الفصل الثالث

الطرق التقريبية لحل نظام من المعادلات الخطية

NUMERICAL METHODS FOR SOLVING SYSTEM OF LINEAR EQUATIONS

(٣,١) مقدمة

Introduction

تعتبر الطرق التكرارية وسيلة ضرورية لحل نظام من المعادلات غير الخطية، ولكن إذا كانت هذه المعادلات خطية فلدينا الاختيار بين استعمال الطرق التكرارية أو الطرق المباشرة المستخدمة في مقرر الجبر الخطي. ومع تطور طرق الحاسب الآلي نجد في بعض الأحيان أن استخدام الطرق التكرارية قد يكون أقل تكلفة من استخدام الطرق المباشرة لحل نظام من المعادلات الخطية وسوف نقوم بعرضها الآن بصورة بسيطة.

(٣,٢) الطرق المباشرة لحل نظام من المعادلات الخطية

إن نظام المعادلات الخطية من الممكن أن يأخذ الصورة التالية :

$$A\underline{x} = \underline{b}$$

حيث إن A هي مصفوفة المعاملات. ويقال أن المصفوفة ذات ظروف معتلة (مرضية) ill condition matrix عندما تكون $|A| << 1$ أو $|A| >> 1$ وتظهر خطورة تلك النوعية

في أنه عند استخدام طريقة عددية لحل نظام المعادلات الخطية يمكن أن يحدث تغير بسيط في عناصرها نتيجة إلى خطأ بسيط في القياس مثلاً أو نتيجة إلى خطأ وقف التدوير مما يؤدي إلى فرق كبير في الحل، ونوضح ذلك في المثال التالي.

مثال (٣,١)

عند حل النظام الخطي

$$0.9x + y = 1,$$
$$x + 1.1y = 2,$$

نجد أن الحل $y = -80$,$x = 90$ ولكن حل النظام الخطي:

$$x + 1.1y = 1,$$
$$0.9x + y = 2,$$

هو $x = -120, \quad y = 110$

نلاحظ أنه مع اختلاف بسيط في المعاملات ولكن الحل قد اختلف تماماً عن الأول؛ ولذلك يجب التأكد من أن المصفوفة ليست ذات ظروف معتلة (مرضية). وقد تم دراسة طرق الحذف وطريقة كرامر وطريقة المصفوفة العكسية سابقاً في الجبر الخطي، وسوف نعرض الطرق المباشرة أولاً.

(٣,٢,١) طريقة كرامر Cramer's method

بالتأكيد أنه تم التعرض لطريقة كرامر بالتفصيل في مقرر الجبر الخطي، وهنا نأخذ ثلاث معادلات في ثلاثة مجاهيل لتوضيح كيفية استخدام طريقة كرامر في حل النظام الخطي:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1,$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2,$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3,$$

نجد أن :

$$\Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}, \qquad \Delta_1 = \begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix},$$

$$\Delta_2 = \begin{vmatrix} a_{11} & b_1 & a_{13} \\ a_{21} & b_2 & a_{23} \\ a_{31} & b_3 & a_{33} \end{vmatrix}, \qquad \Delta_3 = \begin{vmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ a_{31} & a_{32} & b_3 \end{vmatrix}$$

إذن يكون الحل للنظام على الصورة :

$$x_1 = \frac{\Delta_1}{\Delta}, \qquad x_2 = \frac{\Delta_2}{\Delta}, \qquad x_3 = \frac{\Delta_3}{\Delta}$$

ومن عيوب هذه الطريقة التي لابد أن تؤخذ في الاعتبار أن عدد العمليات كبير جداً وتظهر في حالة وجود عدد كبير من المعادلات، ونجد أنه من الصعب تطبيق طريقة كرامر بسبب عدد العمليات المستخدمة في الحل.

مثال (٣.٢)

حل نظام المعادلات التالية باستخدام طريقة كرامر.

$$2x_1 - 3x_2 - x_3 = -5,$$
$$x_1 - x_2 + x_3 = -4,$$
$$3x_1 + 2x_2 + x_3 = 7,$$

الحل

كما سبق نجد أن :

$$\Delta = 17, \quad \Delta_1 = 17, \quad \Delta_2 = 51, \quad \Delta_3 = -34,$$

إذن يكون الحل هو :

$$x_1 = 1, \qquad x_2 = 3, \qquad x_3 = -2 \,.$$

تمارين (٣,١)

١- استخدم قاعدة كرامر لحل كل من الأنظمة التالية :

أ)
$$\begin{aligned} x - 3y - z &= -7 \\ x - y - z &= -2 \\ x - 6y - 2z &= -3 \end{aligned}$$

ب)
$$\begin{aligned} x - y + z - w &= 6 \\ 2x + y + w &= -1 \\ x + 2y + z - 2w &= 5 \\ 3x - y - 2z + w &= -2 \end{aligned}$$

(٣,٢,٢) طريقة جاوس الحذفية **Gauss-elimination method**

وتتلخص طريقة جاوس الحذفية في إيجاد المصفوفة الموسعة ثم باستخدام العمليات الصفية الأولية نحصل على مصفوفة مثلثية ونحصل على نظام يسهل التعامل معه في الحل بالتدريج حسب المصفوفة الناتجة.

مثال رقم (٣,٣)

حل المعادلات التالية بطريقة جاوس الحذفية :

$$2x_1 + x_2 - x_3 = 5,$$

$$x_1 + 2x_2 + 4x_3 = 10,$$

$$5x_1 + 4x_2 - x_3 = 14,$$

الحل

نقوم بتكوين المصفوفة الموسعة ومن ثم نستخدم العلميات الأولية :

$$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 5 \\ 1 & 2 & 4 & 10 \\ 5 & 4 & -1 & 14 \end{array}\right] \Rightarrow \left[\begin{array}{ccc|c} 2 & 1 & -1 & 5 \\ 0 & 1.5 & 4.5 & 7.5 \\ 0 & 0 & -3 & -6 \end{array}\right]$$

ومن الصف الثالث نحصل على :

$$-3x_3 = -6, \therefore x_3 = 2$$

ومن الصف الثاني نحصل على :

$$1.5x_2 + 4.5x_3 = 7.5 \therefore x_2 = -1$$

ومن الصف الأول نحصل على :

$$2x_1 + x_2 - x_3 = 5 \therefore x_1 = 4$$

وهو الحل للنظام السابق.

البرنامج رقم (٣.١)

```
C*************************************************************
C  GAUSS ELIMINATION WITH BACKWARD SUBSTITUTION
C*************************************************************
C     TO SOLVE THE N BY N LINEAR SYSTEM
C   E1:  A(1,1) X(1) + A(1,2) X(2) + ... + A(1,N) X(N) = A(1,N+1)
C   E2:  A(2,1) X(1) + A(2,2) X(2) + ... + A(2,N) X(N) = A(2,N+1)
C   EN:  A(N,1) X(1) + A(N,2) X(2) + ... + A(N,N) X(N) = A(N,N+1)
C   INPUT: NUMBER OF UNKNOWNS AND EQUATIONS N; AUGMENTED
C              MATRIX A = (A(I,J)) WHERE 1<=I<=N AND 1<=J<=N+1.
```

```
C  OUTPUT: SOLUTION X(1),X(2),..,X(N) OR A MESSAGE THAT THE
LINEAR
C          SYSTEM HAS NO UNIQUE SOLUTION.
C    INITIALZATION
     DIMENSION A(10,11), X(10)
     CHARACTER NAME*30,NAME1*30,AA*1
     INTEGER INP,OUP,FLAG
     LOGICAL OK
     OPEN(UNIT=5,FILE='CON',ACCESS='SEQUENTIAL')
     OPEN(UNIT=6,FILE='CON',ACCESS='SEQUENTIAL')
     WRITE(6,*) 'This is Gaussian Elimination.'
     WRITE(6,*) 'The array will be input from a text file in the'
     WRITE(6,*) ' order: A(1,1), A(1,2), ..., A(1,N+1), A(2,1),'
     WRITE(6,*) ' A(2,2), ..., A(2,N+1)..., A(N,1), A(N,2),'
     WRITE(6,*) ' ..., A(N,N+1) '
     WRITE(6,*) 'Place as many entries as desired on each line,'
     WRITE(6,*) ' but separate entries with at least one blank.'
     OK = .FALSE.
     WRITE(6,*) 'Has the input file been created?'
     WRITE(6,*) 'Enter Y or N - letter within quotes '
     WRITE(6,*) ' '
     READ(5,*)  AA
     IF (( AA .EQ. 'Y' ) .OR.( AA .EQ. 'y' )) THEN
    WRITE(6,*) 'Input the file name in the form - '
    WRITE(6,*) 'drive:name.ext  contained in quotes'
    WRITE(6,*) 'as example:    "A:DATA.DTA" '
    WRITE(6,*) ' '
     READ(5,*)  NAME
     INP = 4
     OPEN(UNIT=INP,FILE=NAME,ACCESS='SEQUENTIAL')
     OK = .FALSE.
9    IF (OK) GOTO 11
     WRITE(6,*) 'Input the number of equations - an integer '
     WRITE(6,*) ' '
     READ(5,*) N
     IF (N .GT. 0) THEN
     M = N+1
    READ(INP,*) ((A(I,J), J=1,M),I=1,N)
    OK = .TRUE.
    CLOSE(UNIT=INP)
    ELSE
    WRITE(6,*) 'The number must be a positive integer'
    ENDIF
    GOTO 9
```

```
   ELSE
   WRITE(6,*) 'The program will end so the input file can '
   WRITE(6,*) 'be created. '
   OK = .FALSE.
   ENDIF
11 IF( .NOT. OK) GOTO 400
   WRITE(6,*) 'Select output destination: '
   WRITE(6,*) '1. Screen '
   WRITE(6,*) '2. Text file '
   WRITE(6,*) 'Enter 1 or 2 '
   WRITE(6,*) ' '
   READ(5,*) FLAG
   IF ( FLAG .EQ. 2 ) THEN
   WRITE(6,*) 'Input the file name in the form - '
   WRITE(6,*) 'drive:name.ext'
   WRITE(6,*) 'with the name contained within quotes'
   WRITE(6,*) 'as example:   "A:OUTPUT.DTA" '
   WRITE(6,*) ' '
    READ(5,*) NAME1
    OUP = 3
    OPEN(UNIT=OUP,FILE=NAME1,STATUS='NEW')
   ELSE
    OUP = 6
   ENDIF
   WRITE(OUP,*) 'GAUSSIAN ELIMINATION'
C  ICHG COUNTS NUMBER OF INTERCHANGES
   ICHG = 0
  WRITE(OUP,3)
  WRITE(OUP,4) ((A(I,J),J=1,M),I=1,N)
C STEP 1
C ELIMINATION PROCESS
   NN = N-1
   DO 10 I=1,NN
C  STEP 2
C  USE IP IN PLACE OF P
    IP = I
100      IF (ABS(A(IP,I)).GE.1.0E-20 .OR. IP.GT.N) GOTO 200
     IP = IP+1
     GOTO 100
200      IF(IP.EQ.N+1)THEN
C            SYSTEM DOES NOT HAVE UNIQUE SOLUTION
      WRITE(OUP,5)
      GOTO 400
     END IF
```

```
C      STEP 3
       IF(IP.NE.I) THEN
       DO 20 JJ=1,M
       C = A(I,JJ)
       A(I,JJ) = A(IP,JJ)
20     A(IP,JJ) = C
       ICHG = ICHG+1
       END IF
C      STEP 4
       JJ = I+1
       DO 30 J=JJ,N
C      STEP 5
C     USE XM IN PLACE OF M(J,I)
      XM = A(J,I)/A(I,I)
C      STEP 6
       DO 40 K=JJ,M
40     A(J,K) = A(J,K)-XM*A(I,K)
C       MULTIPLIER XM COULD BE SAVED IN A(J,I)
30      A(J,I) = 0
10    CONTINUE
C     STEP 7
    IF(ABS(A(N,N)).LT.1.0E-20) THEN
C     SYSTEM DOES NOT HAVE UNIQUE SOLUTION
      WRITE(OUP,5)
      GOTO 400
      END IF
C      STEP 8
C     START BACKWARD SUBSTITUTION
     X(N) = A(N,N+1)/A(N,N)
C     STEP 9
      L = N-1
      DO 15 K=1,L
      I = L-K+1
      JJ = I+1
      SUM = 0.0
      DO 16 KK=JJ,N
16    SUM = SUM-A(I,KK)*X(KK)
15    X(I) = (A(I,N+1)+SUM)/A(I,I)
      WRITE(OUP,6)((A(I,J),J=1,M),I=1,N)
C    STEP 10
C    PROCEDURE COMPLETED SUCCESSFULLY
     WRITE(OUP,7)(X(I),I=1,N)
     WRITE(OUP,8) ICHG
400  CLOSE(UNIT=5)
```

```
      CLOSE(UNIT=OUP)
      IF(OUP.NE.6) CLOSE(UNIT=6)
      STOP
 5    FORMAT(1X,'THE PRECEDING SYSTEM HAS NO UNIQUE SOLUTION')
 4    FORMAT(5(1X,E15.8))
 6    FORMAT(1X,'THE REDUCED SYSTEM:',/,(5(1X,E14.8)))
 7    FORMAT(1X,'HAS SOLUTION VECTOR',/,4(1X,E14.8))
 8    FORMAT(1X,'NUMBER OF INTERCHANGES = ',3X,I2)
 3    FORMAT(1X,'ORIGINAL SYSTEM',/)
      END
```

وهنا نطرح سؤالاً ، ماذا يحدث إذا كان عنصر من عناصر القطر الرئيس يساوي صفراً $(a_{ii} = 0)$ ؟ وللإجابة على هذا السؤال فإننا نقوم بتبديل الصف رقم i بصف آخر رقم m حيث إن $a_{mm} \neq 0$ ، ولتقليل مقدار الخطأ يجب أن ترتب المعادلات حيث إن a_{ii} يكون أكبر عنصر في العمود رقم i وتسمى هذه العملية بعملية الارتكاز pivoting.

(٣,٢,٣) طريقة التحليل لمصفوفة مثلثية سفلى وعليا

LU decomposition (Cholesky or Crout)

إن طريقة التحليل هي عبارة عن تحسين لطريقة الحذف لجاوس وتسمى طريقة اختزال كراوت Crout وتسمى أيضاً طريق تشولسكي Cholesky وتستخدم في البرمجة باستعمال الحاسب الآلي لتصميم برنامج لحل نظام معادلات جبرية ، وفي هذه الطريقة يتم تحويل مصفوفة المعاملات A إلى حاصل ضرب مصفوفتين L,U ، حيث إن L هي مصفوفة مثلثية سفلى وU مصفوفة مثلثية عليا. وبالتأكيد تم دراسة الطريقة بالتفصيل في مقرر الجبر الخطي ونعرض الطريقة في صورة بسيطة. ونوضح أنه إذا كانت A من النظام 4×4 فإننا سوف نحصل على التالي :

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} \ell_{11} & 0 & 0 & 0 \\ \ell_{21} & \ell_{22} & 0 & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} & 0 \\ \ell_{41} & \ell_{42} & \ell_{43} & \ell_{44} \end{bmatrix} = \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ومن خواص المصفوفات أنه يمكن إيجاد باقي عناصر المصفوفتين u_{ij} , ℓ_{ij} .

مثال (٣،٤)

إذا كانت $A = \begin{bmatrix} 3 & -1 & 2 \\ 1 & 2 & 3 \\ 2 & -2 & -1 \end{bmatrix}$ أوجد المصفوفتين L ، U بحيث إن $A = LU$.

الحل

$$\begin{bmatrix} 3 & -1 & 2 \\ 1 & 2 & 3 \\ 2 & -2 & -1 \end{bmatrix} = \begin{bmatrix} \ell_{11} & 0 & 0 \\ \ell_{21} & \ell_{22} & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

وبالحل نحصل على التالي :

$$L = \begin{pmatrix} 3 & 0 & 0 \\ 1 & \frac{7}{3} & 0 \\ 2 & -\frac{4}{3} & -1 \end{pmatrix}, \qquad U = \begin{pmatrix} 1 & -\frac{1}{3} & \frac{2}{3} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

ولحل النظام $A\underline{x} = \underline{b}$ باستخدام طريقة التحليل نتبع الخطوات التالية :

١- $LU\underline{X} = \underline{b}$

٢- $U\underline{x} = \underline{Y}$

٣- $L\underline{Y} = \underline{b}$ ومنها نحصل على $\underline{Y}$

٤- $U\underline{x} = \underline{Y}$ ومنها نحصل على $\underline{X}$.

مثال (٣،٥)

حل نظام المعادلات التالية باستخدام طريقة التحليل:

$$\begin{aligned}
4x_1 - x_2 &= 100,\\
-x_1 + 4x_2 - x_3 &= 200,\\
-x_2 + 4x_3 - x_4 &= 200,\\
-x_3 + 4x_4 - x_5 &= 200,\\
-x_4 + 4x_5 &= 200,
\end{aligned}$$

الحل

نحصل الحل كالتالي:

$$x_1 = 46.1538, \qquad x_2 = 84.6153$$
$$x_3 = 92.3075, \qquad x_4 = 84.6153$$
$$x_5 = 46.1538$$

تمارين (٣،٢)

١- حل نظام المعادلات التالي بطريقة التحليل إلى LU.

$$\begin{aligned}
2x_1 - x_2 + x_3 &= 3,\\
5x_1 - 8x_2 + 3x_3 &= 2,\\
x_1 + 3x_2 - x_3 &= 5
\end{aligned}$$

٢- باستخدام طريقة التحليل LU حل النظام التالي:

$$\begin{aligned}
x + 2y + 3z + 4w &= -3,\\
2x + 13y + 6z + 14w &= 0,\\
3x + 6y + 25z + 8w &= -61,
\end{aligned}$$

$$4x + 14y + 8z + 57w = 3$$

٣- حل نظام المعادلات التالي بطريقة جاوس للحذف.

$$\begin{aligned} 4x_1 + 3x_2 + 5x_3 &= 3, \\ 2x_1 + x_2 + 3x_3 &= 1, \\ x_1 + 5x_2 + 6x_3 &= 4 \end{aligned}$$

(٣,٢,٤) طريقة جاوس- جوردان Gauss-Jordan method

إن طريقة جاوس- جوردان هي نفس طريقة جاوس الحذفية ولكن نستمر في استخدام العمليات الصفية الأولية حتى نحول المصفوفة إلى مصفوفة الوحدة وبذلك نحصل على الحلول مباشرة بدون أي من التعويضات السابقة.

مثال (٣,٦)

حل نظام المعادلات التالية باستخدام طريقة جاوس- جوردان.

$$\begin{aligned} 5x_1 - x_2 &= 9, \\ -x_1 + 5x_2 - x_3 &= 4, \\ x_1 - 6x_2 + 6x_3 &= -10 \end{aligned}$$

الحل

نجد المصفوفة الموسعة ونستخدم العمليات الصفية الأولية.

$$\left[\begin{array}{ccc|c} 5 & -1 & 0 & 9 \\ -1 & 5 & -1 & 4 \\ 1 & -6 & 6 & -10 \end{array}\right] \therefore \left[\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right]$$

ونجد مباشرة أن الحل هو:

$$x_1 = 2, \qquad x_2 = 1, \qquad x_3 = -1$$

البرنامج رقم (٣،٢)

```
C***********************************************************
C   GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
C***********************************************************
C    TO SOLVE THE N BY N LINEAR SYSTEM
C   E1:  A(1,1) X(1) + A(1,2) X(2) + ... + A(1,N) X(N) = A(1,N+1)
C   E2:  A(2,1) X(1) + A(2,2) X(2) + ... + A(2,N) X(N) = A(2,N+1)
C   EN:  A(N,1) X(1) + A(N,2) X(2) + ... + A(N,N) X(N) = A(N,N+1)
C    INPUT:   NUMBER OF UNKNOWNS AND EQUATIONS N;
AUGMENTED MATRIX
C            A=(A(I,J)) WHERE 1<=I<=N AND 1<=J<=N+1.
C    OUTPUT:  SOLUTION X(1),X(2),...,X(N) OR A MESSAGE THAT THE
C            LINEAR SYSTEM HAS NO UNIQUE SOLUTION.
C    INITIALIZATION
     DIMENSION A(10,11),NROW(10),X(10)
     CHARACTER NAME*30,NAME1*30,AA*1
     INTEGER INP,OUP,FLAG
     LOGICAL OK
     OPEN(UNIT=5,FILE='CON',ACCESS='SEQUENTIAL')
     OPEN(UNIT=6,FILE='CON',ACCESS='SEQUENTIAL')
     WRITE(6,*) 'This is Gaussian Elimination with '
     WRITE(6,*) 'Partial Pivoting'
     WRITE(6,*) 'The array will be input from a text file in the'
     WRITE(6,*) ' order: A(1,1), A(1,2), ..., A(1,N+1), A(2,1),'
     WRITE(6,*) ' A(2,2), ..., A(2,N+1)..., A(N,1), A(N,2),'
     WRITE(6,*) ' ..., A(N,N+1) '
     WRITE(6,*) 'Place as many entries as desired on each line,'
     WRITE(6,*) ' but separate entries with at least one blank.'
     OK = .FALSE.
     WRITE(6,*) 'Has the input file been created?'
     WRITE(6,*) 'Enter Y or N - letter within quotes '
     WRITE(6,*) ' '
     READ(5,*)  AA
     IF (( AA .EQ. 'Y' ) .OR.( AA .EQ. 'y' )) THEN
     WRITE(6,*) 'Input the file name in the form - '
     WRITE(6,*) 'drive:name.ext  contained in quotes'
     WRITE(6,*) 'as example:   "A:DATA.DTA" '
     WRITE(6,*) ' '
     READ(5,*)  NAME
     INP = 4
     OPEN(UNIT=INP,FILE=NAME,ACCESS='SEQUENTIAL')
     OK = .FALSE.
19   IF (OK) GOTO 11
```

```
      WRITE(6,*) 'Input the number of equations - an integer '
      WRITE(6,*) ' '
      READ(5,*) N
      IF (N .GT. 0) THEN
      M = N+1
      READ(INP,*) ((A(I,J), J=1,M),I=1,N)
      OK = .TRUE.
      CLOSE(UNIT=INP)
      ELSE
      WRITE(6,*) 'The number must be a positive integer'
      ENDIF
      GOTO 19
      ELSE
      WRITE(6,*) 'The program will end so the input file can '
      WRITE(6,*) 'be created. '
      OK = .FALSE.
      ENDIF
   11 IF ( .NOT. OK) GOTO 400
      WRITE(6,*) 'Select output destination: '
      WRITE(6,*) '1. Screen '
      WRITE(6,*) '2. Text file '
      WRITE(6,*) 'Enter 1 or 2 '
      WRITE(6,*) ' '
      READ(5,*) FLAG
      IF ( FLAG .EQ. 2 ) THEN
      WRITE(6,*) 'Input the file name in the form - '
      WRITE(6,*) 'drive:name.ext'
      WRITE(6,*) 'with the name contained within quotes'
      WRITE(6,*) 'as example:   "A:OUTPUT.DTA" '
      WRITE(6,*) ' '
      READ(5,*) NAME1
      OUP = 3
      OPEN(UNIT=OUP,FILE=NAME1,STATUS='NEW')
      ELSE
      OUP = 6
      ENDIF
      WRITE(OUP,*) 'GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING'
C     ICHG COUNTS NUMBER OF INTERCHANGES
      ICHG=0
      WRITE(OUP,3)
      WRITE(OUP,4) ((A(I,J),J=1,M),I=1,N)
C     STEP 1
      DO 10 I=1,N
C     INITIALIZE ROW POINTER
```

```
10  NROW(I)=I
C   STEP 2
    NN = N-1
    DO 20 I=1,NN
C   STEP 3
    IMAX=NROW(I)
    AMAX=ABS(A(IMAX,I))
    IMAX=I
    JJ=I+1
    DO 30 IP=JJ,N
    JP=NROW(IP)
    IF(ABS(A(JP,I)).GT.AMAX) THEN
    AMAX=ABS(A(JP,I))
    IMAX=IP
    END IF
30  CONTINUE
C   STEP 4
     IF(AMAX.LT.1.0E-20) THEN
     WRITE(OUP,5)
     GOTO 400
     END IF
C    STEP 5
     IF(NROW(I).NE.NROW(IMAX)) THEN
C    SIMULATE ROW INTERCHANGE
     ICHG = ICHG+1
     NCOPY=NROW(I)
     NROW(I)=NROW(IMAX)
     NROW(IMAX)=NCOPY
     END IF
C    STEP 6
     I1=NROW(I)
     DO 40 J=JJ,N
     J1=NROW(J)
C    STEP 7
     XM=A(J1,I)/A(I1,I)
C    STEP 8
     DO 50 K=JJ,M
50   A(J1,K)=A(J1,K)-XM*A(I1,K)
C    MULTIPLIER XM COULD BE SAVED IN A(J1,I)
40   A(J1,I)=0.0
20  CONTINUE
C   STEP 9
    N1=NROW(N)
    IF(ABS(A(N1,N)).LT.1.0E-20) THEN
```

```
C      SYSTEM HAS NO UNIQUE SOLUTION
       WRITE(OUP,5)
       GOTO 400
       END IF
C      STEP 10
C      START BACKWARD SUBSTITUTION
       X(N)=A(N1,N+1)/A(N1,N)
C      STEP 11
       DO 60 K=1,NN
       I=NN-K+1
       JJ=I+1
       N2=NROW(I)
       SUM=0.0
       DO 70 KK=JJ,N
70     SUM=SUM-A(N2,KK)*X(KK)
60     X(I)=(A(N2,N+1)+SUM)/A(N2,I)
C      STEP 12
C      PROCEDURE COMPLETED SUCCESSFULLY
       WRITE(OUP,6)
       WRITE(OUP,13) ((A(I,J),J=1,N),I=1,N)
       WRITE(OUP,7) (X(I),I=1,N)
       WRITE(OUP,8) ICHG
       WRITE(OUP,9) (NROW(I),I=1,N)
400    CLOSE(UNIT=5)
       CLOSE(UNIT=OUP)
       IF(OUP.NE.6) CLOSE(UNIT=6)
       STOP
5      FORMAT(1X,'THE PRECEDING SYSTEM HAS NO UNIQUE SOLUTION')
4      FORMAT((5(1X,E14.8)))
6      FORMAT(1X,'THE REDUCED SYSTEM:')
7      FORMAT(1X,'HAS SOLUTION VECTOR',/,4(1X,E14.8))
8      FORMAT(1X,'NUMBER OF INTERCHANGES = ',3X,I2)
3      FORMAT(1X,1X,'ORIGINAL SYSTEM:',/)
9      FORMAT(1X,4I5)
13     FORMAT((4(1X,E14.8)))
       END
```

تمارين (٣,٣)

حل مسائل التمارين رقم (٣,١) و (٢ , ٣) ولكن بطريقة جاوس- جوردان للحذف.

(٣,٣) الطرق التكرارية لحل نظام من المعادلات الخطية

Iterative Methods for Solving System of Linear Equations

عندما يكون بالنظام الخطي عدد كبير من المجاهيل فإن الطرق المباشرة تكون مكلفة جداً ؛ ولذلك نلجأ إلى الطرق التكرارية في إيجاد حل أقل تكلفة من الطرق المباشرة، وتظهر أهمية الطرق التكرارية في حالة النظم غير الخطية، وأهم تلك الطرق طريقة الجاكوبي وطريقة جاوس سيدال.

(٣,٣,١) طريقة الجاكوبي Jacobi's method

طريقة الجاكوبي هي استخدام لطريقة النقطة الثابتة لنظام وذلك بتحويل نظام المعادلات $A\underline{x} = \underline{b}$ إلى الشكل :

$$\underline{x} = H\underline{x} + \underline{D}$$

ثم نقوم بتكوين الصورة التكرارية التالية :

$$\underline{x}^{(k+1)} = H\underline{x}^{(k)} + \underline{D}$$

ومن $\underline{x}^{(0)}$ اختيارية يمكن لنا أن نكون متتابعة كما يلي :

$$\underline{x}^{(1)}, \underline{x}^{(2)}, \underline{x}^{(3)}, \cdots$$

وتكون تقاربية إلى الحل الصحيح. أي أن نظام المعادلات $A\underline{x} = \underline{b}$ والذي يكون على الصورة التالية :

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2, \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n,
\end{aligned}$$

يكتب على الصورة التالية :

$$x_i = \frac{1}{a_{ii}}\left[b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j \right], \qquad i = 1(1)n$$

وطريقة جاكوبي تكون العلاقة التكرارية التالية :

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j^{(k)} \right], \qquad i = 1(1)n, \qquad k = 0,1,2,\cdots$$

مثال (٣.٧)

حل نظام المعادلات التالية بطريقة جاكوبي :

$$\begin{aligned}
3x_1 + x_2 \qquad &= 5, \\
x_1 - 4x_2 - x_3 &= -6, \\
2x_2 + 5x_3 &= -1,
\end{aligned}$$

ابدأ بالقيم الابتدائية $\underline{x}^{(0)} = (1,1,1)$ واحسب ثلاث دورات وقارن الحل بالحل الصحيح $x = (1,2,-1)$.

الحل

نضع المعادلات على الصورة التالية :

$$x_1^{(k+1)} = \frac{1}{3}\left[5 - x_2^{(k)}\right],$$

$$x_2^{(k+1)} = -\frac{1}{4}\left[-6 - x_1^{(k)} + x_3^{(k)}\right],$$

$$x_3^{(k+1)} = \frac{1}{5}\left[-1 - 2x_2^{(k)}\right]$$

ومنها نحصل على النتائج الموجودة في الجدول رقم (٣,١) وهي تتقارب إلى الحل الصحيح.

الجدول رقم (٣,١)

k	x_1	x_2	x_3
0	1	1	1
1	1.33333	1.5	-0.6
2	1.6667	1.98333	-0.8
3	1.00556	1.99167	-0.993333
.	.	.	.
.	.	.	.
.	.	.	.
8	1	2	-0.999999

مثال (٣,٨)

حل نظام المعادلات التالية بطريقة جاكوبي :

$$5x_1 + 2x_2 - x_3 = 6,$$

$$x_1 + 6x_2 - 3x_3 = 4,$$

$$2x_1 + x_2 + 4x_3 = 7,$$

ابدأ بالقيم الابتدائية $\underline{x}^{(0)} = (0,0,0)$ واحسب عددة دورات وقارن الحل بالحل الصحيح $x = (1,1,1)$.

الحل

نضع المعادلات على الصورة التالية :

$$x_1^{(k+1)} = \frac{1}{5}\left[6 - 2x_2^{(k)} + x_3^{(k)}\right],$$
$$x_2^{(k+1)} = \frac{1}{6}\left[4 - x_1^{(k)} + 3x_3^{(k)}\right],$$
$$x_3^{(k+1)} = \frac{1}{4}\left[7 - 2x_1^{(k)} - x_2^{(k)}\right]$$

ومنها نحصل على النتائج الموجودة في الجدول رقم (٣,٢) وهي تتقارب إلى الحل الصحيح.

الجدول رقم (٣,٢).

k	x_1	x_2	x_3
0	0	0	0
1	1.2	0.667	1.75
2	1.283	1.342	0.983
3	0.860	0.944	0.773
.	.	.	.
.	.	.	.
.	.	.	.
11	1.001	1.000	1.002

البرنامج رقم (٣,٣)

```
C*************************************************************
C           JACOBI ITERATIVE
C*************************************************************
C    TO SOLVE AX = B GIVEN AN INITIAL APPROXIMATION X(0).
C    INPUT:   THE NUMBER OF EQUATIONS AND UNKNOWNS n; THE
ENTRIES
C          A(I,J), 1<=I, J<=n, OF THE MATRIX A; THE ENTRIES B(I),
C          1<=I<=n, OF THE INHOMOGENEOUS TERM B; THE ENTRIES
C          XO(I), 1<=I<=n, OF X(0); TOLERANCE TOL; MAXIMUM
C          NUMBER OF ITERATIONS N.
C    OUTPUT:  THE APPROXIMATE SOLUTION X(1),...,X(n) OR A MESSAGE
C          THAT THE NUMBER OF ITERATIONS WAS EXCEEDED.
C    INITIALIZATION.
     DIMENSION A(10,11),X1(10),X2(10)
C    USE A(I,n+1) = B(I) FOR 1<=I<=n
C    USE X1 FOR XO
     CHARACTER NAME*30,NAME1*30,AA*1
     INTEGER INP,OUP,FLAG
     LOGICAL OK
     OPEN(UNIT=5,FILE='CON',ACCESS='SEQUENTIAL')
     OPEN(UNIT=6,FILE='CON',ACCESS='SEQUENTIAL')
     WRITE(6,*) 'This is the Jacobi Method for Linear Systems.'
     WRITE(6,*) 'The array will be input from a text file in the'
     WRITE(6,*) ' order: A(1,1), A(1,2), ..., A(1,n+1), A(2,1),'
     WRITE(6,*) ' A(2,2), ..., A(2,n+1)..., A(n,1), A(n,2),'
     WRITE(6,*) ' ..., A(n,n+1) '
     WRITE(6,*) 'Place as many entries as desired on each line,'
     WRITE(6,*) ' but separate entries with at least one blank.'
     WRITE(6,*) 'The initial approximation should follow in the'
     WRITE(6,*) 'same format.'
     OK = .FALSE.
     WRITE(6,*) 'Has the input file been created?'
     WRITE(6,*) 'Enter Y or N - letter within quotes '
     WRITE(6,*) ' '
     READ(5,*)  AA
     IF (( AA .EQ. 'Y' ) .OR.( AA .EQ. 'y' )) THEN
     WRITE(6,*) 'Input the file name in the form - '
     WRITE(6,*) 'drive:name.ext  contained in quotes'
     WRITE(6,*) 'as example:   "A:DATA.DTA" '
     WRITE(6,*) ' '
     READ(5,*)  NAME
     INP = 4
```

```
      OPEN(UNIT=INP,FILE=NAME,ACCESS='SEQUENTIAL')
      OK = .FALSE.
19    IF (OK) GOTO 11
      WRITE(6,*) 'Input the number of equations - an integer '
      WRITE(6,*)
      READ(5,*) N
      IF (N .GT. 0) THEN
      M = N+1
      READ(INP,*) ((A(I,J), J=1,M),I=1,N)
      READ(INP,*) (X1(I),I=1,N)
      OK = .TRUE.
      CLOSE(UNIT=INP)
      ELSE
      WRITE(6,*) 'The number must be a positive integer'
      ENDIF
      GOTO 19
11    OK = .FALSE.
12    IF (OK) GOTO 13
      WRITE(6,*) 'Input the tolerance.'
      WRITE(6,*) ' '
      READ(5,*) TOL
      IF (TOL .GT. 0.0) THEN
      OK = .TRUE.
      ELSE
      WRITE(6,*) 'Tolerance must be positive.'
      ENDIF
      GOTO 12
13    OK = .FALSE.
14    IF (OK) GOTO 15
      WRITE(6,*) 'Input maximum number of iterations.'
      WRITE(6,*)
      READ(5,*) NN
      IF (NN .GT. 0) THEN
      OK = .TRUE.
      ELSE
      WRITE(6,*) 'Number must be a positive integer.'
      ENDIF
      GOTO 14
      ELSE
      WRITE(6,*) 'The program will end so the input file can '
      WRITE(6,*) 'be created. '
      OK = .FALSE.
      ENDIF
15    IF(.NOT. OK) GOTO 400
```

```
      WRITE(6,*) 'Select output destination: '
      WRITE(6,*) '1. Screen '
      WRITE(6,*) '2. Text file '
       WRITE(6,*) 'Enter 1 or 2 '
      WRITE(6,*) ' '
      READ(5,*) FLAG
      IF ( FLAG .EQ. 2 ) THEN
      WRITE(6,*) 'Input the file name in the form - '
      WRITE(6,*) 'drive:name.ext'
      WRITE(6,*) 'with the name contained within quotes'
      WRITE(6,*) 'as example:   "A:OUTPUT.DTA" '
      WRITE(6,*) ' '
      READ(5,*) NAME1
      OUP = 3
      OPEN(UNIT=OUP,FILE=NAME1,STATUS='NEW')
      ELSE
      OUP = 6
      ENDIF
     WRITE(OUP,*) 'JACOBI ITERATIVE METHOD FOR LINEAR SYSTEMS'
     WRITE(OUP,3)
     WRITE(OUP,4)((A(I,J),J=1,M),I=1,N)
     WRITE(OUP,5)
     WRITE(OUP,4)(X1(I),I=1,N)
C     STEP 1
     K = 1
C     STEP 2
100   IF (K.GT.NN) GOTO 200
C      ERR IS USED TO TEST ACCURACY-IT MEASURES THE
C      INFINITY-NORM
      ERR = 0.0
C     STEP 3
      DO 10 I=1,N
      S = 0.0
C     DO-LOOP COMPUTED THE SUMMATION
      DO 20 J=1,N
20    S = S-A(I,J)*X1(J)
      S = (S+A(I,N+1))/A(I,I)
      IF(ABS(S).GT.ERR) ERR=ABS(S)
C    USE X2 FOR X
10   X2(I) = X1(I)+S
     WRITE(OUP,6) K,ERR,(X2(I),I=1,N)
C     STEP 4
     IF(ERR.LE.TOL) THEN
C    PROCESS IS COMPLETE
```

```
      WRITE(OUP,7) K,TOL
      GOTO 400
      END IF
C     STEP 5
      K = K+1
C     STEP 6
       DO 30 I=1,N
30     X1(I) = X2(I)
       GOTO 100
C       STEP 7
C      PROCEDURE COMPLETED UNSUCCESSFULLY
200    CONTINUE
       WRITE(OUP,9)
400    CLOSE(UNIT=5)
       CLOSE(UNIT=OUP)
        IF(OUP.NE.6) CLOSE(UNIT=6)
        STOP
3        FORMAT(1X,'ORIGINAL SYSTEM: '/)
4        FORMAT((1X,4(1X,E15.8)))
5        FORMAT(1X,'INITIAL APPROXIMATION:'/)
6     FORMAT(1X,'ITERATION NUMBER',I3,' GIVES ERROR ',E15.8,/,
     *'FOR APPROX.',4(1X,E15.8))
7     FORMAT(1X,'CONVERGENCE ON INTERATION NUMBER ',I4,
     *' , TOLERANCE= ',E15.8)
9      FORMAT(1X,'MAXIMUM NUMBER OF ITERATIONS EXCEEDED')
      END
```

تمارين (٣,٤)

١ - استخدم طريقة جاكوبي في إيجاد حل نظام المعادلات التالية :

$$10x_1 - x_2 - x_3 = 8,$$
$$-x_1 + 10x_2 - x_3 = 8,$$
$$-x_1 - x_2 + 10x_3 = 8,$$

ابدأ بالقيم الابتدائية $\underline{x}^{(0)} = \underline{0}$.

٢ - أوجد حل نظام المعادلات الخطية التالية باستخدام طريقة جاكوبي :

$$8x_1 + x_2 - x_3 = 8,$$

$$x_1 - 7x_2 + 2x_3 = -4,$$
$$2x_1 + x_2 + 9x_3 = 12,$$

ابدأ بالقيم الابتدائية $\underline{x}^{(0)} = \underline{0}$.

٣- أوجد حل نظام المعادلات الخطية التالية باستخدام طريقة جاكوبي :

$$4x_1 + 0.24x_2 - 0.02x_3 = 8,$$
$$0.04x_1 - 0.08x_2 + 4x_3 = 20,$$
$$0.09x_1 + 3x_2 - 0.15x_3 = 9,$$

ابدأ بالقيم الابتدائية $\underline{x}^{(0)} = \underline{0}$.

(٣,٣,٢) طريقة جاوس- سيدال Gauss-Siedel method

طريقة جاوس-سيدال هي تحسين لطريقة جاكوبي والفرق بين طريقة جاكوبي وطريقة جاوس- سيدال في العملية الحسابية هو أن الدورة $(k+1)$ تستعمل في طريقة جاكوبي x_i في الدورة k، بينما نستعمل في طريقة جاوس-سيدال آخر قيمة تم حسابها. وتأخذ طريقة جاوس- سيدال الصورة التالية :

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_i^{(k)} \right], \quad i = 1(1)n, \quad k = 0,1,2,\cdots$$

مثال رقم (٣,٩)

حل نظام المعادلات الخطية التالية بطريقة جاوس- سيدال

$$3x_1 + x_2 = 5,$$
$$x_1 - 4x_2 - x_3 = -6,$$
$$2x_2 + 5x_3 = -1,$$

الحل

طريقة جاوس- سيدال تحول النظام السابق إلى الصورة التكرارية :

$$x_1^{(k+1)} = \frac{1}{3}\left[5 - x_2^{(k)}\right],$$
$$x_2^{(k+1)} = -\frac{1}{4}\left[-6 - x_1^{(k+1)} + x_3^{k}\right],$$
$$x_3^{(k+1)} = \frac{1}{5}\left[-1 - 2x_2^{(k+1)}\right],$$

وبأخذ $\underline{x}^{(0)} = (1,1,1)$ نحصل على النتائج في الجدول رقم (٣،٣).

الجدول رقم (٣،٣).

k	x_1	X_2	x_3
0	1	1	1
1	1.33333	1.58333	-0.833333
2	1.13889	1.99306	-0.997222
3	1.00231	1.99988	-0.999954
4	1.00004	2	-0.999999
.	.	.	.
.	.	.	.
.	.	.	.
8	1	2	-1

ومن المثالين رقمي (٣،٧) و(٣،٩) نجد أن طريقة جاوس- سيدال أسرع في التقارب من طريقة جاكوبي.

البرنامج رقم (٣,٤)

```
C******************************************************************
C        GAUSS-SEIDEL ITERATIVE
C******************************************************************
C    TO SOLVE AX = B GIVEN AN INITIAL APPROXIMATION X(0):
C    INPUT:  THE NUMBER OF EQUATIONS AND UNKNOWNS n; THE ENTRIES
C          A(I,J), 1<=I, J<=n, OF THE MATRIX A; THE ENTRIES B(I)
C          1<=I<=n, OF THE INHOMOGENEOUS TERM B; THE ENTRIES
C          XO(I), 1<=I<=n, OF X(0); TOLERANCE TOL; MAXIMUM
C          NUMBER OF ITERATIONS N.
C    OUTPUT:  THE APPROXIMATE SOLUTION X(1),...,X(n) OR A MESSAGE
C          THAT THE NUMBER OF ITERATIONS WAS EXCEEDED.
C    INITIALIZATION.
     DIMENSION A(10,11),X1(10)
C    USE NN FOR CAPITAL N
C    B(I) = A(I,n+1) FOR 1<=I<=n
C    USE X1 FOR XO
     CHARACTER NAME*30,NAME1*30,AA*1
     INTEGER INP,OUP,FLAG
     LOGICAL OK
     OPEN(UNIT=5,FILE='CON',ACCESS='SEQUENTIAL')
     OPEN(UNIT=6,FILE='CON',ACCESS='SEQUENTIAL')
     WRITE(6,*) 'This is Gauss-Seidel Method for Linear Systems.'
      WRITE(6,*) 'The array will be input from a text file in the'
      WRITE(6,*) ' order: A(1,1), A(1,2), ..., A(1,n+1), A(2,1),'
      WRITE(6,*) ' A(2,2), ..., A(2,n+1)..., A(n,1), A(n,2),'
      WRITE(6,*) ' ..., A(n,n+1) '
      WRITE(6,*) 'Place as many entries as desired on each line,'
      WRITE(6,*) ' but separate entries with at least one blank.'
      WRITE(6,*) 'The initial approximation should follow in the'
      WRITE(6,*) 'same format.'
      OK = .FALSE.
      WRITE(6,*) 'Has the input file been created?'
      WRITE(6,*) 'Enter Y or N - letter within quotes '
       WRITE(6,*) ' '
       READ(5,*)  AA
       IF (( AA .EQ. 'Y' ) .OR.( AA .EQ. 'y' )) THEN
       WRITE(6,*) 'Input the file name in the form - '
       WRITE(6,*) 'drive:name.ext  contained in quotes'
       WRITE(6,*) 'as example:    "A:DATA.DTA" '
       WRITE(6,*) ' '
       READ(5,*)  NAME
       INP = 4
       OPEN(UNIT=INP,FILE=NAME,ACCESS='SEQUENTIAL')
       OK = .FALSE.
19      IF (OK) GOTO 11
        WRITE(6,*) 'Input the number of equations - an integer '
         WRITE(6,*)
```

```
      READ(5,*) N
      IF (N .GT. 0) THEN
      M = N+1
      READ(INP,*) ((A(I,J), J=1,M),I=1,N)
      READ(INP,*) (X1(I),I=1,N)
      OK = .TRUE.
      CLOSE(UNIT=INP)
      ELSE
      WRITE(6,*) 'The number must be a positive integer'
       ENDIF
       GOTO 19
11     OK = .FALSE.
12     IF (OK) GOTO 13
      WRITE(6,*) 'Input the tolerance.'
      WRITE(6,*) ' '
      READ(5,*) TOL
      IF (TOL .GT. 0.0) THEN
      OK = .TRUE.
      ELSE
      WRITE(6,*) 'Tolerance must be positive.'
      ENDIF
      GOTO 12
13    OK = .FALSE.
14     IF (OK) GOTO 15
       WRITE(6,*) 'Input maximum number of iterations.'
       WRITE(6,*)
       READ(5,*) NN
       IF (NN .GT. 0) THEN
      OK = .TRUE.
      ELSE
      WRITE(6,*) 'Number must be a positive integer.'
      ENDIF
      GOTO 14
      ELSE
      WRITE(6,*) 'The program will end so the input file can '
      WRITE(6,*) 'be created. '
       OK = .FALSE.
      ENDIF
15    IF(.NOT. OK) GOTO 400
      WRITE(6,*) 'Select output destination: '
      WRITE(6,*) '1. Screen '
      WRITE(6,*) '2. Text file '
      WRITE(6,*) 'Enter 1 or 2 '
      WRITE(6,*) ' '
      READ(5,*) FLAG
      IF ( FLAG .EQ. 2 ) THEN
     WRITE(6,*) 'Input the file name in the form - '
     WRITE(6,*) 'drive:name.ext'
     WRITE(6,*) 'with the name contained within quotes'
```

```
      WRITE(6,*) 'as example:   "A:OUTPUT.DTA" '
      WRITE(6,*) ' '
      READ(5,*) NAME1
      OUP = 3
      OPEN(UNIT=OUP,FILE=NAME1,STATUS='NEW')
     ELSE
     OUP = 6
     ENDIF
    WRITE(OUP,*) 'GAUSS-SEIDEL METHOD FOR LINEAR SYSTEMS'
    WRITE(OUP,3)
    WRITE(OUP,4)((A(I,J),J=1,M),I=1,N)
    WRITE(OUP,5)
    WRITE(OUP,4)(X1(I),I=1,N)
C    STEP 1
    K = 1
C    STEP 2
100  IF (K.GT.NN) GOTO 200
C     ERR IS USED TO TEST ACCURACY AND MEASURES THE
C      INFINITY-NORM
      ERR = 0.0
C     STEP 3
      DO 10 I=1,N
      S = 0.0
C     DO-LOOP COMPUTED THE SUMMATION
      DO 20 J=1,N
20     S = S-A(I,J)*X1(J)
       S = (S+A(I,N+1))/A(I,I)
       IF(ABS(S).GT.ERR)  ERR=ABS(S)
10     X1(I) = X1(I)+S
      WRITE(OUP,6) K,ERR,(X1(I),I=1,N)
C      STEP 4
       IF(ERR.LE.TOL) THEN
C       PROCESS IS COMPLETE
       WRITE(OUP,7) K,TOL
       GOTO 400
        END IF
C       STEP 5
       K = K+1
C   STEP 6--IS NOT USED SINCE ONLY ONE VECTOR IS REQUIRED
       GOTO 100
C   STEP 7
C   PROCEDURE COMPLETED UNSUCCESSFULLY
200     CONTINUE
        WRITE(OUP,9)
400     CLOSE(UNIT=5)
        CLOSE(UNIT=OUP)
        IF(OUP.NE.6) CLOSE(UNIT=6)
        STOP
3       FORMAT(1X,'ORIGINAL SYSTEM: '/)
```

```
4       FORMAT((1X,4(1X,E15.8)))
5       FORMAT(1X,'INITIAL APPROXIMATION:'/)
6     FORMAT(1X,'ITERATION NUMBER',I3,' GIVES ERROR ',E15.8,/,
     *'FOR APPROX.',4(1X,E15.8))
7       FORMAT(1X,'CONVERGENCE ON ITERATION NUMBER ',I4,/,
     *'TOLERANCE= ',E15.8)
9   FORMAT(1X,'Maximum Number of Iterations Exceeded.')
    END
```

تمارين (٣,٥)

حل كلاً من الأنظمة التالية باستخدام طريقة جاوس- سيدال :

أ)
$$\begin{aligned} 5x_1 + x_2 - 2x_3 &= 1, \\ x_1 - 4x_2 + x_3 &= -1, \\ 2x_1 - x_2 + 6x_3 &= 8 \end{aligned}$$

ب)
$$\begin{aligned} x_1 - 3x_2 + x_3 &= 4, \\ x_1 + 2x_2 + 5x_3 &= -7, \\ 2x_1 \quad + x_3 &= 3 \end{aligned}$$

ج)
$$\begin{aligned} 9x_1 \quad - 5x_3 &= 10, \\ 20x_2 - 12x_3 &= -2, \\ -5x_1 - 12x_2 + 20x_3 &= 0 \end{aligned}$$

د)
$$\begin{aligned} x_1 + x_2 + x_3 &= 0, \\ x_1 + x_2 + 3x_3 &= 0, \\ 3x_1 + 5x_2 + 7x_3 &= 1 \end{aligned}$$

(٣,٣,٣) فوق الاسترخاء المتعاقبة Successive over-relaxation

والآن نستطيع تعميم طريقة جاوس- سيدال بما تسمى بطريقة فوق الاسترخاء المتعاقبة Successive Over-Relaxation (SOR) ، وفي هذه الطريقة يتم كتابة نظام المعادلات $A\underline{x} = \underline{b}$ على الصورة التالية :

$$x_i^{(k+1)} = (1-w)x_i^{(k)} + \frac{w}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_i^{(k)} \right],$$

$$i = 1,2,\cdots,n, \qquad k = 0,1,2,\cdots, \qquad 0 < w < 2$$

وفي حالة $w = 1$ نحصل على طريقة جاوس- سيدال.

مثال (٣،١٠)

حل نظام المعادلات الخطية التالي باستخدام طريقة فوق الاسترخاء المتعاقبة SOR:

$$\begin{aligned} 4x_1 + 3x_2 \quad\quad &= 24, \\ 3x_1 + 4x_2 - x_3 &= 30, \\ -x_2 + 4x_3 &= -24 \end{aligned}$$

$w = 1.25$ ، $w = 1.1$ ، $\underline{x}^{(0)} = (1,1,1)$ وقارن مع الحل الصحيح $(3,4,-5)$.

الحل

نضع المعادلات على الصورة التالية:

$$x_1^{(k+1)} = (1-w)x_1^{(k)} + \frac{w}{4}\left[24 - 3x_2^{(k)}\right],$$

$$x_2^{(k+1)} = (1-w)x_2^{(k)} + \frac{w}{4}\left[30 - 3x_1^{(k+1)} + x_3^{(k)}\right],$$

$$x_3^{(k+1)} = (1-w)x_3^{(k)} + \frac{w}{4}\left[-24 + x_2^{(k+1)}\right],$$

وتوجد نتائج العمليات التكرارية في الجدول رقم (٣،٤)، مع ملاحظة أنه في حالة وضع $w \notin [0,2]$ فإننا نحصل على متتابعة تباعدية لا تؤول إلى الحل لأنه شرط أن $0 < w < 2$.

الجدول رقم (٣.٤).

k	$w = 1.25$ x_1	x_2	x_3	$w = 1.1$ x_1	x_2	x_3
0	1	1	1	1	1	1
1	6.3125	3.51953	-6.65015	5.675	3.74312	-5.67064
2	2.62231	3.95833	-4.60042	2.94442	3.8874	-4.963981
3	3.1333	4.01026	-5.09669	3.09869	3.93978	-5.02016
4	2.95705	4.00748	-4.97349	3.03982	3.96763	-5.00689
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
8	2.99975	4.00007	-4.99989	3.0035	3.9973	-5.00063

البرنامج رقم (٣.٥)

```
C**************************************************************
C                  SOR ALGORITHM
C**************************************************************
C    TO SOLVE AX = B GIVEN THE PARAMETER W AND AN ITITIAL
APPROXIMATION
C    X(O):
C    INPUT:   THE NUMBER OF EQUATIONS AND UNKNOWNS n; THE
ENTRIES
C            A(I,J), 1<=I, J<=n, OF THE MATRIX A; THE ENTRIES B(I),
C            1<=I<=n, OF THE INHOMOGENEOUS TERM B; THE ENTRIES
XO(I),
C            1<=I<=n, OF X(0); TOLERANCE TOL;
C            MAXIMUM NUMBER OF ITERATIONS N; PARAMETER W
(OMEGA).
C    OUTPUT:  THE APPROXIMATE SOLUTION X(1),...,X(n) OR A
MESSAGE
C            THAT THE NUMBER OF ITERATIONS WAS EXCEEDED.
C    INITIALIZATION
     DIMENSION A(10,11),X1(10)
C    USE NN FOR CAPITAL N
C    USE W FOR OMEGA
C    B(I) = A(I,n+1) FOR 1<=I<=n
C    USE X1 FOR XO
     CHARACTER NAME*30,NAME1*30,AA*1
     INTEGER INP,OUP,FLAG
     LOGICAL OK
     OPEN(UNIT=5,FILE='CON',ACCESS='SEQUENTIAL')
```

```
      OPEN(UNIT=6,FILE='CON',ACCESS='SEQUENTIAL')
      WRITE(6,*) 'This is the SOR Method for Linear Systems.'
      WRITE(6,*) 'The array will be input from a text file in the'
      WRITE(6,*) ' order: A(1,1), A(1,2), ..., A(1,n+1), A(2,1),'
      WRITE(6,*) ' A(2,2), ..., A(2,n+1)..., A(n,1), A(n,2),'
      WRITE(6,*) ' ..., A(n,n+1) '
      WRITE(6,*) 'Place as many entries as desired on each line,'
      WRITE(6,*) ' but separate entries with at least one blank.'
      WRITE(6,*) 'The initial approximation should follow in the'
      WRITE(6,*) 'same format.'
      OK = .FALSE.
      WRITE(6,*) 'Has the input file been created?'
      WRITE(6,*) 'Enter Y or N - letter within quotes '
      WRITE(6,*) ' '
      READ(5,*)  AA
      IF (( AA .EQ. 'Y' ) .OR.( AA .EQ. 'y' )) THEN
        WRITE(6,*) 'Input the file name in the form - '
        WRITE(6,*) 'drive:name.ext  contained in quotes'
        WRITE(6,*) 'as example:   "A:DATA.DTA" '
        WRITE(6,*) ' '
        READ(5,*)  NAME
        INP = 4
        OPEN(UNIT=INP,FILE=NAME,ACCESS='SEQUENTIAL')
        OK = .FALSE.
19      IF (OK) GOTO 111
        WRITE(6,*) 'Input the number of equations - an integer '
        WRITE(6,*)
        READ(5,*) N
        IF (N .GT. 0) THEN
          M = N+1
          READ(INP,*) ((A(I,J), J=1,M),I=1,N)
          READ(INP,*) (X1(I),I=1,N)
         OK = .TRUE.
          CLOSE(UNIT=INP)
        ELSE
          WRITE(6,*) 'The number must be a positive integer'
        ENDIF
        GOTO 19
111       OK = .FALSE.
12      IF (OK) GOTO 13
        WRITE(6,*) 'Input the tolerance.'
        WRITE(6,*) ' '
        READ(5,*) TOL
        IF (TOL .GT. 0.0) THEN
```

```
          OK = .TRUE.
        ELSE
          WRITE(6,*) 'Tolerance must be positive.'
        ENDIF
        GOTO 12
13      OK = .FALSE.
14      IF (OK) GOTO 15
        WRITE(6,*) 'Input maximum number of iterations.'
        WRITE(6,*)
        READ(5,*) NN
        IF (NN .GT. 0) THEN
          OK = .TRUE.
        ELSE
          WRITE(6,*) 'Number must be a positive integer.'
        ENDIF
        GOTO 14
15      WRITE(6,*) 'Input parameter w (omega).'
        WRITE(6,*) ' '
        READ(5,*) W
        GOTO 16
      ELSE
        WRITE(6,*) 'The program will end so the input file can '
        WRITE(6,*) 'be created. '
        OK = .FALSE.
      ENDIF
16    IF(.NOT. OK) GOTO 400
      WRITE(6,*) 'Select output destination: '
      WRITE(6,*) '1. Screen '
      WRITE(6,*) '2. Text file '
      WRITE(6,*) 'Enter 1 or 2 '
      WRITE(6,*) ' '
      READ(5,*) FLAG
      IF ( FLAG .EQ. 2 ) THEN
        WRITE(6,*) 'Input the file name in the form - '
        WRITE(6,*) 'drive:name.ext'
        WRITE(6,*) 'with the name contained within quotes'
        WRITE(6,*) 'as example:   "A:OUTPUT.DTA" '
        WRITE(6,*) ' '
        READ(5,*) NAME1
        OUP = 3
        OPEN(UNIT=OUP,FILE=NAME1,STATUS='NEW')
      ELSE
        OUP = 6
      ENDIF
```

```
    WRITE(OUP,*) 'SOR ITERATIVE METHOD FOR LINEAR SYSTEMS'
    WRITE(OUP,3)
    WRITE(OUP,4) ((A(I,J),J=1,M),I=1,N)
    WRITE(OUP,5)
    WRITE(OUP,4) ( X1(I) ,I=1,N)
    WRITE(OUP,11) W
C   STEP 1
    K=1
C   STEP 2
100 IF (K.GT.NN) GOTO 200
       ERR=0.0
C         ERR WILL BE USED TO TEST ACCURACY, IT MEASURES THE
C         INFINITY-NORM
C         STEP 3
C         THE DO-LOOP COMPUTES THE SUMMATION
       DO 10 I=1,N
          S=0.0
          DO 20 J=1,N
20             S=S-A(I,J)*X1(J)
          S=W*(S+A(I,N+1))/A(I,I)
          IF(ABS(S).GT.ERR) ERR=ABS(S)
C            X IS NOT USED, SINCE ONLY ONE VECTOR IS NEEDED
10           X1(I)=X1(I)+S
       WRITE(OUP,6) K,ERR,(X1(I),I=1,N)
C         STEP 4
       IF(ERR.LE.TOL) THEN
          WRITE(OUP,7) K,TOL
          GOTO 400
       END IF
C         STEP 5
       K=K+1
C         STEP 6 IS NOT NEEDED
    GOTO 100
C   STEP 7
C   PROCEDURE COMPLETED SUCCESSFULLY
200 CONTINUE
    WRITE(OUP,8)
400 CLOSE(UNIT=5)
    CLOSE(UNIT=OUP)
    IF(OUP.NE.6) CLOSE(UNIT=6)
    STOP
3    FORMAT(1X,'ORIGINAL SYSTEM:')
4    FORMAT((1X,4(1X,E15.8)))
5    FORMAT(1X,'INITIAL APPROXIMATION:')
```

```
6    FORMAT(1X,'ITERATION NUMBER ',I3,' GIVES ERROR ',E15.8,/,
    *'FOR APPROX.',3(1X,E15.8))
7    FORMAT(1X,'CONVERGENCE ON INTERATION NUMBER ',I4,/,
    *'TOLERANCE = ',E15.8)
8    FORMAT(1X,'Maximum Number of Iterations Exceeded.')
11   FORMAT(1X,'OMEGA IS ',E15.8)
END
```

ومن الملاحظ أن جميع الطرق التكرارية يمكن كتابتها على الصورة الاتجاهية التالية:

$$\underline{x}^{(n+1)} = \underline{B}\ \underline{x}^{(n)} + \underline{G}$$

وندرس شرط التقارب للطرق التكرارية في النظرية التالية.

نظرية (٣،١)

الشرط الكافي لتقارب الطرق التكرارية

$$\underline{x}^{(n+1)} = \underline{B}\ \underline{x}^{(n)} + \underline{G}$$

هو أن يكون $\|B\| < 1$.

البرهان:

إذا كان $\underline{p}$ هو الحل الصحيح فإن:

$$\underline{p} = B\underline{p} + \underline{G}$$

والعلاقة التكرارية $\underline{x}^{(n+1)} = B\ \underline{x}^{(n)} + \underline{G}$

وبطرح المعادلتين السابقتين نحصل على التالي:

$$\underline{x}^{(n+1)} - \underline{p} = B\,(\underline{x}^{(n)} - \underline{p})$$

إذن :

$$\underline{e}^{(n+1)} = B\ \underline{e}^{(n)}, \qquad n \geq 0$$

وبأخذ معيار الطرفين نجد أن :

$$\left\|\underline{e}^{(n+1)}\right\| \leq \|\mathrm{B}\|\ \left\|\underline{e}^{(n)}\right\|$$
$$\leq \|\mathrm{B}\|^{n+1}\ \left\|\underline{e}^{(0)}\right\|$$

ومنها نجد أن الشرط الكافي للتقارب أي $\underline{e}^{(n+1)} \to 0$ عندما يكون $n \to \infty$ هو $\|B\| < 1$. وهو المطلوب.

تعريف: يقال ان المصفوفة A ذات الأبعاد $n \times n$ متقاربة إذا كانت

$$\lim_{k\to\infty} (A^k)_{ij} = 0, \ \ \forall\ i = 1(1)n,\ \ j = 1(1)n$$

مثال (٣.١١)

بفرض المصفوفة A على الصورة $A = \begin{pmatrix} \frac{1}{2} & 0 \\ \frac{1}{4} & \frac{1}{2} \end{pmatrix}$ ، فان $A^k = \begin{pmatrix} \left(\frac{1}{2}\right)^k & 0 \\ \frac{k}{2^{k+1}} & (\frac{1}{2})^k \end{pmatrix}$

حيث إن $\lim_{k\to\infty} \frac{k}{2^{k+1}} = 0,\ \lim_{k\to\infty}\left(\frac{1}{2}\right)^k = 0$ ولذلك فإن المصفوفة A متقاربة.

نظرية (٣.٢)

إذا كانت A مصفوفة مربعة فإن الأجزاء التالية تكون متكافئة :

١- A مصفوفة تقاربية أي أن $\lim_{n\to\infty} A^n = 0$.

٢- $\lim_{n\to\infty} \|A^n\| = 0$.

٣- $\rho(A) < 1$ حيث إن $\rho(A) = \max|\lambda|$ هي نصف القطر الطيفي للمصفوفة A ، حيث إن λ هى القيم المميزة.

٤- $\lim_{n\to\infty} A^n \underline{x} = \underline{0}$ لكل $\underline{x}$.

وهذه النظرية تستخدم في برهان النظرية التالية :

نظرية (٣،٣)

إن الشرط اللازم والكافي لتقارب طريقة التحسين المتتالي التالية :

$$\underline{x}^{(n+1)} = B\ \underline{x}^{(n)} + \underline{C}$$

هو $|\rho(B)| < 1$ حيث إن $\rho(B)$ هي نصف القطر الطيفي للمصفوفة B .

البرهان:

نفرض أن $|\rho(B)| < 1$

إذن :
$$\underline{x}^{(n+1)} = B\ \underline{x}^{(n)} + \underline{C} = B^2 \underline{x}^{(n-1)} + (B+I)\underline{C} \cdots$$

$$\underline{x}^{(n+1)} = B^{n+1} \underline{x}^{(0)} + (I + B + B^2 + \cdots + B^n)\underline{C}$$

وحيث إن $\rho(B) < 1$ فإن B تكون تقاربية من النظرية السابقة. وبأخذ نهاية الطرفين عندما تكون $n \to \infty$ فإننا نحصل على :

$$\therefore \lim_{n\to\infty} \underline{x}^{(n+1)} = (I - B)^{-1} \underline{C} \qquad (١)$$

وبفرض أن $\underline{x}$ هو الحل الصحيح نحصل على :

$$\underline{x} = B\underline{x} + \underline{C}$$
$$\because (I - B)\underline{x} = \underline{C}$$

$$\therefore \underline{x} = (I - B)^{-1}\underline{C} \qquad (٢)$$

ومن المعادلتين (١) و (٢) نحصل على التالي :

$$\lim_{n\to\infty} \underline{x}^{(n+1)} = \underline{x}$$

وهذا هو المطلوب.

نظرية (٣,٤)

يعطى الحد الأعلى للخطأ في طريقة التحسين المتتالي $\underline{x}^{(n+1)} = B\ \underline{x}^{(n)} + \underline{C}$ بالعلاقة التالية :

$$\left\|\underline{e}^{(n)}\right\| \le \|B\|^n \left(\frac{\|\underline{C}\|}{1-\|B\|} + \left\|\underline{x}^{(0)}\right\|\right)$$

البرهان:

كما في النظرية السابقة نجد أن :

$$\underline{x}^{(n+1)} = B^{n+1}\ \underline{x}^{(0)} + (I + B + B^2 + \cdots + B^n)\underline{C} \qquad (١)$$

$$\underline{x} = (I - B)^{-1}\underline{C} \qquad (٢)$$

بطرح المعادلتين نحصل على المعادلة التالية :

$$\underline{x}^{(n+1)} - \underline{x} = B^{n+1}\underline{x}^{(0)} + [B^{n+1} + B^{n+2} + \cdots)\underline{C}$$

بأخذ معيار الطرفين نجد أن :

$$\left\|\underline{e}^{(n+1)}\right\| \le \left\|B^{n+1}\right\| \left((1 + \|B\| + \|B\|^2 + \cdots)\|\underline{C}\| + \left\|\underline{x}^{(0)}\right\|\right)$$

$$\therefore \left\|\underline{e}^{(n+1)}\right\| \le \left\|B\right\|^{n+1}\left(\frac{\left\|C\right\|}{1-\left\|B\right\|}+\left\|\underline{x}^{(0)}\right\|\right)$$

ومنها نحصل على التالي:

$$\left\|\underline{e}^{(n)}\right\| \le \left\|B\right\|^{n}\left(\frac{\left\|C\right\|}{1-\left\|B\right\|}+\left\|\underline{x}^{(0)}\right\|\right)$$

وهذا هو المطلوب.

تمارين (٣,٦)

١- حل النظام الخطي التالي باستخدام طريقة جاوس- سيدال وطريقة جاكوبي:

$$\begin{aligned}10x_1 - x_2 + 2x_3 &= 6,\\ -x_1 + 11x_2 - x_3 + 3x_4 &= 25,\\ 2x_1 - x_2 + 10x_3 - x_4 &= -11,\\ 3x_2 - x_3 + 8x_4 &= 15,\end{aligned}$$

مع أخذ $\underline{x}^{(0)} = \underline{0}$ وقارن مع الحل الحقيقي $\underline{x} = (1,2,-1,1)$.

٢- حل النظام الخطي باستخدام طريقة فوق الاسترخاء المتعاقبة SOR مع $w = 1.5$ لحل النظام الخطي التالي:

$$\begin{aligned}4x_1 + 3x_2 \qquad &= 24,\\ 3x_1 + 4x_2 - x_3 &= 30,\\ -x_2 + 4x_3 &= -24,\end{aligned}$$

وأخذ $\underline{x}^{(0)} = (1,1,1)$ مع المقارنة بالحل الصحيح $\underline{x} = (3,4,-5)$.

٣- أوجد التقريب الثاني بطريقة جاكوبي لحل النظام الخطي التالي، وخذ $\underline{x}^{(0)} = \underline{0}$:

أ)
$$\begin{aligned}
3x_1 - x_2 + x_3 &= 1,\\
3x_1 + 6x_2 + 2x_3 &= 0,\\
3x_1 + 3x_2 + 7x_3 &= 4,
\end{aligned}$$

ب)
$$\begin{aligned}
10x_1 - x_2 \qquad &= 9,\\
-x_1 + 10x_2 - 2x_3 &= 7,\\
-2x_2 + 10x_3 &= 6,
\end{aligned}$$

٤- حل المسألة رقم (٣) مرة أخرى بطريقة جاوس- سيدال ثم قارن بين كل منهما.

٥- أوجد التقريب الثاني بطريقة فوق الاسترخاء المتعاقبة SOR ، $w = 1.1, 1.2, 1.3$ و $\underline{x}^{(0)} = \underline{0}$ لكل من النظم التالية:

أ)
$$\begin{aligned}
3x_1 - x_2 + x_3 &= 1,\\
3x_1 + 6x_2 + 2x_3 &= 0,\\
3x_1 + 3x_2 + 7x_3 &= 4,
\end{aligned}$$

ب)
$$\begin{aligned}
10x_1 - x_2 \qquad &= 9,\\
-x_1 + 10x_2 - 2x_3 &= 7,\\
-2x_2 + 10x_3 &= 6,
\end{aligned}$$