الفصل السابع

مسائل القيم الذاتية

يحتوي هذا الفصل على:

- 1.7 القيم الذاتية لمصفوفة حقيقية متناسقة.
- 2.7 طريقة جاكوبي Jacobi Method .
- 3.7 المتجهات الذاتية.
- 4.7 مسائل قيم ذاتية عامة.

1.7 القيم الذاتية لمصفوفة حقيقية متناسقة

(Eigenvalues of a real symmetric matrix).

في كثير من المسائل الفيزيائية نجد أنفسنا مضطرين لحل المعادلات التالية:

$$\begin{pmatrix} a_{11}-\lambda & a_{12} & a_{13} \\ a_{21} & a_{22}-\lambda & a_{23} \\ a_{31} & a_{32} & a_{33}-\lambda \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \underset{\sim}{0}$$

أو

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

أو بشكل آخر:

$$A\underset{\sim}{x} = \lambda\, I\, \underset{\sim}{x} \qquad \ldots\ldots (1.7)$$

حيث:

A مصفوفة متناسقة.

$\underset{\sim}{x}$ متجه لمتغيرات مستقلة.

λ قيم ذاتية تمثل قيم فيزيائية مثل الترددات الطبيعية لأنظمة متذبذبة كالوتر المهتز.

ومسألة القيم الذاتية تتلخص في إيجاد قيم λ وهي القيم الذاتية و $\underset{\sim}{x}$ والتي تسمى بالمتجهات الذاتية. والطريقة الشائعة والمعتادة لإيجاد حلول هذه المسائل هي طريقة جاكوبي.

2.7 طريقة جاكوبي (Jacobi's Method)

تتلخص طريقة جاكوبي في الوصول بالمعادلة (1.7) إلى الشكل

$$\begin{pmatrix} b_{11} & 0 & 0 \\ 0 & b_{22} & 0 \\ 0 & 0 & b_{33} \end{pmatrix}\begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}\begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{pmatrix}$$

أو أن:

$$B\underset{\sim}{\bar{x}} = \underset{\sim}{\lambda}\,\underset{\sim}{\bar{x}} \qquad \ldots\ldots (2.7)$$

حيث B و $\underset{\sim}{\lambda}$ مصفوفتان قطريتان . لو تمكنا من ذلك عندئذ نرى أن الحلول هي $\underset{\sim}{\lambda}$ و $\underset{\sim}{\bar{x}}$ ؛ وحيث تكون $\underset{\sim}{\lambda}$ هنا هي B .

ولكي نستخدم طريقة جاكوبي دعنا نستذكر بعضاً من معلوماتنا حول تحويل الإحداثيات في المستوى . ففي المستوى وللتحويل. من الإحداثيات $\underset{\sim}{x}$ إلى $\underset{\sim}{\bar{x}}$ بدوران θ للمحاور نقوم بالعملية:

$$\underset{\sim}{x} = T\,\underset{\sim}{\bar{x}} \qquad \ldots\ldots (3.7)$$

حيث:

$$T = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \quad \text{و} \quad \underset{\sim}{\bar{x}} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix} \quad \text{و} \quad \underset{\sim}{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

هي مصفوفة الدوران. ولكن:

$$A\underset{\sim}{x} = \lambda\,\underset{\sim}{x} \qquad \ldots\ldots (4.7)$$

نعوض بالمعادلة (3.7) في المعادلة (4.7) لنحصل على:

$$AT\underset{\sim}{\bar{x}} = \lambda\, T\underset{\sim}{\bar{x}} \qquad \ldots\ldots (5.7)$$

بضرب (5.7) في T^T من اليسار واستعمال الخاصية أن $T^T\, T = 1$ نرى أن:

$$T^T AT\underset{\sim}{\bar{x}} = \lambda\, T^T\, T\underset{\sim}{\bar{x}} = \lambda\,\underset{\sim}{\bar{x}} \qquad \ldots\ldots (6.7)$$

الآن لكي نطبق طريقة جاكوبي ونصل إلى الهدف المطلوب وهو جعل المصفوفة على يسار المعادلة (6.7) قطرية ، نختار θ بحيث تكون $B = T^T AT$ قطرية؛ . ولكن:

$$B = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

ولكي تكون B قطرية يجب أن يكون الحد غير القطري مساوياً للصفر وهذا يعني أن:

$$a_{12}\left(\cos^2\theta - \sin^2\theta\right) + \cos\theta\sin\theta\left(a_{22} - a_{11}\right) = 0 \qquad \ldots\ldots (7.7)$$

و.هذه تعطي العلاقة:

$$\tan 2\theta = \frac{2a_{12}}{a_{11} - a_{22}} \qquad \ldots\ldots (8.7)$$

بينما تصبح B على الشكل:

$$B = \begin{pmatrix} a_{12}\cos^2\theta + 2a_{12}\cos\theta\sin\theta + a_{22}\sin^2\theta & 0 \\ 0 & a_{11}\sin^2\theta - 2a_{12}\cos\theta\sin\theta + a_{22}\cos^2\theta \end{pmatrix} \qquad \ldots\ldots (9.7)$$

مثال (1.7)

استخدم طريقة جاكوبي لجعل المصفوفة $A=\begin{pmatrix} 4 & 1 \\ 1 & 2 \end{pmatrix}$ قطرية . وعين القيم الذاتية للمصفوفة .

الحل:

بالرجوع للمعادلة (8.7) نرى أن θ التي تجعل A قطرية هي تلك التي تحقق المعادلة:

$$\tan 2\theta = \frac{2(1)}{4-2} = 1$$

أي أن $\theta = \pi/8$ ومنها ومن المعادلة (9.7) نحصل على:

$$B = \underset{\sim}{\lambda} = \begin{pmatrix} 4.414 & 0 \\ 0 & 1.586 \end{pmatrix}$$

آي أن القيم الذاتية للمصفوفة هي $\lambda_1 = 4.414$ و $\lambda_2 = 1.586$.

هذه هي طريقة جاكوبي في أبسط صورها حيث تم استخدامها في المستوى.

الآن لو أردنا العمل بمصفوفات من النوع (3×3) وهي المصفوفات التي عادة ما تواجهنا بالحياة العملية؛ فإننا نقوم بتحويلات متتابعة في المستوى لنتخلص من العناصر غير القطرية كلها . فمثلا تكون هذه التحويلات النوع:

$$T_1 = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

وهو يمثل دوراناً في المستوى $x-y$. و

$$T_2 = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}$$

ويمثل دوراناً في المستوى $x-z$. و

$$T_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}$$

ويمثل دوراناً في المستوى $y-z$.

ولتوضيح الخطوات المتبعة لجعل المصفوفة قطرية دعنا نقوم بحل المثال التالي:

مثال (2.7)

باستخدام طريقة جاكوبي أوجد القيم الذاتية للمصفوفة المتناسقة.

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

الحل:

1. لنأخذ أكبر عنصر غير قطري وهو هنا a_{12} أو a_{23} وكلاهما يساوي -1 عليه نأخذ a_{12} على أنه اكبر عنصر. (في القيمة)

2. نستخدم $T_1 = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$ ومنها نرى أن:

$$\tan 2\theta = \frac{-2}{2-2} = -\frac{2}{0} \to -\infty$$

أي أن: $\theta = -\pi/4$

وهكذا حصلنا على زاوية الدوران التي تجعل من العنصرين a_{12} و a_{21} يتلاشيان.

بينما تكون مصفوفة الدوران:

$$T_1 = \begin{pmatrix} \sqrt{2/2} & +\sqrt{2/2} & 0 \\ -\sqrt{2/2} & \sqrt{2/2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

3. نعين المصفوفة:

$$A_1 = T_1^T A T_1$$

وتصبح A الجديدة هي A_1 .

4. نعيد الخطوات 1-3 على المصفوفة A_1 ونتخلص من أكبر عنصر غير قطري؛ وهكذا حتى نحصل على A_m (m هو عدد مرات الدوران) تكون قطرية . وهكذا نحصل على القيم الذاتية.

لو قمنا بكل الخطوات سابقة الذكر فإنه بعد خمس دورانات يمكننا الحصول عل القيم الذاتية التالية:

$$\lambda_1 = 3.4142$$
$$\lambda_2 = 1.9998$$
$$\lambda_3 = 0.5859$$

ملاحظات

1) بالمثال السابق لاحظ أن $a_{31}=a_{13}=0$ وأنها متناسقة وعليه يجب التخلص من عنصرين فقط وهما a_{12} و a_{23}.

2) يعتقد أحدنا، من الملاحظة (1)، أنه يلزمنا دورانان لإنهاء العملية ولكن في الحقيقة وكما لاحظنا ليس هذا صحيحاً، فلقد احتجنا إلى خمس دورانات هنا . ويرجع السبب في ذلك إلى أنه بعد ضرب المصفوفات تتكون لدينا أرقام جديدة بمواضع أخرى بالرغم من اختفاء أحد العناصر غير القطرية.

ولكن لحسن الحظ أن كل دوران يجعل أكبر قيمة للعناصر غير القطرية تتضاءل وهكذا تدريجياً يتقارب A إلى الصيغة القطرية.

نعود الآن إلى الصيغة:

$$\tan 2\theta = \frac{2a_{ij}}{a_{ii}-a_{jj}} \qquad \ldots\ldots (10.7)$$

حيث a_{ij} هو أكبر عنصر غير قطري موجود بالصف i والعمود j ؛ وحيث أن:

$$\tan 2\theta = \frac{2\tan\theta}{1-\tan^2\theta} \qquad \ldots\ldots (11.7)$$

فإنه من المعادلتين (10.7) و (11 .7) نرى أن:

$$2a_{ij}\tan^2\theta + 2\left(a_{ii}-a_{jj}\right)\tan\theta - 2a_{ij} = 0$$

وهي معادلة من الدرجة الثانية في $\tan\theta$ وحلولها:

$$\tan\theta = \frac{-\left(a_{ii}-a_{jj}\right) \pm \sqrt{\left(a_{ii}-a_{jj}\right)^2 + 4a_{ij}^2}}{2a_{ij}}$$

بالضرب في المرافق نحصل على:

$$\tan\theta = \frac{\pm 2a_{ij}}{\left|a_{ii}-a_{jj}\right| + \sqrt{\left(a_{ii}-a_{jj}\right)^2 + 4a_{ij}^2}} \qquad \ldots\ldots (12.7)$$

و تعتمد الإشارة ($\pm$) على ما إذا كان a_{ii} أكبر من أو أصغر من a_{jj}. لاحظ أيضاً أن $-\pi/2 \le \theta \le \pi/2$.

الآن بحساب $\tan\theta$ من (12.7) نستطيع حساب $\cos\theta$ و $\sin\theta$ من العلاقتين:

$$\cos\theta = \left(1+\tan^2\theta\right)^{-1/2} \qquad \ldots\ldots (13.7)$$

$$\sin\theta = \cos\theta \cdot \tan\theta \qquad \ldots\ldots (14.7)$$

والمعادلات (12.7) ، (13.7) و (14.7) هي المستعملة عادة في الحسابات. خصوصا عند الاستنجاد بالحواسيب بدلا من المعادلة (10.7).

3.7 المتجهات الذاتية (Eigenvectors).

بالنسبة لحساب المتجهات الذاتية $\underset{\sim}{x}$؛ نلاحظ أنه توجد n من القيم الذاتية $\lambda_1, \lambda_2, \ldots\ldots, \lambda_n$ وذلك طبقاً لبعد المصفوفة $(n \times n)$. و إذا كان V هو مجموعة هذه المتجهات $\underset{\sim}{x}_1, \underset{\sim}{x}_2 \ldots\ldots\ldots, x_n$ في مصفوفة مربعة والمناظرة لمصفوفات التحويل $T_1, T_2, \ldots.$ و T_m وبحيث تقابل $\underset{\sim}{x}_1$ القيمة الذاتية λ_1 و $\underset{\sim}{x}_2$ القيمة الذاتية λ_2، و...... . وهكذا.

فإن:

$$AV = \underset{\sim}{\lambda} V$$

وهذا يعنى أن:

$$V^{-1} AV = \underset{\sim}{\lambda} \qquad \ldots\ldots (15.7)$$

وبمقارنة المعادلة (15.7) بالمعادلة (لاحظ أن $T_1^{-1} = T_1^T$ و $V^{-1} = V^T$):

$$B = T_m^{-1} \quad T_{m-1}^{-1} \ldots\ldots T_1^{-1} \quad AT_1 \ldots\ldots T_m \qquad \ldots\ldots (16.7)$$

نرى أن:

$$V = T_1\, T_2 \ldots\ldots T_m \qquad \ldots\ldots (17.7)$$

فعلى سبيل المثال لو رجعنا للمثال (1.7)فإن :

$$V = T_1 = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \text{ و } A = \begin{pmatrix} 4 & 1 \\ 1 & 2 \end{pmatrix}$$

أو أن:

$$V = T_1 = \begin{pmatrix} 0.924 & -0.383 \\ 0.383 & 0.924 \end{pmatrix}$$

و أن:

$$\underset{\sim}{x_1} = \begin{pmatrix} 0.924 \\ 0.383 \end{pmatrix} \text{ و } \qquad \underset{\sim}{x_2} = \begin{pmatrix} -0.383 \\ 0.924 \end{pmatrix}$$

بينما نجد بالنسبة للمثال (2.7) أن:

$$V = T_1 .....T_5 = \begin{pmatrix} 0.5 & 0.707 & 0.5 \\ -0.707 & 0 & 0.707 \\ 0.5 & -0.707 & 0.5 \end{pmatrix}$$

وهكذا يكون:

العمود الأول هو أول متجه ذاتي يماثل القيمة الذاتية 3.4142

العمود الثاني هو ثاني متجه ذاتي يماثل القيمة الذاتية 1.9998

العمود الثالث هو ثالث متجه ذاتي يماثل القيمة الذاتية 0.5859

مثال (3.7)

أستخدم طريقة جاكوبي لحساب القيم والمتجهات الذاتية للمصفوفة أسفله مستخدماً الحاسوب.

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

الحل:

حيث إن المصفوفة متناسقة ء كما سبق ذكره ، ولاستخدام طريقة جاكوبي نكتب البرنامج الفرعي JACOBI وبحيث نضع شرطا على الحدود غير القطرية لتتضاءل إلى الصفر بأن نطلب أن تكون قيمتها أقل من عدد صغير مثل $\varepsilon = 10^{-8}$ والبرنامج الفرعي هو:

SUBROUTINE JACOBI (N,Q,JVEC,M,V)

حيث N هو رتبة المصفوفة المتناسقة Q وهي أكبر من أو تساوي 2.

و V هي مصفوفة المتجهات الذاتية الناتجة.

و M هو عدد الدورانات المعمولة للحصول على المصفوفة القطرية، بينما $JVEC$ هو بارامتر يأخذ القيمة صفرا إذا أردنا طباعة القيم الذاتية بينما يأخذ القيم ±1,±2,............ إذا أردنا الحصول على القيم الذاتية والمتجهات الذاتية معاً.

في هذا المثال إذا كتبنا البرنامج الموضح بالشكل (1.7) والذي يشتمل على البرنامج الفرعي **JACOBI** فإنه يمكننا الحصول على النتائج الموضحة بالجدول (1.7) والذي من خلاله نرى أن القيم الذاتية الثلاث هي:

$$\lambda_1 = 3.41423$$
$$\lambda_2 = 2.000$$
$$\lambda_3 = 0.58579$$

وهي تقريباً نفس النتائج التي سبق وأن حصلنا عليها يدويا بالمثال السابق (2.7)، غير أنه لزم الأمر هنا تسعة دورانات للحصول على هذه القيم نظراً للدقة التي تم فرضها بالبرنامج.

```
C
C     THIS IS MAIN PROGRAM FOR ELGEN
C     VALUE PROBLEM IN THE FORM AX =LX
C
      DATA IREAD , IWRITE /5,2 /
      DIMENSION Q (12,12) , V(12,12)
C
C     READ IN THE ORDER OF MATRIX Q
C
      READ (IREAD,*) N
      DO 11 I= 1,N
   11 READ (*,*)(Q(I,J),J=1,N)
      WRITE(IWRITE,30)
   30 FORMAT('JVEC','MATRIX A'/)
```

```
      DO 32 I=1,N
      DO 31 J=1,N
   31 Q(J,I)=Q(I,J)
   32 WRITE (IWRITE,40)(Q(I,J),J=1,N)
   40 FORMAT(5F15.5)

      CALL JACOBI(N,Q,JVEC,M,V)

C     M NUMBER OF ROTATION

      WRITE(IWRITE,80)M
   80 FORMAT(//,  'THE NUMBER OF ROTATION=',I3//)
C
C     WRITE OUT THE ELGEN VALUES AND THE CORRESPONDING
C     ELGEN VECTORS
C
      DO 46 J=1,N
      WRITE(IWRITE,50)J,Q(J,J)
   50 FORMAT(/'EIGENVALUE(',I2,')=',F15.5)
      WRITE(IWRITE,60)
   60 FORMAT(/'EIGENVECTRS'/)
      DO 46 I=1,N
   46 WRITE(IWRITE,7)V(I,J)
    7 FORMAT(2X,F15.5)
      STOP
      END

C      SUPROGRAM FOR DIAGONALIION OF MATRIX Q
C      BY SUCCESSIVE ROTATIONS
C
      SUBROUTINE JACOBI(N,Q,JVEC,M,V)
      DIMENSION Q(12,12),V(12,12),X(12),IH(12)
      JVEC=+1
C
C  NEXT8 STATMENTS FOR SITING INITIAL VALUSOF MATRIX V
C
      IF(JVEC.EQ.0)GOTO 15
      DO 14 I=1,N
      DO 14 J=1,N
   14 V(I,J)=(I/J)*(J/I)
   15 M=0
C
C      NEXT 8STATMENTS SCAN FOR LARGEST OFF DIAG.
C      ELEMT. IN EACH ROW
C      X(I) CONTAINS LARGEST ELEMENT IN THE ROW
C      IH(I) HOLDS SECOND SUBSCRIPT DEFINING POSTION
C      OF ELEMENT
C
      MI=N-1

   DO 30 I=1,MI
   X(I)=0
   MJ=I+1
   DO 30 J=MJ,N
   IF(X(I).GT.ABS(Q(I,J)))GOTO 30
   X(I)=ABS(Q(I,J))
   IH(I)=J
30 CONTINUE
!
:   NEXT 7 STATEMENT FIND FOR MAXIMUMOF X(I)
:   FOR PIVOT ELEMENT
:
```

```
   40 DO 70 I=1,MI
      IF(I.LE.1)GOTO 60
      IF(XMAX.GT.X(I))GOTO 70
   60 XMAX=X(I)
      IP=I
      JP=IH(I)
   70 CONTINUE
C
C       NEXT TWO STATMENT TEST FOR XMAX , IF LESS
C       THAN 10**8, GO TO 1000
C
      EPSI=1.E-8
      IF (XMAX.LE.EPSI)GOTO 1000
      M=M+1
C
C    NEXT11STATMENT FOR COMPUTING,SIN,COS,Q(I,J),Q(J,J)
C
      IF(Q(IP,IP).GT.Q(JP,JP)) GOTO 151
      TANG=-2.*Q(IP,JP)/(ABS(Q(IP,IP)-Q(JP,JP))+SQRT((
     1Q(IP,IP)-Q(JP,JP))**2+4.*Q(IP,JP)**2))
      GOTO 160
  151 TANG=+2.*Q(IP,JP)/(ABS(Q(IP,IP)-Q(JP,JP))+SQRT((
     1Q(IP,IP)-Q(JP,JP))**2+4.*Q(IP,JP)**2))
  160 COSN=1./SQRT(1.+TANG**2)
      SINE1= COSN*TANG
      QII=Q(IP,IP)
      Q(IP,IP)=COSN**2*(QII+TANG*(2.*Q(IP,JP)+TANG*Q(JP,JP)))
      Q(JP,JP)=COSN**2*(Q(JP,JP)-TANG*(2.*Q(IP,JP)-TANG*QII))
      Q(IP,JP)=0
C
C       NEXT 4 STATMENT FOR PSEUDO RANK EIGENVALUES
C
      IF(Q(IP,IP).GE.Q(JP,JP))GOTO 153
      TEMP=Q(IP,IP)
      Q(IP,IP)=Q(JP,JP)
      Q(JP,JP)=TEMP
C
C       NEXT 6 STATMENT ADJUST SIN,COS,FORCOMPUTATION
C        OF Q(I,K),V(I)
C
      IF (SINE1.GE.0.)GOTO 155
      TEMP=-COSN
      GOTO 170
  155 TEMP=-COSN
  170 COSN=ABS(SINE1)
      SINE1=TEMP
  153 DO 350 I=1,MI
      If(((I.EQ.IP).OR.(I.EQ.JP)).OR.((IH(I).NE.IP).
     1AND.(IH(I).NE.JP)))GOTO 350
      K=IH(I)
      TEMP=Q(I,K)
      Q(I,K)=0
      MJ=I+1
      X(I)=0.
C
C       NEXT 5 STATMENT SEARCH IN DEPLETED ROW FOR
C       ROW MAXIMUM
C
      DO 320 J=MJ,N
      IF(X(I).GT.ABS(Q(I,J)))GOTO 320
      X(I)=ABS(Q(I,J))
```

```
      IH(I)=J
  320 CONTINUE
      Q(I,K)=TEMP
  350 CONTINUE
      X(IP)=0.
      X(JP)=0.
C
C      NEXT 30 STATMENT FOR CHANGING THE OTHER
C      ELEMENT OF Q
C
      DO 530 I=1,N
      IF(I.EQ.IP)GOTO 530
      IF(I.GT.IP) GOTO 420
      TEMP=Q(I,IP)
      Q(I,IP)=COSN*TEMP+SINE1*Q(I,JP)
      IF(X(I).GE.ABS(Q(I,IP))) GOTO 390
      X(I)=ABS(Q(I,IP))
      IH(I)=IP
  390 Q(I,JP)=-SINE1*TEMP+COSN* Q(I,JP)
      IF(X(I).GE.ABS(Q(I,JP))) GOTO 530
      X(I)=ABS(Q(I,JP))
      IH(I)=JP
      GOTO 530
  420 IF(I-JP)430,530,480
  430 TEMP=Q(IP,I)
      Q(IP,I)=COSN*TEMP+SINE1*Q(I,JP)
      IF(X(IP).GE.ABS(Q(IP,I)))GOTO 450
      X(IP)=ABS(Q(IP,I))
      IH(IP)=I
  450 Q(I,JP)=-SINE1*TEMP+COSN*Q(I,JP)
      IF(X(IP).GE.ABS(Q(I,JP)))GOTO 530
  480 TEMP=Q(IP,I)
      Q(IP,I)=COSN*TEMP+SINE1*Q(JP,I)
      IF(X(IP).GE.ABS(Q(IP,I)))GOTO 500
      X(IP)=ABS(Q(IP,I))
      IH(IP)=I
  500 Q(JP,I)=-SINE1*TEMP+COSN*Q(JP,I)
      IF(X(JP).GE.ABS(Q(JP,I))) GOTO 530
      X(JP)=ABS(Q(JP,I))
      IH(JP)=I
  530 CONTINUE
C
C   NEXT6 STATMENT TEST FOR COMPUTATION OF EIGENECTORS
C
      IF(JVEC.EQ.0) GOTO 40
      DO 550 I=1,N
      TEMP=V(I,IP)
      V(I,JP)=COSN*TEMP+SINE1*V(I,JP)
  550 V(I,IP)=SINE1*TEMP+COSN*V(I,JP)
      GOTO 40
 1000 RETURN
      END
```

الشكل (1.7) برنامج يحسب القيم والمتجهات الذاتي للمصفوفة

$$A=\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$ **باستخدام طريقة جاكوبي.**

```
JVECMATRIX A

             2.00000        -1.00000         0.00000
            -1.00000         2.00000        -1.00000
             0.00000        -1.00000         2.00000


THE NUMBER OF ROTATION =  9


EIGENVALUE(1)=  3.41421

EIGENVECTRS
                 0.50000
                -0.70711
                 0.50000

EIGENVALUE(2)=  2.00000

EIGENVECTRS
                 0.70711
                -0.00000
                -0.70711

EIGENVALUE(3)=  0.58579

EIGENVECTRS
                 0.50000
                -0.70711
                 0.50000
```

الجدول (1.7) نتائج البرنامج الموضح بالشكل (1.7)

نعطي كذلك نفس المسألة ولكن بلغة C و باستخدام بيئة التطوير البرمجية دلفي. وذلك بالأشكال (4.7) – (7.7).

```
#include<conio.h>
#include<iostream.h>
#include<math.h>

angle_fun();
trans_funz();
trans_funy();
trans_funx();
int j,h,i,l,k,p,q,R,g;
float A[10][10],B[10][10],b[10][10],v[10][10],v[10][10];
float angle,C=0.0,Vz[10][10],Vy[10][10],Vx[10][10];

main()
{
   clrscr();
   cout<<"\nEnter number of Rotations R =";
    cin>>R;
```

```
  for( j=0 ; j <3 ; j++)
 { for( i=0 ; i< 3 ; i++)
     { cout<<"\nEnter A["<<j<<"]["<<i<<"]= ";
    cin>>A[j][i];
    }  }
   cout<<"\nThe Array A[j][i]\n\n";
   for( j=0 ; j < 3 ; j++)
    { cout<<endl;
  for( i=0 ; i< 3 ; i++)
     { cout<<A[j][i]<<"\t\t";
    } }
    for (h=1 ; h<= R ; h++)
    {
    angle_fun();

     if (l+k == 1 )
     trans_funz();
     else
     {
     if( l+k == 2 )
     trans_funy();
     else
     trans_funx();
     }
   for( j=0 ; j < 3 ; j++)
    {
  for( i=0 ; i< 3 ; i++)
 A[j][i]=b[j][i];
 } }

   for( j=0 ; j < 3 ; j++)
    { cout<<"\nIenvalue["<<j<<"]="<<b[j][j]<<endl;
    cout<<"Igemvectrs:\n";
    for( i=0 ; i< 3 ; i++)

    {  cout<< v[i][j]<<endl;}}

         getche();
 return(0);
}
/*********************************************/
angle_fun()
{
float m=0,w,s;
 for( j=0 ; j < 3 ; j++)
    {   for( i=0 ; i< 3 ; i++)
     {   if ( i!=j)
     {   if( A[j][i] < 0)
     w = A[j][i] *(-1);
     else  w = A[j][i];
      if ( m < w )
      {   m = w; l=j; k=i;

     }}}}

     s = 2 * A[l][k]; w = A[l][l] - A[k][k];
     if( w == 0  && s < 0 )
         angle = - 3.14159265/4;
         else { if ( w == 0 && s > 0 )
         { angle = 3.14159265/4;
          } else
          { angle = (atan(s/w))/2;}}
 return(0);
}
/*****************************************************/
trans_funz()
{
float Tz[3][3]={cos(angle),-sin(angle),0,sin(angle),cos(angle),0,0,0,1};
float Tt[3][3]={cos(angle),sin(angle),0,-sin(angle),cos(angle),0,0,0,1};

    for( j=0 ; j < 3 ; j++)
     {p=0;
     for (g=0; g< 3;g++)
```

```
 {q=0;
 for( i=0 ; i< 3 ; i++)
 {C= C+Tt[j][i]*A[q][p];
  q=q+1;
   }
   B[j][g]=C ; C=0.0 ; p=p+1 ;
    }}
  for( j=0 ; j < 3 ; j++)
   {p=0;
   for (g=0; g< 3;g++)
   {q=0;
 for( i=0 ; i< 3 ; i++)
 {C= C+B[j][i]*Tz[q][p];
  q=q+1;
   }
   b[j][g]=C ; C=0.0 ; p=p+1 ;
    }}
  for( j=0 ; j < 3 ; j++)
   { for( i=0 ; i< 3 ; i++)
    { if ( h == 1)
    V[j][i]=Tz[j][i];
    else
    Vz[j][i]=Tz[j][i]; }}

    if(h != 1 )
    {
      for( j=0 ; j < 3 ; j++)
   {p=0;
   for (g=0; g< 3;g++)
   {q=0;
 for( i=0 ; i< 3 ; i++)
 {C= C+V[j][i]*Vz[q][p];
  q=q+1;
   }
   v[j][g]=C ; C=0.0 ; p=p+1 ;
    }}
    for (j=0;j<=3;j++)
   { for(i=0;i<=3;i++)
    V[j][i]=v[j][i];}}
return(0);
}
/*****************************************************/
 trans_funy()
{
float Ty[3][3]={cos(angle),0,-sin(angle),0,1,0,sin(angle),0,cos(angle)};
float Tt[3][3]={cos(angle),0,sin(angle),0,1,0,-sin(angle),0,cos(angle)};
   for( j=0 ; j < 3 ; j++)
   {p=0;
   for (g=0; g< 3;g++)
   {q=0;
 for( i=0 ; i< 3 ; i++)
 {C= C+Tt[j][i]*A[q][p];
  q=q+1;
   }
   B[j][g]=C ; C=0.0 ; p=p+1 ;
    }}
  for( j=0 ; j < 3 ; j++)
   {p=0;
   for (g=0; g< 3;g++)
   {q=0;
 for( i=0 ; i< 3 ; i++)
 {C= C+B[j][i]*Ty[q][p];
  q=q+1;
   }
   b[j][g]=C ; C=0.0 ; p=p+1 ;
    }}
     for( j=0 ; j < 3 ; j++)
   {   for( i=0 ; i< 3 ; i++)
    {  if ( h == 1)
    v[j][i]=Ty[j][i];
    else
    Vy[j][i]=Ty[j][i];}}
```

```
    if(h != 1 )
    {
      for( j=0 ; j < 3 ; j++)
   {p=0;
   for (g=0; g< 3;g++)
   {q=0;
 for( i=0 ; i< 3 ; i++)
 {C= C+v[j][i]*Vy[q][p];
  q=q+1;
   }
    v[j][g]=C ; C=0.0 ; p=p+1 ;
    }}
    for (j=0;j<=3;j++)
   { for(i=0;i<=3;i++)
    V[j][i]=v[j][i]; }}
return(0);
}
/*************************************************************/
trans_funx()
{
float Tx[3][3]={1,0,0,0,cos(angle),-sin(angle),0,sin(angle),cos(angle)};
float Tt[3][3]={1,0,0,0,cos(angle),sin(angle),0,-sin(angle),cos(angle)};
    for( j=0 ; j < 3 ; j++)
   {p=0;
   for (g=0; g< 3;g++)
   {q=0;
 for( i=0 ; i< 3 ; i++)
 {C= C+Tt[j][i]*A[q][p];
  q=q+1;
   }
   B[j][g]=C ; C=0.0 ; p=p+1 ;
    }}
    for( j=0 ; j < 3 ; j++)
   {p=0;
   for (g=0; g< 3;g++)
   {q=0;
 for( i=0 ; i< 3 ; i++)
 {C= C+B[j][i]*Tx[q][p];
  q=q+1;
   }
   b[j][g]=C ; C=0.0 ; p=p+1 ;
    }}
   for( j=0 ; j < 3 ; j++)
   { for( i=0 ; i< 3 ; i++)
    { if ( h == 1)
    V[j][i]=Tx[j][i];
    else

        Vx[j][i]=Tx[j][i];  }}

        if(h != 1 )
        {
          for( j=0 ; j < 3 ; j++)
       {p=0;
       for (g=0; g< 3;g++)
       {q=0;
     for( i=0 ; i< 3 ; i++)
     {C= C+V[j][i]*Vx[q][p];
      q=q+1;
       }
        v[j][g]=C ; C=0.0 ; p=p+1 ;
        }}
        for (j=0;j<=3;j++)
        {for(i=0;i<=3;i++)
        V[j][i]=v[j][i];}}
   return(0);
   }
   /***************************************************/
```

الشكل (4.7) – المثال (3.7) بلغة C.

```
The Array A[j][i]

2               -1              0
-1              2               -1
0               -1              2
Ienvalue[0]=3.414213
Igemvectrs:
0.5
-0.707107
0.5

Ienvalue[1]=0.585786
Igemvectrs:
0.5
0.707107
0.5

Ienvalue[2]=2
Igemvectrs:
-0.707107
7.640574e-08
0.707107
```

الشكل (5.7) نتائج المثال (3.7) بلغة C.

```
procedure Jacobi(N: integer; var Q : TNByN; var JVec :
real; var M : integer; var V : TNByN);
var
  X: array[1..12] of real;
  IH : array[1..12] of integer;
  i, j : integer;
  MI, MJ, IP, JP, k : integer;
  XMax, EPSI, Tang, Cosn, Sine, QII, Temp : real;

label L40;
begin
//--------Initialization-----------------------------
  for i := 1 to 12 do
  begin
    x[i] := 0; IH[i] := 0;
    for j := 1 to 12 do V[i,j] := 0;
  end;
//-------------------------------------------------------
  JVec := 3;

  if JVec <> 0 then
  begin
    for i := 1 to n do
      for j := 1 to n do
        V[i,j] := (i/j)*(j/i);
  end;
  //-----------------------------------
  M := 0;
  MI := n-1;
  //-----------------------------------
  for i := 1 to MI do
  begin
    X[i] := 0;
```

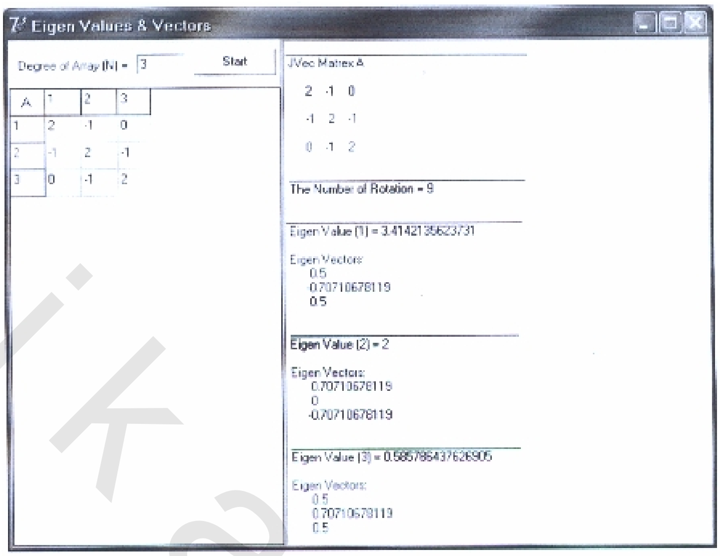
```
    MJ := i + 1;
    for j := MJ to n do
    begin
      if X[i] > Abs(Q[i,j]) then continue;
      X[i] := Abs(Q[i,j]);
      IH[i] := j          //
    end;

  end;
  //-----------------------------------
  L40:
  for i := 1 to MI do
  begin
    if i > 1 then if XMax > X[i] then continue;
    XMax := X[i];
    IP := I;
    JP := IH[i];
  end;
  //-----------------------------------

  EPSI := Power(10, -8);
  if XMax < EPSI  then exit;

  M := M + 1;
  //-------------------------------
  if Q[IP,IP] > Q[JP,JP] then
    Tang := 2.0 * Q[IP,JP]/(Abs(Q[IP,Ip] - Q[JP,JP]) +
SQrt(Power(Q[IP,IP]-Q[JP,JP],2)+4.0*Power(Q[IP,JP],2)))
  else
    Tang :=  -2.0 * Q[IP,JP]/(Abs(Q[IP,Ip] - Q[JP,JP]) +
SQrt(Power(Q[IP,IP]-Q[JP,JP],2)+4.0*Power(Q[IP,JP],2)));
  //-------------------------------

  Cosn := 1.0/Sqrt(1.0 + Power(Tang,2));
  Sine := Tang * Cosn;

  QII := Q[IP,IP];

  Q[IP,IP] := Power(Cosn,2) * (QII + Tang * (2.0*Q[IP,JP]
+ Tang * Q[JP,JP]));
  Q[JP,JP] := Power(Cosn,2) * (Q[JP,JP] - Tang *
(2.0*Q[IP,JP] - Tang * QII));

  Q[IP,JP] := 0;
  //-----------------------------------
  if Q[IP,IP] < Q[JP,JP] then
  begin
    Temp := Q[IP,IP];
    Q[IP,IP] := Q[JP,JP];
    Q[JP,JP] := Temp;
    if Sine >= 0 then Temp := -Cosn else Temp := Cosn;


  Cosn := Abs(sine);
  Sine := Temp;
```

```
  end;
  //-------------------------------------
  for i := 1 to MI do
  begin
    if (i = IP) or (i = JP) or (IH[i] <> IP) or (IH[i] <>
JP) then continue;
    k := IH[i];
    Temp := Q[i,k];
    Q[i,k] := 0;
    MJ := i  + 1;
    X[i] := 0;

    for j := MJ to n do
    begin
      if X[i] > Abs(Q[i,j]) then continue;
      X[i] := Abs(Q[i,j]);
      IH[i] := j;
    end;  //for j
    Q[i,k] := Temp;
  end; //for i
  //--------------------------------------

  X[IP] := 0; X[JP] := 0;
  //--------------------------------------

  for i := 1 to n do
  begin
    if i = IP then continue;
    if i <= IP then
    begin
      Temp := Q[i,IP];
      Q[i,IP] := Cosn * Temp + Sine * Q[i,JP];
      if X[i] < Abs(Q[i,IP]) then
      begin
        X[i] := Abs(Q[i,IP]);
        IH[i] := IP;
      end;
      Q[i,JP] := -Sine * Temp + Cosn * Q[i,JP];
      if X[i] >= Abs(Q[i,JP]) then continue;
      X[i] := Abs(Q[i,JP]);
      IH[i] := JP;
      continue;
    end;

    if (i - JP) < 0 then
    begin
      Temp := Q[IP,i];
      Q[IP,i] := Cosn * Temp + Sine * Q[i,JP];
      if X[IP] < Abs(Q[IP,i]) then
```

```
      begin
        X[IP] := Abs(Q[IP,i]);
        IH[IP] := i;
      end;
      Q[i,JP] := - Sine * temp + Cosn * Q[i,JP];
      if X[IP] >= Abs(Q[i,JP]) then continue;
    end else if (i - JP) = 0 then continue;
```

```
//     else if (i - JP) > 0 then
//     begin
//     end;
     Temp := Q[IP,i];
     Q[IP,i] := Cosn * Temp + Sine * Q[JP,i];
     if X[IP] < Abs(Q[IP,i]) then
     begin
       X[IP] := Abs(Q[IP,i]);
       IH[IP] := i
     end;
     Q[JP,i] := -sine * Temp + Cosn * Q[JP,i];
     if X[JP] >= Abs(Q[JP,i]) then continue;
     X[JP] := Abs(Q[JP,i]);
     IH[JP] := i

   end; //for i
//-----------------------------------------------------------

   if JVec = 0 then goto L40;

   for i := 1 to n do
   begin
     Temp := V[i,IP];
     V[i,JP] := Cosn * Temp + Sine * V[i,JP];
     V[i,IP] := sine * Temp + Cosn * V[i,JP];
   //  showmessage(inttostr(i));
   end;

   goto L40

end;
```

الشكل (6.7) – المثال (3.7) بلغة دلفي.



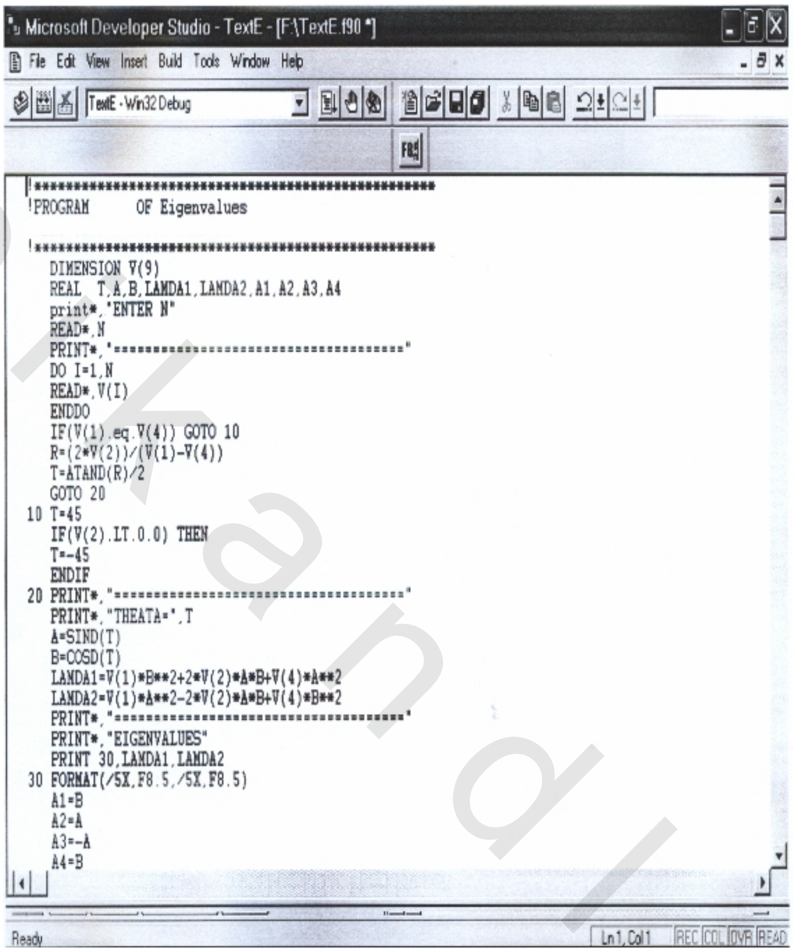
الشكل (7.7) – نتائج المثال (3.7) بلغة دلفي.

مثال (4.7)

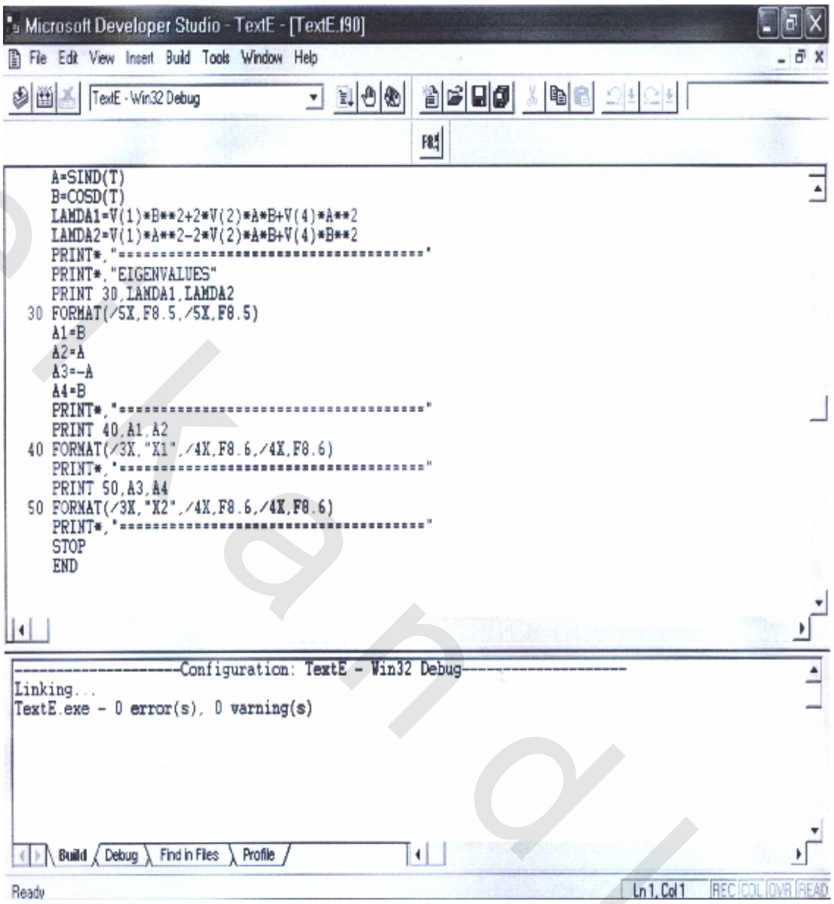
أكتب برنامجاً حاسوبيا يحسب القيم والمتجهات الذاتية للمصفوفة $A=\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$ باستخدام طريقة جاكوبي.

الحل:

نستخدم طريقة جاكوبي كما سبق شرحه ثم نكتب الخوارزمية فالبرنامج. في الأشكال (8.7) – (11.7) نعطى هذه الحسابات بلغتي فورتران وبيسك المرئية.



```
!****************************************************
!PROGRAM      OF Eigenvalues

!****************************************************
   DIMENSION V(9)
   REAL  T,A,B,LAMDA1,LAMDA2,A1,A2,A3,A4
   print*,"ENTER N"
   READ*,N
   PRINT*,"===================================="
   DO I=1,N
   READ*,V(I)
   ENDDO
   IF(V(1).eq.V(4)) GOTO 10
   R=(2*V(2))/(V(1)-V(4))
   T=ATAND(R)/2
   GOTO 20
10 T=45
   IF(V(2).LT.0.0) THEN
   T=-45
   ENDIF
20 PRINT*,"===================================="
   PRINT*,"THEATA=",T
   A=SIND(T)
   B=COSD(T)
   LAMDA1=V(1)*B**2+2*V(2)*A*B+V(4)*A**2
   LAMDA2=V(1)*A**2-2*V(2)*A*B+V(4)*B**2
   PRINT*,"===================================="
   PRINT*,"EIGENVALUES"
   PRINT 30,LAMDA1,LAMDA2
30 FORMAT(/5X,F8.5,/5X,F8.5)
   A1=B
   A2=A
   A3=-A
   A4=B
```



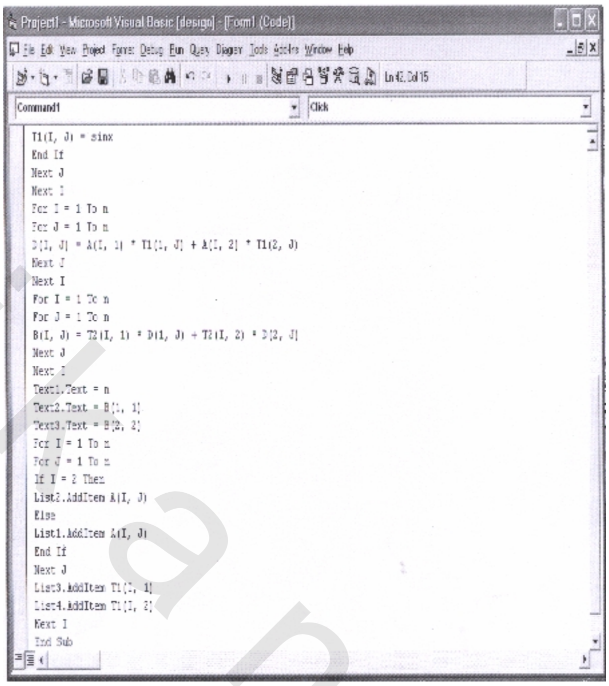
```
Microsoft Developer Studio - TextE - [TextE.f90]
File Edit View Insert Build Tools Window Help
TextE - Win32 Debug

      A=SIND(T)
      B=COSD(T)
      LAMDA1=V(1)*B**2+2*V(2)*A*B+V(4)*A**2
      LAMDA2=V(1)*A**2-2*V(2)*A*B+V(4)*B**2
      PRINT*,"===================================="
      PRINT*,"EIGENVALUES"
      PRINT 30,LAMDA1,LAMDA2
   30 FORMAT(/5X,F8.5,/5X,F8.5)
      A1=B
      A2=A
      A3=-A
      A4=B
      PRINT*,"===================================="
      PRINT 40,A1,A2
   40 FORMAT(/3X,"X1",/4X,F8.6,/4X,F8.6)
      PRINT*,"===================================="
      PRINT 50,A3,A4
   50 FORMAT(/3X,"X2",/4X,F8.6,/4X,F8.6)
      PRINT*,"===================================="
      STOP
      END

--------------------Configuration: TextE - Win32 Debug--------------------
Linking...
TextE.exe - 0 error(s), 0 warning(s)

Build  Debug  Find in Files  Profile
Ready                                   Ln 1, Col 1   REC COL OVR READ
```

الشكل (**8.7**) – المثال (**4.7**) بلغة فورتران.

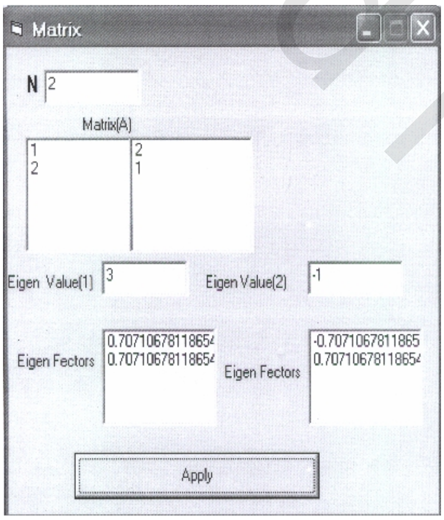
```
C:\MSDEV\Projects\TextE.exe تحديد
ENTER N
4
========================================
1
2
2
1
========================================
THEATA=        45.000000
========================================
EIGENVALUES

     3.00000
    -1.00000
========================================

  X1
     .707107
     .707107
========================================

  X2
    -.707107
     .707107
========================================
Stop - Program terminated.

Press any key to continue
```

الشكل (9.7) – نتائج المثال (4.7) بلغة فورتران.

```
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
Ln 42, Col 15
Command1                                  Click

Private Sub Command1_Click()
n = InputBox("Input Degree of Array")
Dim A(100, 100)
Dim T1(100, 100)
Dim T2(100, 100)
Dim D(100, 100)
Dim B(100, 100)
For I = 1 To n
For J = 1 To n
A(I, J) = InputBox("Value of Matrix")
Next J
Next I
tanx = (2 * A(2, 1)) / ((Abs(A(2, 2) - A(1, 1))) + Sqr(((A(2, 2) - A(1, 1)) ^ 2) + (4 * A(2, 1) ^ 2)))
cosx = (1 + tanx ^ 2) ^ (-0.5)
sinx = tanx * cosx
For I = 1 To n
For J = 1 To n
If I = J Then
T2(I, J) = cosx
T1(I, J) = cosx
End If
If I < J Then
T2(I, J) = sinx
T1(I, J) = -sinx
End If
If I > J Then
T2(I, J) = -sinx
T1(I, J) = sinx
End If
Next J
Next I
```



الشكل (10.7) – المثال (4.7) بلغة بيسك المرئية.



الشكل (11.7) – نتائج المثال (4.7) بلغة بيسك المرئية.

4.7 مسائل قيم ذاتية عامة

حتى الآن ركزنا كل دراستنا على حل مسائل القيم الذاتية من النوع:

$$A\underset{\sim}{x} = \lambda B \underset{\sim}{x} \qquad \ldots\ldots (18.7)$$

حيث B هي مصفوفة الوحدة . في هذا البند نتطرق إلى الحالة الأعم وندرس المسألة عندما تكون B قطرية و عندما تكون B متناسقة.

الحالة الأولى : B قطرية (B is diagonal)

في هذه الحالة يمكن أن نكتب B على النحو:

$$B = G^T G = GG \qquad \ldots\ldots (19.7)$$

حيث G قطرية، وعليه فإن:

$$g_{ii} = \sqrt{b_{ii}}$$

وبالتعويض من (19.7) في (18.7) نرى أن:

$$A\underset{\sim}{x} = \lambda GG \underset{\sim}{x} \qquad \ldots\ldots (20.7)$$

بضرب (20.7) في G^{-1} نجد أن:

$$G^{-1} AG^{-1} Gx = \lambda G \underset{\sim}{x}$$

فإذا فرضنا أن $Q = G^{-1} AG^{-1}$ و $y = G\underset{\sim}{x}$ لوجدنا أن:

$$Q\underset{\sim}{y} = \lambda \underset{\sim}{y} \qquad \ldots\ldots (21.7)$$

والمعادلة (21.7) هي نفس المعادلة التي تعرضنا لها بالبنود السابقة لإيجاد القيم والمتجهات الذاتية. إضافة إلى ذلك نرى أن القيم الذاتية المطلوبة هي تلك التي تجعل من Q قطرية.

أما المتجهات الذاتية فنحصل عليها من $\underset{\sim}{y}$ بالضرب في G^{-1} أي أن:

$$\underset{\sim}{x} = G^{-1}\,\underset{\sim}{y} \qquad \ldots\ldots (22.7)$$

وهكذا نرى في هذه الحالة أن حل المعادلة (18.7) يكمن في حل المعادلة (21.7) وحساب (22.7).

مثال (5.7)

إذا كانت مسألة القيم الذاتية الناتجة عن دراسة الذبذبات الحرة لنظام يتكون من ثلاث كتل $m, 2m, 3m$ تتلخص في إيجاد حل المعادلة:

$$\begin{pmatrix} 2k & -k & 0 \\ -k & 2k & -k \\ 0 & -k & 2k \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} m\omega^2 & 0 & 0 \\ 0 & 2m\omega^2 & 0 \\ 0 & 0 & 3m\omega^2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \qquad \ldots\ldots (23.7)$$

حيث ω هو التردد و $\underset{\sim}{x}$ متجه الإزاحة و $k = {}^{s}\!/\!_{\ell}$ حيث s هو الشد في الخيط و 4ℓ طول الخيط. احسب القيم والمتجهات الذاتية لهذه المسألة.

الحل:

نضع $\lambda = \dfrac{m\omega^2}{k}$ لتصبح المعادلة على النحو:

$$A\underset{\sim}{x} = \lambda\, B\underset{\sim}{x}$$

حيث:

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

و

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

وبهذا تصبح المعادلة من النوع (14.7) [الحالة الأولى]. نكتب برنامجاً يحسب G ومن ثم Q (وهنا نستخدم برنامجاً لضرب المصفوفات) ثم نلجأ إلى JACOBI والذي من خلاله نحسب القيم الذاتية لـ Q وبالتالي القيم الذاتية لـ A كما نحسب $\underset{\sim}{y}$ $\underset{\sim}{x} = G^{-1}$.
إذا قمنا بذلك نحصل على النتائج التالية:

$$\lambda_1 = 2.38742 \quad , \quad \underset{\sim}{x_1} = \begin{pmatrix} 0.87134 \\ -0.33758 \\ 0.06539 \end{pmatrix}$$

$$\lambda_1 = 1.0 \quad , \quad \underset{\sim}{x_2} = \begin{pmatrix} 0.40824 \\ 0.40824 \\ -0.40824 \end{pmatrix}$$

$$\lambda_3 = 0.27924 \quad , \quad \underset{\sim}{x_3} = \begin{pmatrix} 0.27218 \\ 0.46837 \\ 0.40297 \end{pmatrix}$$

(انظر التمرين 11 بآخر هذا الفصل)

الحالة الثانية: B متناسقة (B is symmetric)

نحسب B كما يلي:

$$B = VDV^T \qquad \ldots\ldots (24.7)$$

حيث D قطرية و V هو مصفوفة المتجهات الذاتية التي تجعل من B قطرية.

نقوم بالتعويض بالمعادلة (24.7) في المعادلة (18.7) لنحصل على:

$$A\underset{\sim}{x} = \lambda VDV^T \underset{\sim}{x} \qquad \ldots\ldots (25.7)$$

بضرب (25.7) في V^T نرى أن:

$$V^T AVV^T \underset{\sim}{x} = \lambda \; DV^T \underset{\sim}{x}$$

ولو أخذنا $H = V^T AV$ و $y = V^T \underset{\sim}{x}$ فإننا نحصل على:

$$H \underset{\sim}{y} = \lambda \; D \underset{\sim}{y} \qquad \ldots\ldots (26.7)$$

والمعادلة (26.7) هي إحدى مثيلات المعادلة (18.7) للحالة الأولى.

وهكذا نعيد نفس الخطوات لتلك الحالة، وهي الحالة التي تكون فيها B قطرية، حيث نضع $D = GG$ ثم $Q = G^{-1}HG^{-1}$ و $\underset{\sim}{z} = G \underset{\sim}{y}$ وبذلك نجد أن:

$$\underset{\sim}{x} = VG^{-1} \underset{\sim}{z}$$

ملاحظة: في الحالتين الأولى والثانية افترضنا أن B مصفوفة حقيقية و متناسقة وموجبة تحديدا (positive definite). هذا بالطبع يمكننا من افتراض صحة المعادلة (19.7) ومثيلتها في الحالة الثانية.

تمارين(7)

1. هل يمكننا استخدام طريقة جاكوبي دائماً لحل مسائل القيم الذاتية ؟ اشرح.

2. ماذا تعني العبارة "أن B يجب أن تكون موجبة تحديدا" ؟ اضرب بعض الأمثلة على ذلك.

3. لماذا نهتم بدراسة مسائل القيم الذاتية التي تكون فيها المصفوفة A متناسقة ؟

4. أوجد القيم الذاتية والمتجهات الذاتية لنظام المعادلات

$$\begin{aligned} x_1 + x_2 + x_3 &= \lambda\, x_1 \\ x_1 + 2x_2 + 2x_3 &= \lambda\, x_2 \\ x_1 + 2x_2 + 3x_3 &= \lambda\, x_3 \end{aligned}$$

5. أوضح أنه يمكن استخدام دورانين فقط لجعل المصفوفة A أسفله قطرية.اكتب القيم الذاتية في صورة كسرية.

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

6. أوجد عزوم القصور الذاتية الأساسية لجسم جاسئ مصفوفته المعطاة أسفله؛ هل يمكنك أيضاً تعيين المحاور الأساسية؟

$$I = mr^2 \begin{pmatrix} 10 & 0.134 & -0.866 \\ 0.134 & 6.5 & -1.0 \\ -0.866 & -1.0 & 7.5 \end{pmatrix}$$

7. في مسألة إجهاد ثلاثية الأبعاد نحصل على المصفوفة:

$$\begin{pmatrix} \sigma_x & \delta_{xy} & \delta_{xz} \\ \delta_{yx} & \sigma_y & \delta_{yz} \\ \delta_{zx} & \delta_{yz} & \sigma_z \end{pmatrix}$$

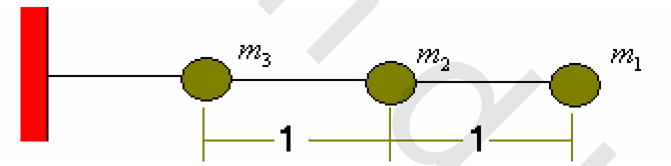
أوجد القيمة الأساسية للإجهاد إذا علمت أن:

$$\delta_{zx} = 180, \delta_{xy} = 65, \delta_{yz} = 75, \sigma_z = 150, \sigma_y = 200, \sigma_x = 120$$

8. عين تردد النظام الموضح بالشكل أسفله علماً بأن معادلة الحركة هي :

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \frac{\omega^2 c}{27} \begin{pmatrix} 27 & 14 & 4 \\ 14 & 8 & 2.5 \\ 4 & 2.5 & 1 \end{pmatrix} \begin{pmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

حيث $m_3 = 3, m_1 = m_2 = 2$ و c ثابت.



9.

أ- أكتب مصفوفة التحويل للمسألة $A\underset{\sim}{x} = \lambda \underset{\sim}{x}$ حيث: $A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$.

ب- ما هي القيم الذاتية للمسألة بالفقرة (أ).

ج- أكتب المتجهات الذاتية للمسألة بالفقرة (أ).

د- أوجد القيم الذاتية للمسألة $C\underset{\sim}{x} = \lambda A\underset{\sim}{x}$ حيث A كما جاء بالفقرة (أ). و $C = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}$.

10. هل يمكنك اقتراح طريقة لإيجاد القيم الذاتية لأي مصفوفة $A=(a_{ij})$ في الحالة العامة التي تكون فيها A غير متناسقة وذلك من خلال دراستك لهذا الفصل ؟ وضح !

11. قم بالتأكد من صحة نتائج المثال (5.7) وذلك بإجراءات الحسابات بإحدى لغات البرمجة.