3.3　反復法による固有値

（1）べき乗法（Power Iteration Method）

■考え方

n 行 n 列の行列 $\boldsymbol{A}$ の固有値と固有ベクトルを$(\lambda_i, \boldsymbol{x}_i)$ とし，

$$|\lambda_1| < |\lambda_2| < \cdots < |\lambda_n|$$

の順に並んでいるものとします。

固有ベクトルの任意の組は，正規直交基底となりますから，任意のベクトルは次のように表すことができます。

$$\boldsymbol{u} = c_1\boldsymbol{x}_1 + c_2\boldsymbol{x}_2 + \cdots + c_2\boldsymbol{x}_2$$

両辺に $\boldsymbol{A}$ を乗じると，

$$\begin{aligned}\boldsymbol{Au} &= c_1\boldsymbol{Ax}_1 + c_2\boldsymbol{Ax}_2 + \cdots + c_n\boldsymbol{Ax}_n \\ &= c_1\lambda_1\boldsymbol{x}_1 + c_2\lambda_2\boldsymbol{x}_2 + \cdots + c_n\lambda_n\boldsymbol{x}_n\end{aligned}$$

この処理を k 回繰り返すと

$$\begin{aligned}\boldsymbol{A}^k\boldsymbol{u} &= c_1{\lambda_1}^k\boldsymbol{x}_1 + c_2{\lambda_2}^k\boldsymbol{x}_2 + \cdots + c_{n-1}{\lambda_{n-1}}^k\boldsymbol{x}_{n-1} + c_n{\lambda_n}^k\boldsymbol{x}_n \\ &= {\lambda_n}^k\left\{c_1\left(\frac{\lambda_1}{\lambda_n}\right)^k\boldsymbol{x}_1 + c_2\left(\frac{\lambda_2}{\lambda_n}\right)^k\boldsymbol{x}_2 + \cdots + c_{n-1}\left(\frac{\lambda_{n-1}}{\lambda_n}\right)^k\boldsymbol{x}_{n-1} + c_n\boldsymbol{x}_n\right\}\end{aligned}$$

k を大きくしていくと，$|\lambda_1| < |\lambda_2| < \cdots < |\lambda_{n-1}| < |\lambda_n|$ ですから，

$$\left(\frac{\lambda_1}{\lambda_n}\right)^k \to 0, \quad \left(\frac{\lambda_2}{\lambda_n}\right)^k \to 0, \quad \cdots, \quad \left(\frac{\lambda_{n-1}}{\lambda_n}\right)^k \to 0$$

すなわち，$\boldsymbol{A}^k\boldsymbol{u} \to {\lambda_n}^k c_n\boldsymbol{x}_n$ となりますので，$\boldsymbol{A}^{k+1}\boldsymbol{u}$ と $\boldsymbol{A}^k\boldsymbol{u}$ の比は λ_n に近づくことになります。

■処理手順

べき乗法では，以下のような手順で固有値を求めます。

① 初期値として，最大成分が 1 であるような列ベクトルを設定します。例えば，以下のようなベクトルです。

$$\boldsymbol{x}^{(0)} = (1 \quad 0 \quad \cdots \quad 0 \quad 0)$$

② 以下の計算を行います。

$$\boldsymbol{x}^{(k)} = \boldsymbol{Ax}^{k-1} \quad (k = 1, 2, \cdots)$$

③ $\lambda_n = \left|\boldsymbol{x}^{(k)}\right|$ とし，$\boldsymbol{x}^{(k)} = \dfrac{\boldsymbol{x}^{(k)}}{\lambda_n}$ として単位ベクトル化します。

④ 次のような収束判定を行い，収束していなければ②から④を繰り返します。

$$\frac{\left|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\right|}{\left|\boldsymbol{x}^{(k)}\right|} < \varepsilon$$

■シートの定義

確認のために，$\lambda\boldsymbol{x}$, $\boldsymbol{Ax}$ を計算する式定義を入れておくほうがよいでしょう。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	A	1	2	3	4									べき乗法による最大固有値	
2	1	4	2	1	3										
3	2	2	3	1	2										
4	3	1	1	2	1										
5	4	3	2	1	1										
6		（Aは対称行列である）													
7														λx	Ax
8				べき乗法による最大固有値										0.0000	0.0000
9														0.0000	0.0000
10														0.0000	0.0000
11														0.0000	0.0000

表 3-1 べき乗法のプログラム

```
Private A() As Double
Private D() As Double
Private X() As Double
Function べき乗データ設定()
 N = 4
 ReDim A(N, N), X(N)
 With Worksheets("Sheet1")
  For i = 1 To N
   For j = 1 To N
    A(i, j) = Val(.Cells(i + 1, j + 1))
   Next
  Next
 End With
 べき乗データ設定 = N
End Function
Sub べき乗法結果設定(X, λ, N)
  With Worksheets("Sheet1")
     .Cells(2, 14) = λ
     For i = 1 To N
       .Cells(i + 1, 15) = X(i)
     Next
  End With
End Sub
Sub ボタン5_Click()
   N = べき乗データ設定()
   EPS = 0.000001
   べき乗法 A, EPS, X, λ, N
   べき乗法結果設定 X, λ, N
End Sub
Public Sub べき乗法(A, EPS, X, λ, N)
  Dim i, K1, K2, Iter
  Dim IterMax As Integer
  Dim T As Double
  Dim XTemp() As Double
  IterMax = 100
  ReDim XTemp(2, N)
  K1 = 2:  K2 = 1
  XTemp(K2, 1) = 1
  For i = 2 To N
    XTemp(K2, i) = 0
  Next
  E = EPS * 100
  Iter = 0
```

```
Do While Iter < IterMax And E > EPS
     Iter = Iter + 1:
     KK = K2: K2 = K1: K1 = KK
        ' K1：前回計算, K2:今回
     For i = 1 To N  ' 行列式の乗算
       T = 0
       For j = 1 To N
         T = T + A(i, j) * XTemp(K1, j)
       Next
       XTemp(K2, i) = T
     Next
     T = 0   'ノルム 1=1 による方法
     For i = 1 To N
       T = T + XTemp(K2, i) * XTemp(K2, i)
     Next
     λ = Sqr(T)
     If Abs(λ) < EPS Then
        '固有値が 0 のとき収束しない
        E = EPS * 100: Exit Do
     End If
     For i = 1 To N
       XTemp(K2, i) = XTemp(K2, i) / λ
     Next
     T1 = 0: T2 = 0
     For i = 1 To N
       DX = XTemp(K2, i) - XTemp(K1, i)
       T1 = T1 + DX * DX
       T2 = T2 + XTemp(K2, i) * XTemp(K2, i)
     Next
     If Abs(T2) < EPS Then
        '固有ベクトルが 0 のとき収束しない
        E = EPS * 100: Exit Do
     End If
     E = T1 / T2
  Loop
  For i = 1 To N
   X(i) = XTemp(K2, i)
  Next
  If E > EPS Then
     MsgBox "収束しません " & λ & _
        " 繰返し回数" & Iter & " E=" & E
  Else
     MsgBox λ & " 繰返し回数" & Iter & _
               " E=" & E
  End If
End Sub
```


(2)逆反復法(Inverse Iteration Method)

■考え方

正則対称行列 $\boldsymbol{A}$ の場合，逆行列を求め，

$$\boldsymbol{A}^{-1}\boldsymbol{x} = \frac{1}{\lambda}\boldsymbol{x}$$

として，べき乗法により最大固有値を求め，その逆数をとれば最小固有値となります。

したがって，適当な $\boldsymbol{x}^{(0)}$を初期値として，

$$\boldsymbol{x}^{(k)} = \boldsymbol{A}^{-1}\boldsymbol{x}^{(k-1)} \quad (k = 1, 2, \cdots)$$

を計算し，$\boldsymbol{x}^{(k)}$ の絶対値の逆数をとればよいことになります。

逆行列の計算のかわりに，以下の連立方程式を解いてもかまいません。

$$\boldsymbol{A}\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(k-1)} \quad (k = 1, 2, \cdots)$$

何回も同じ行列に対して連立方程式を解きますから，まず最初に三角分解すれば，計算量を減らすことができます。

■シートの定義

逆行列を求めた後，べき乗法を用いる方法を使いますので，べき乗法のシートに追加する形で定義します。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	A	1	2	3	4									べき乗法による最大固有値		べき乗法による最小固有値	
2	1	4	2	1	3									8.0262	0.6694		
3	2	2	3	1	2										0.5073		
4	3	1	1	2	1										0.2730		
5	4	3	2	1	1										0.4691		
6		(Aは対称行列である)															
7																	
8		べき乗法による最大固有値						べき乗法による最小固有値									
9																	
10														最大固有値確認		最小固有値確認	
11	逆行列	1	2	3	4									λx	Ax	λx	Ax
12	1	0.111	−0.33	−0.11	0.44									5.3726	5.3724	0.0000	0.0000
13	2	−0.33	0.5	−0.17	0.17									4.0716	4.0718	0.0000	0.0000
14	3	−0.11	−0.17	0.611	0.06									2.1916	2.1918	0.0000	0.0000
15	4	0.444	0.167	0.056	−0.7									3.7649	3.7648	0.0000	0.0000