fprintf ( 1, '   Normal end of execution.\n' );
62 fprintf ( 1, '\n' );
63 timestamp ( );
64 return
65 end
66 function value = u0f ( x, y )
67 %*****80
68 %
69 %% U0F evaluates the initial condition for U.
70 %
71 %   Licensing:
72 %
73 %       Copyright (C) 2014 Marcus R. Garvie.
74 %       See 'mycopyright.txt' for details.
75 %
76 %   Modified:
77 %
78 %       26 April 2014
79 %
80 %   Author:
81 %
82 %       Marcus R. Garvie.
83 %
84 %   Parameters:
85 %
86 %       Input, real X, Y, a location in the region.
87 %
88 %       Output, real VALUE, the initial condition for U at (X,Y).
89 %
90 value = 6.0 / 35.0 - 2.0E-07 * ( x - 0.1 * y - 225.0 ) * ( x - 0.1 * y - 675.0 );
91 return
92 end
93 function value = v0f ( x, y )
94 %*****80
95 %
96 %% V0F evaluates the initial condition for V.
97 %
98 %   Licensing:
99 %
100 %       Copyright (C) 2014 Marcus R. Garvie.
101 %       See 'mycopyright.txt' for details.
102 %
103 %   Modified:
104 %
105 %       26 April 2014
106 %
107 %   Author:
108 %
109 %       Marcus R. Garvie.
110 %
111 %   Parameters:
112 %

```

```

113 %     Input, real X, Y, a location in the region.
114 %
115 %     Output, real VALUE, the initial condition for V at (X,Y).
116 %
117     value = 116.0 / 245.0 - 3.0E-05 * ( x - 450.0 ) - 1.2E-04 * ( y - 150.0 );
118     return
119 end
120 function value = gluf ( x, y, t )
121 %*****80
122 %
123 %% G1UF evaluates the Dirichlet boundary condition for U.
124 %
125 %     Licensing:
126 %
127 %     Copyright (C) 2014 Marcus R. Garvie.
128 %     See 'mycopyright.txt' for details.
129 %
130 %     Modified:
131 %
132 %     28 April 2014
133 %
134 %     Author:
135 %
136 %     Marcus R. Garvie.
137 %
138 %     Parameters:
139 %
140 %     Input, real X, Y, a location on the boundary.
141 %
142 %     Input, real T, the time.
143 %
144 %     Output, real VALUE, the prescribed value for U at (X,Y,T).
145 %
146     value = 0.0;
147     return
148 end
149 function value = glvf ( x, y, t )
150 %*****80
151 %
152 %% G1VF evaluates the Dirichlet boundary condition for V.
153 %
154 %     Licensing:
155 %
156 %     Copyright (C) 2014 Marcus R. Garvie.
157 %     See 'mycopyright.txt' for details.
158 %
159 %     Modified:
160 %
161 %     28 April 2014
162 %
163 %     Author:
164 %
165 %     Marcus R. Garvie.
166 %
167 %     Parameters:
168 %
169 %     Input, real X, Y, a location on the boundary.

```

```

170 %
171 %     Input, real T, the time.
172 %
173 %     Output, real VALUE, the prescribed value for V at (X,Y,T).
174 %
175     value = 0.0;
176     return
177 end
178 function value = g2uf ( x, y, t )
179 %*****80
180 %
181 %% G2UF evaluates the Neumann boundary condition for U.
182 %
183 %   Licensing:
184 %
185 %       Copyright (C) 2014 Marcus R. Garvie.
186 %       See 'mycopyright.txt' for details.
187 %
188 %   Modified:
189 %
190 %       28 April 2014
191 %
192 %   Author:
193 %
194 %       Marcus R. Garvie.
195 %
196 %   Parameters:
197 %
198 %       Input, real X, Y, a location on the boundary.
199 %
200 %       Input, real T, the time.
201 %
202 %       Output, real VALUE, the prescribed value for dU/dn at (X,Y,T).
203 %
204     value = 0.0;
205     return
206 end
207 function value = g2vf ( x, y, t )
208 %*****80
209 %
210 %% G2VF evaluates the Neumann boundary condition for V.
211 %
212 %   Licensing:
213 %
214 %       Copyright (C) 2014 Marcus R. Garvie.
215 %       See 'mycopyright.txt' for details.
216 %
217 %   Modified:
218 %
219 %       28 April 2014
220 %
221 %   Author:
222 %
223 %       Marcus R. Garvie.
224 %
225 %   Parameters:
226 %

```

```
227 %    Input, real X, Y, a location on the boundary.
228 %
229 %    Input, real T, the time.
230 %
231 %    Output, real VALUE, the prescribed value for dV/dn at (X,Y,T).
232 %
233     value = 0.0;
234     return
235 end
```
