

```

1 function fe2d_r_fast_test ( )
2 %*****80
3 %
4 %% FE2D_R_FAST_TEST tests the FE2D_R_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_r 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_R_FAST_TEST:\n' );
34 fprintf ( 1, '  Test the FE2D_R_FAST function\n' );
35 fprintf ( 1, '  which applies Robin 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 k1 = 0.01;
52 k2 = 0.01;
53 t = tic;
54 fe2d_r_fast ( alpha, beta, gamma, delta, T, delt, @u0f, @v0f, k1, k2 );
55 t = toc ( t );

```

```

56 fprintf ( 1, ' Execution took %10.2g minutes \n', t / 60.0 );
57 %
58 % Terminate.
59 %
60 fprintf ( 1, '\n' );
61 fprintf ( 1, 'FE2D_R_FAST_TEST:\n' );
62 fprintf ( 1, ' Normal end of execution.\n' );
63 fprintf ( 1, '\n' );
64 timestamp ( );
65 return
66 end
67 function value = u0f ( x, y )
68 %*****80
69 %
70 %% U0F evaluates the initial condition for U.
71 %
72 % Licensing:
73 %
74 % Copyright (C) 2014 Marcus R. Garvie.
75 % See 'mycopyright.txt' for details.
76 %
77 % Modified:
78 %
79 % 26 April 2014
80 %
81 % Author:
82 %
83 % Marcus R. Garvie.
84 %
85 % Parameters:
86 %
87 % Input, real X, Y, a location in the region.
88 %
89 % Output, real VALUE, the initial condition for U at (X,Y).
90 %
91 value = 6.0 / 35.0 - 2.0E-07 * ( x - 0.1 * y - 225.0 ) * ( x - 0.1 * y - 675.0 );
92 return
93 end
94 function value = v0f ( x, y )
95 %*****80
96 %
97 %% V0F evaluates the initial condition for V.
98 %
99 % Licensing:
100 %
101 % Copyright (C) 2014 Marcus R. Garvie.
102 % See 'mycopyright.txt' for details.
103 %
104 % Modified:
105 %
106 % 26 April 2014
107 %
108 % Author:
109 %
110 % Marcus R. Garvie.
111 %
112 % Parameters:

```

```
113 %  
114 %    Input, real X, Y, a location in the region.  
115 %  
116 %    Output, real VALUE, the initial condition for V at (X,Y).  
117 %  
118 value = 116.0 / 245.0 - 3.0E-05 * ( x - 450.0 ) - 1.2E-04 * ( y - 150.0 );  
119 return  
120 end
```
