

```

1 function fe2dx_p_fast_test ( )
2 %*****80
3 %
4 %% FE2DX_P_FAST_TEST tests the FE2DX_P_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_p algorithm.
10 %
11 % Licensing:
12 %
13 %     Copyright (C) 2014 Marcus R. Garvie.
14 %     See 'mycopyright.txt' for details.
15 %
16 % Modified:
17 %
18 %     25 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_P_FAST_TEST:\n' );
34 fprintf ( 1, '  Test the FE2DX_P_FAST function\n' );
35 fprintf ( 1, '  which applies periodic 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 a = 0.0;
45 b = 400.0;
46 %
47 % Use h = 2.0 for standard run.
48 % Use h = 40 for tiny run.
49 %
50 h = 2.0;
51 % h = 40.0;
52 %
53 % Use T=150.0 for normal run.
54 % Use T=0.50 for a "quick" run that might take 15 minutes of computing.
55 %

```

```

56 % T = 150.0;
57 T = 0.50;
58 delt = 1.0 / 384.0;
59 t = tic;
60 fe2dx_p_fast ( alpha, beta, gamma, delta, a, b, h, T, delt, @u0f, @v0f );
61 t = toc ( t );
62 fprintf ( 1, ' Execution took %10.2g minutes \n', t / 60.0 );
63 %
64 % Terminate.
65 %
66 fprintf ( 1, '\n' );
67 fprintf ( 1, 'FE2DX_P_FAST_TEST:\n' );
68 fprintf ( 1, ' Normal end of execution.\n' );
69 fprintf ( 1, '\n' );
70 timestamp ( );
71 return
72 end
73 function value = u0f ( x, y )
74 %*****80
75 %
76 %% U0F evaluates the initial condition for U.
77 %
78 % Licensing:
79 %
80 % Copyright (C) 2014 Marcus R. Garvie.
81 % See 'mycopyright.txt' for details.
82 %
83 % Modified:
84 %
85 % 26 April 2014
86 %
87 % Author:
88 %
89 % Marcus R. Garvie.
90 %
91 % Parameters:
92 %
93 % Input, real X, Y, a location in the region.
94 %
95 % Output, real VALUE, the initial condition for U at (X,Y).
96 %
97 value = 6.0 / 35.0 - 2.0E-07 * ( x - 0.1 * y - 225.0 ) * ( x - 0.1 * y - 675.0 );
98 return
99 end
100 function value = v0f ( x, y )
101 %*****80
102 %
103 %% V0F evaluates the initial condition for V.
104 %
105 % Licensing:
106 %
107 % Copyright (C) 2014 Marcus R. Garvie.
108 % See 'mycopyright.txt' for details.
109 %
110 % Modified:
111 %
112 % 26 April 2014

```

```
113 %
114 % Author:
115 %
116 %     Marcus R. Garvie.
117 %
118 % Parameters:
119 %
120 %     Input, real X, Y, a location in the region.
121 %
122 %     Output, real VALUE, the initial condition for V at (X,Y).
123 %
124     value = 116.0 / 245.0 - 3.0E-05 * ( x - 450.0 ) - 1.2E-04 * ( y - 150.0 );
125     return
126 end
```

---