```
// *** 時頻數值計算(Time-Frequency Numerical Computations) ***

// 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)
// 由齊次多空間維度之微分方程式，求得系統矩陣(狀態矩陣A)，進而求
// 得A = Q * D * Qi。D，Q，Qi 分別為系統特徵值、特徵向量、逆特徵
// 向量矩陣，進而求得訊號響應值 [y''|y'|y] = Hexp(D, Q, t) * d +
// [y''|y'|y]p , 稱 Hexp(D, Q, t)為轉換矩陣，d是由初始值或是
// 邊界值而定的係數向量，兩者均為複數矩陣和複數向量。
// 本求解法可對應於Laplace、Fourier、Z Transform或是捲積積分法等等。

using System;
using Matrix_0;

namespace ConsoleApp49
{
    internal class Program
    {
        static void Main(string[] args)
        {



// 空間維度有m個自由度。
int m = 4;
// 微分方程式有r個階度(Order)。
int r = 3;

// 建構初始矩陣 N, M、C、K、O、I。
ReMatrix N = (new Zero(m)).GetMatrix;
ReMatrix M = (new Zero(m)).GetMatrix;
ReMatrix C = (new Zero(m)).GetMatrix;
ReMatrix K= (new Zero(m)).GetMatrix;
ReMatrix O = (new Zero(m)).GetMatrix;
ReMatrix I = (new Iden(m)).GetMatrix;

// 建構系統矩陣 A (m x r)X(m x r)
ReMatrix A = (new Zero(m * r)).GetMatrix;
CxMatrix D = (new Zero(m * r)).GetMatrix;
CxMatrix Q = (new Zero(m * r)).GetMatrix;

// 狀態響應。速度，變位，加速度。(Step = 0.001秒，共計 t = 0.05秒)
double step = 0.001;
int iRow = (int)(0.05 / step + 1);

// 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。
```

```
47  int iColD = m * r + 1;
48  CxMatrix CxVal = new CxMatrix(iRow, iColD);
49  ReMatrix ReVal = new ReMatrix(iRow, iColD);
50
51  for (int i = 0; i != iRow; i++)
52  {
53      double t = step * i;
54
55      // 建構 N、M、C、K 變數矩陣。
56      N.Matrix[0, 0] = -2.7 * t * t * Math.Sin(1.3 * t);
57      N.Matrix[0, 1] = -5.5;
58      N.Matrix[0, 2] = 0;
59      N.Matrix[0, 3] = 5.5;
60      N.Matrix[1, 0] = 3.5;
61      N.Matrix[1, 1] = -8.5;
62      N.Matrix[1, 2] = -9.8 * t * t;
63      N.Matrix[1, 3] = -4.8;
64      N.Matrix[2, 0] = 6.7;
65      N.Matrix[2, 1] = 27.9;
66      N.Matrix[2, 2] = 8.5;
67      N.Matrix[2, 3] = -20.5 * t * t * Math.Cos(1.9 * t);
68      N.Matrix[3, 0] = -1.5 * t * Math.Cos(1.9 * t);
69      N.Matrix[3, 1] = 4.8;
70      N.Matrix[3, 2] = 0;
71      N.Matrix[3, 3] = 1.5 * t * t * t;
72      //  End of N
73
74      M.Matrix[0, 0] = 19;
75      M.Matrix[0, 1] = -1.5;
76      M.Matrix[0, 2] = -2 + 13.3 * Math.Sin(0.85 * t);
77      M.Matrix[0, 3] = 1.1;
78      M.Matrix[1, 0] = -1;
79      M.Matrix[1, 1] = 15;
80      M.Matrix[1, 2] = 0;
81      M.Matrix[1, 3] = 1.3;
82      M.Matrix[2, 0] = -10 - 2.7 * Math.Cos(1.3 * t);
83      M.Matrix[2, 1] = -3;
84      M.Matrix[2, 2] = 27;
85      M.Matrix[2, 3] = 4.5;
86      M.Matrix[3, 0] = 5.5;
87      M.Matrix[3, 1] = 2.7;
88      M.Matrix[3, 2] = -2.3 * t;
89      M.Matrix[3, 3] = -3.5 * t * t;
90      // End of M
91
92      C.Matrix[0, 0] = 35;
93      C.Matrix[0, 1] = -1 - 13.2 * Math.Sin(0.35 * t);
```

```
 94      C.Matrix[0, 2] = -0.5;
 95      C.Matrix[0, 3] = 2.5;
 96      C.Matrix[1, 0] = -1.5;
 97      C.Matrix[1, 1] = 40;
 98      C.Matrix[1, 2] = -1.5;
 99      C.Matrix[1, 3] = 0;
100      C.Matrix[2, 0] = -1.2 + 22.5 * Math.Cos(1.95 * t);
101      C.Matrix[2, 1] = -1.5;
102      C.Matrix[2, 2] = 75;
103      C.Matrix[2, 3] = 0;
104      C.Matrix[3, 0] = -27.5;
105      C.Matrix[3, 1] = 18.3;
106      C.Matrix[3, 2] = 9.5;
107      C.Matrix[3, 3] = -50.9 * t * Math.Sin(2.5 * t);
108      // End of C
109
110      K.Matrix[0, 0] = 60;
111      K.Matrix[0, 1] = -8;
112      K.Matrix[0, 2] = -2 - 332 * Math.Sin(1.37 * t);
113      K.Matrix[0, 3] = -2.7;
114      K.Matrix[1, 0] = -16;
115      K.Matrix[1, 1] = 180;
116      K.Matrix[1, 2] = -120;
117      K.Matrix[1, 3] = 100;
118      K.Matrix[2, 0] = -20;
119      K.Matrix[2, 1] = -100 + 579 * Math.Cos(0.24 * t);
120      K.Matrix[2, 2] = 300;
121      K.Matrix[2, 3] = 20;
122      K.Matrix[3, 0] = 1.5 * Math.Sin(t);
123      K.Matrix[3, 1] = -9.8;
124      K.Matrix[3, 2] = 150;
125      K.Matrix[3, 3] = 11.5 * t * t * Math.Cos(t);
126      // End of K
127
128      // 隨時間變化的系統(狀態)矩陣 A , A 矩陣為12X12的實數矩陣( m =  ↵
           4, r = 3)。
129      ReMatrix Ni = ~N;
130      A = (-1.0 * Ni * M) & (-1.0 * Ni * C) & (-1.0 * Ni * K);
131      A = A | (I & 0 & 0) | (0 & I & 0);
132
133      Console.WriteLine(" i = {0}    t = {1}  ", i, t);
134      Console.WriteLine("計算特徵值和特徵向量矩陣之前 : ");
135      // 隨時間變化的系統特徵值矩陣 D , 特徵向量 Q 。
136      D = (new EIG(A)).CxMatrixD;
137      Q = (new EIG(A)).CxMatrixQ;
138      Console.WriteLine("   *** 計算特徵值和特徵向量之後 : **");
139
```

```
140     // 將時間轉爲複數值。
141     CxScalar cxScalar = new CxScalar(t, 0);
142     // 隨時間變化的特徵值矩陣。
143     CxVal[i, 0] = new CxMatrix(cxScalar);
144     CxVal[i, 1] = D[0, 0];
145     CxVal[i, 2] = D[1, 1];
146     CxVal[i, 3] = D[2, 2];
147     CxVal[i, 4] = D[3, 3];
148     CxVal[i, 5] = D[4, 4];
149     CxVal[i, 6] = D[5, 5];
150     CxVal[i, 7] = D[6, 6];
151     CxVal[i, 8] = D[7, 7];
152     CxVal[i, 9] = D[8, 8];
153     CxVal[i, 10] = D[9, 9];
154     CxVal[i, 11] = D[10, 10];
155     CxVal[i, 12] = D[11, 11];
156
157     // 隨時間變化的角頻率(實數值轉爲矩陣)。
158     double[,] tMatrix = { { t } };
159
160     ReVal[i, 0] = (ReMatrix)tMatrix;
161     ReVal[i, 1] = D[0, 0].Im;
162     ReVal[i, 2] = D[1, 1].Im;
163     ReVal[i, 3] = D[2, 2].Im;
164     ReVal[i, 4] = D[3, 3].Im;
165     ReVal[i, 5] = D[4, 4].Im;
166     ReVal[i, 6] = D[5, 5].Im;
167     ReVal[i, 7] = D[6, 6].Im;
168     ReVal[i, 8] = D[7, 7].Im;
169     ReVal[i, 9] = D[8, 8].Im;
170     ReVal[i, 10] = D[9, 9].Im;
171     ReVal[i, 11] = D[10, 10].Im;
172     ReVal[i, 12] = D[11, 11].Im;
173
174 }
175
176 Console.WriteLine("\n***  時間和特徵值(有十二組)，合計十三組複數值  ***");
177 Console.WriteLine("\n{0}\n\n", new PR(CxVal));
178
179 Console.WriteLine("\n***  特徵值矩陣的虛數值即角頻率  ***");
180 Console.WriteLine("       時間 t      ....   十二個角頻率 ");
181 Console.WriteLine("\n{0}\n", new PR(ReVal));
182
183 // 轉爲序列方式，以便使用python程式繪圖。
184 Console.WriteLine("\n時間序列:   t\n{0}\n", new PR4(ReVal, 0));
185 Console.WriteLine("\n角頻率序列:w0\n{0}\n", new PR4(ReVal, 1));
```

```
186  Console.WriteLine("\n角頻率序列:w1\n{0}\n", new PR4(ReVal, 2));
187  Console.WriteLine("\n角頻率序列:w2\n{0}\n", new PR4(ReVal, 3));
188  Console.WriteLine("\n角頻率序列:w3\n{0}\n", new PR4(ReVal, 4));
189  Console.WriteLine("\n角頻率序列:w4\n{0}\n", new PR4(ReVal, 5));
190  Console.WriteLine("\n角頻率序列:w5\n{0}\n", new PR4(ReVal, 6));
191  Console.WriteLine("\n角頻率序列:w6\n{0}\n", new PR4(ReVal, 7));
192  Console.WriteLine("\n角頻率序列:w7\n{0}\n", new PR4(ReVal, 8));
193  Console.WriteLine("\n角頻率序列:w8\n{0}\n", new PR4(ReVal, 9));
194  Console.WriteLine("\n角頻率序列:w9\n{0}\n", new PR4(ReVal, 10));
195  Console.WriteLine("\n角頻率序列:w10\n{0}\n", new PR4(ReVal, 11));
196  Console.WriteLine("\n角頻率序列:w11\n{0}\n", new PR4(ReVal, 12));
197
198
199
200          }
201      }
202  }
203  // 輸出結果和圖表，參見儲存庫中的檔案。
204
```