```
1  // *** 時變頻率數值計算(Time-Variant-Frequency Numerical Computations) ***
2
3  // 三階微分方程式: N(t)*y'''(t)+M(t)*y''(t)+C(t)*y'(t)+K(t)*y(t)= f(t)
4  // 由齊次解，狀態空間(State-Spce)時變(Time-Variant)矩陣微分方程式，求得系統矩陣A，
5  // 進而求得A = Q * D * Qi。其中 D，Q，Qi 分別為特徵矩陣、模態矩陣、逆模態矩陣。
6  // 進而求得訊號響應值 [y''|y'|y]g = Hexp(D, Q, t) * d + [y''|y'|y]p ，
7  // 稱 Hexp(D, Q, t)為狀態空間響應函數，d是由初始值或是
8  // 邊界值而定的係數向量，兩者的預設值均為複數矩陣和複數的向量。
9  // 本求解法可對應於一般的Laplace、Fourier、Z Transform或是捲積積分法等等，
10 // 上述都是間接的求解方法。但本法則是直接求取系統響應值(矩陣微分方程式求解法)。
11
12 using Matrix_0;
13
14 int m = 4;  // 空間維度有m個自由度。
15 int r = 3;  // 狀態維度有r個自由度，即r階(Order)微分方程，A 是 12 X 12 矩陣。
16
17 // 建構初始(空)矩陣 N, M、C、K、Zero、Id。
18 ReMatrix N = (new Zero(m)).GetMatrix;
19 ReMatrix M = (new Zero(m)).GetMatrix;
20 ReMatrix C = (new Zero(m)).GetMatrix;
21 ReMatrix K = (new Zero(m)).GetMatrix;
22 ReMatrix Zero = (new Zero(m)).GetMatrix;
23 ReMatrix Id = (new Iden(m)).GetMatrix;
24
25 // 實數系統矩陣A，特徵矩陣D、模態矩陣Q。
26 ReMatrix A;
27 CxMatrix D;
28 CxMatrix Q;
29
30 // 狀態響應。速度，變位，加速度。(Step = 0.001秒，共計 t = 0.05秒)
31 double step = 0.001;
32 int iRow = (int)(0.05 / step + 1);
33
34 // 建構時間軸上的儲存矩陣，增加時間t壹行，故儲存矩陣有(m * r) + 1 行。
35 int iColD = m * r + 1;
36 CxMatrix CxVal = new CxMatrix(iRow, iColD);
37 ReMatrix ReVal = new ReMatrix(iRow, iColD);
38
```

```
39  for (int i = 0; i != iRow; i++)
40  {
41      double t = step * i;
42
43      // 建構 N、M、C、K 變數矩陣。
44      N.Matrix[0, 0] = -2.7 * t * t * Math.Sin(1.3 * t);
45      N.Matrix[0, 1] = -5.5;
46      N.Matrix[0, 2] = 0;
47      N.Matrix[0, 3] = 5.5;
48      N.Matrix[1, 0] = 3.5;
49      N.Matrix[1, 1] = -8.5;
50      N.Matrix[1, 2] = -9.8 * t * t;
51      N.Matrix[1, 3] = -4.8;
52      N.Matrix[2, 0] = 6.7;
53      N.Matrix[2, 1] = 27.9;
54      N.Matrix[2, 2] = 8.5;
55      N.Matrix[2, 3] = -20.5 * t * t * Math.Cos(1.9 * t);
56      N.Matrix[3, 0] = -1.5 * t * Math.Cos(1.9 * t);
57      N.Matrix[3, 1] = 4.8;
58      N.Matrix[3, 2] = 0;
59      N.Matrix[3, 3] = 1.5 * t * t * t;
60      //  End of N Matrix
61
62      M.Matrix[0, 0] = 19;
63      M.Matrix[0, 1] = -1.5;
64      M.Matrix[0, 2] = -2 + 13.3 * Math.Sin(0.85 * t);
65      M.Matrix[0, 3] = 1.1;
66      M.Matrix[1, 0] = -1;
67      M.Matrix[1, 1] = 15;
68      M.Matrix[1, 2] = 0;
69      M.Matrix[1, 3] = 1.3;
70      M.Matrix[2, 0] = -10 - 2.7 * Math.Cos(1.3 * t);
71      M.Matrix[2, 1] = -3;
72      M.Matrix[2, 2] = 27;
73      M.Matrix[2, 3] = 4.5;
74      M.Matrix[3, 0] = 5.5;
75      M.Matrix[3, 1] = 2.7;
76      M.Matrix[3, 2] = -2.3 * t;
77      M.Matrix[3, 3] = -3.5 * t * t;
78      // End of M Matrix
79
80      C.Matrix[0, 0] = 35;
81      C.Matrix[0, 1] = -1 - 13.2 * Math.Sin(0.35 * t);
82      C.Matrix[0, 2] = -0.5;
83      C.Matrix[0, 3] = 2.5;
84      C.Matrix[1, 0] = -1.5;
85      C.Matrix[1, 1] = 40;
```

```
 86      C.Matrix[1, 2] = -1.5;
 87      C.Matrix[1, 3] = 0;
 88      C.Matrix[2, 0] = -1.2 + 22.5 * Math.Cos(1.95 * t);
 89      C.Matrix[2, 1] = -1.5;
 90      C.Matrix[2, 2] = 75;
 91      C.Matrix[2, 3] = 0;
 92      C.Matrix[3, 0] = -27.5;
 93      C.Matrix[3, 1] = 18.3;
 94      C.Matrix[3, 2] = 9.5;
 95      C.Matrix[3, 3] = -50.9 * t * Math.Sin(2.5 * t);
 96      // End of C Matrix
 97
 98      K.Matrix[0, 0] = 60;
 99      K.Matrix[0, 1] = -8;
100      K.Matrix[0, 2] = -2 - 332 * Math.Sin(1.37 * t);
101      K.Matrix[0, 3] = -2.7;
102      K.Matrix[1, 0] = -16;
103      K.Matrix[1, 1] = 180;
104      K.Matrix[1, 2] = -120;
105      K.Matrix[1, 3] = 100;
106      K.Matrix[2, 0] = -20;
107      K.Matrix[2, 1] = -100 + 579 * Math.Cos(0.24 * t);
108      K.Matrix[2, 2] = 300;
109      K.Matrix[2, 3] = 20;
110      K.Matrix[3, 0] = 1.5 * Math.Sin(t);
111      K.Matrix[3, 1] = -9.8;
112      K.Matrix[3, 2] = 150;
113      K.Matrix[3, 3] = 11.5 * t * t * Math.Cos(t);
114      // End of K Matrix
115
116      // 隨時間變化的系統矩陣A, (12X12矩陣)(m = 4, r = 3)。
117      ReMatrix Ni = ~N;
118
119      A = ((-1.0 * Ni * M) & (-1.0 * Ni * C) & (-1.0 * Ni * K)) |
120          (Id & Zero & Zero) | (Zero & Id & Zero);
121
122      Console.WriteLine(" i = {0}    t = {1}  ", i, t);
123      Console.WriteLine(
124          "\n*** 因為計算特徵矩陣和模態矩陣的時間較長，顯示執行狀況: ↵
               \n");
125
126      // 隨時間變化的系統特徵矩陣 D, 模態矩陣 Q 。
127      D = (new EIG(A)).CxMatrixD;
128      Q = (new EIG(A)).CxMatrixQ;
129      Console.WriteLine("\n    *** 計算特徵值和特徵向量之後 : ***\n");
130
131      // 將時間轉爲複數值。
```

```
132         CxScalar cxScalar = new CxScalar(t, 0);
133         // 隨時間變化的特徵矩陣。
134         CxVal[i, 0] = new CxMatrix(cxScalar);
135         CxVal[i, 1] = D[0, 0];
136         CxVal[i, 2] = D[1, 1];
137         CxVal[i, 3] = D[2, 2];
138         CxVal[i, 4] = D[3, 3];
139         CxVal[i, 5] = D[4, 4];
140         CxVal[i, 6] = D[5, 5];
141         CxVal[i, 7] = D[6, 6];
142         CxVal[i, 8] = D[7, 7];
143         CxVal[i, 9] = D[8, 8];
144         CxVal[i, 10] = D[9, 9];
145         CxVal[i, 11] = D[10, 10];
146         CxVal[i, 12] = D[11, 11];
147 
148         // 隨時間變化的角頻率(實數值轉爲矩陣)。
149         double[,] tMatrix = { { t } };
150 
151         ReVal[i, 0] = (ReMatrix)tMatrix;
152         ReVal[i, 1] = D[0, 0].Im;
153         ReVal[i, 2] = D[1, 1].Im;
154         ReVal[i, 3] = D[2, 2].Im;
155         ReVal[i, 4] = D[3, 3].Im;
156         ReVal[i, 5] = D[4, 4].Im;
157         ReVal[i, 6] = D[5, 5].Im;
158         ReVal[i, 7] = D[6, 6].Im;
159         ReVal[i, 8] = D[7, 7].Im;
160         ReVal[i, 9] = D[8, 8].Im;
161         ReVal[i, 10] = D[9, 9].Im;
162         ReVal[i, 11] = D[10, 10].Im;
163         ReVal[i, 12] = D[11, 11].Im;
164 
165     }
166 
167     Console.WriteLine("\n***  時間和特徵值(有十二組)，合計十三組複數值  ***");
168     Console.WriteLine("\n{0}\n\n", new PR(CxVal));
169 
170     Console.WriteLine("\n***  特徵值矩陣的虛數值即角頻率  ***");
171     Console.WriteLine("       時間 t      ....   十二個角頻率 ");
172     Console.WriteLine("\n{0}\n", new PR(ReVal));
173 
174     // 轉爲序列方式，以便使用python程式繪圖。
175     Console.WriteLine("\n時間序列:   t\n{0}\n", new PR4(ReVal, 0));
176     Console.WriteLine("\n角頻率序列:w0\n{0}\n", new PR4(ReVal, 1));
177     Console.WriteLine("\n角頻率序列:w1\n{0}\n", new PR4(ReVal, 2));
```

```
178 Console.WriteLine("\n角頻率序列:w2\n{0}\n", new PR4(ReVal, 3));
179 Console.WriteLine("\n角頻率序列:w3\n{0}\n", new PR4(ReVal, 4));
180 Console.WriteLine("\n角頻率序列:w4\n{0}\n", new PR4(ReVal, 5));
181 Console.WriteLine("\n角頻率序列:w5\n{0}\n", new PR4(ReVal, 6));
182 Console.WriteLine("\n角頻率序列:w6\n{0}\n", new PR4(ReVal, 7));
183 Console.WriteLine("\n角頻率序列:w7\n{0}\n", new PR4(ReVal, 8));
184 Console.WriteLine("\n角頻率序列:w8\n{0}\n", new PR4(ReVal, 9));
185 Console.WriteLine("\n角頻率序列:w9\n{0}\n", new PR4(ReVal, 10));
186 Console.WriteLine("\n角頻率序列:w10\n{0}\n", new PR4(ReVal, 11));
187 Console.WriteLine("\n角頻率序列:w11\n{0}\n", new PR4(ReVal, 12));
188
189 /*輸出結果:
190
191 ***  時間和特徵值(有十二組), 合計十三組複數值  ***
192
193  0.00000 +  0.00000i, -11.77658 +   0.00000i,  -3.74980 +          ➲
       0.00000i,
194 -2.20404 +  2.54725i,  -2.20404 -   2.54725i,   2.97024 +          ➲
      0.49164i,
195  2.97024 -  0.49164i,   1.50230 +   2.28138i,   1.50230 -          ➲
       2.28138i,
196 -0.03875 +  2.53751i,  -0.03875 -   2.53751i,  -1.03565 +          ➲
      1.15840i,
197      .
198      .
199      .
200 */
```