บทที่ 4
การหาผลเฉลยของระบบเชิงเส้นและของระบบไม่เป็นเชิงเส้น (Linear and Nonlinear Systems) โดยใช้การคำนวณแบบ Fixed Precision

หน้า

บทที่ 4
การหาผลเฉลยของระบบเชิงเส้นและของระบบไม่เป็นเชิงเส้น (Linear and Nonlinear Systems) โดยใช้การคำนวณแบบ Fixed Precision

ในบทนี้จะอธิบายการใช้วิธีการตรง (Direct methods) จากบทที่ 3 อย่างมีประสิทธิภาพเมื่อใช้การคำนวณแบบ fixed precision (ในบทที่ 3 เราใช้ exact arithmetic) วิธีการทำซ้ำสำหรับการหาผลเฉลยของระบบเชิงเส้นที่จะกล่าวในบทนี้คือ วิธีเกาส์-ไซเดล นอกจากนั้นจะแสดงการใช้วิธีของนิวตัน-ราฟสันสำหรับระบบไม่เป็นเชิงเส้น

(นักศึกษาต้องทบทวนการใช้ Triangular Decomposition Algorithm เพื่อหาผลเฉลยระบบเชิงเส้นระบบหนึ่งคือ Ax = b หรือ ระบบเชิงเส้นหลาย ๆ ระบบพร้อม ๆกัน นั่นคือ AX = B)

4.1 <u>การทำให้การปัดเศษ (Roundoff) มีผลน้อยที่สุด</u>

ตัวอย่างต่อไปนี้จะแสดงว่าเมื่อหาผลเฉลยของ Ax = b ใน fixed-precision arithmetic นั้น ความแม่นยำของผลเฉลยจะขึ้นอยู่กับ Pivoting strategy ที่ใช้

<u>ตัวอย่าง</u> จงหาผลเฉลยของระบบเชิงเส้นขนาด (2 × 2)

$(E_1)\quad 0.01x_1 + 100x_2 = 100$

$(E_2)\quad x_1 + x_2 = 2$

$$[A:b] = \begin{bmatrix} 0.01 & 100 & : & 100 \\ 1 & 1 & : & 2 \end{bmatrix} \qquad \ldots(1)$$

ใช้ 3s arithmetic (นั่นคือปัดให้เหลือ 3s หลังจากการคำนวณแต่ละครั้ง)

<u>Solution 1</u> โดยการใช้ Basic Pivoting เราจะได้

$$\hat{L}\backslash\hat{U} = \left[\begin{array}{c|c} 0.01 & 10,000 \\ \hline 1 & -10,000 \end{array}\right] = L\backslash U \qquad \ldots(2a)$$

โปรดสังเกตว่ามีความคลาดเคลื่อนจากการปัดเศษใน pivot ตัวที่สอง l_{22} เพราะว่า

$l_{22} = 1-(1)(10,000) = -9999 \doteq -1.00 \times 10^{4} = -10,000 (3s)$ ...(2b)

หา $[L\backslash U:b:\bar{c}:\bar{x}]$ โดยใช้ 3s arithmetic จะได้

[0.01 0.01 10,000 10,000 :: 100 2 :: 10,000 1.00 :: 0.00 1.00]

ดังนั้น $\bar{x}_{BP}$ = co.00 1.001’ ...(3)

การคำนวณใน (3) คือ

$\bar{c}_2 = (2-1\bar{c}_1)/(-10,000) = (2-10,000)/(-10,000)$

$\doteq (-10,000)/(-10,000) = 1.0$ ...(4)

$\bar{x}_1 = 10,000-10,000\bar{x}_2 = 10,000-10,000\bar{c}_2 = 0.00$ ...(5)

<u>Solution 2</u> โดยการใช้ Partial Pivoting เราจะได้

$$\hat{L}\backslash\hat{U} = \left[\begin{array}{c|c} 1 & 1 \\ \hline 0.01 & 100 \end{array}\right] \quad \rho_1 \rightleftarrows \rho_2 \qquad \ldots(6a)$$

เกิดความคลาดเคลื่อนจากการปัดเศษในการเก็บ pivot ตัวที่สอง

$\hat{l}_{22} = 100-(0.01)(1) = 99.99 \doteq 1.00 \times 10^{2} = 100 (3s)$ ...(6b)

หา $[\hat{L}\backslash\hat{U}:\hat{b}:\bar{c}:\bar{x}]$ โดยใช้ 3s arithmetic คือ

$$\left[\begin{array}{cc} 0.01 & 1 \\ 100.1 & 100.2 \end{array} \;\vdots\; 2 \;\; 1.00 \;\vdots\; 1.00 \;\; 1.00\right]$$

ดังนั้น $\bar{x}_{PP} = [1.00 \quad 1.00]^t$ ...(7)

การคำนวณใน (7) คือ

$$\bar{c}_2 = (100-0.01\bar{c}_1)/100 = (100-0.02)/100 \doteq 100/100$$

$$= 1.00 \qquad \text{...(8)}$$

$$\bar{x}_1 = 2-1\bar{x}_2 = 2-1\bar{c}_2 = 1.00 \qquad \text{...(9)}$$

การหาผลเฉลยแม่นตรง $\bar{x}$ เราใช้สูตรที่ใช้ A^{-1}

$$\bar{x}_{exact} = A^{-1}b = (-1)/99.99 \begin{bmatrix} 1 & -100 \\ -1 & 0.01 \end{bmatrix} \begin{bmatrix} 100 \\ 2 \end{bmatrix}$$

$$= (-1)/99.99 \begin{bmatrix} -100 \\ -99.96 \end{bmatrix} \doteq \begin{bmatrix} 1.0001 \\ 0.9999 \end{bmatrix} \qquad \text{...(10)}$$

เมื่อเปรียบเทียบ (10) กับ (3) และ (7) เราจะพบว่ามี "serious error" ใน $\bar{x}_{BP}$ ที่ได้จาก Basic Pivoting ส่วน $\bar{x}_{PP}$ ที่ได้จาก Partial Pivoting นั้นมีความแม่นยำถึง 3s

นอกจาก Pivoting Strategy ข้างต้นแล้ว ยังมีการปรับปรุงวิธีการให้มีประสิทธิภาพดีขึ้น เช่น Scaled Partial Pivoting (SPP), Partial Extended Precision (PEP) Strategy วิธีหลังนี้การคำนวณสมาชิก

ของ $\hat{L}\backslash\hat{U}$, $\bar{c}$ และ $\bar{x}$ นั้นทำใน extended precision แต่เก็บค่าโดยใช้ single precision

4.2 การหาผลเฉลยของ AX = B โดยใช้คอมพิวเตอร์

4.2A Full Pivoting Strategy (FP)

The **Full Pivoting Strategy for Selecting the jth Pivot.**

Choose as the jth pivot the element of largest magnitude in the (n-j+1) x (n-j+1) matrix obtained by deleting the rows and columns containing the preceding (j-1) pivot entries from the cofficiient part of current augmented matrix.

เราอาจใช้ Full Pivoting strategy กับการกำจัดของเกาส์ หรือการกำจัดของเกาส์-ชอร์ดอง แต่ไม่ใช้กับ Triangular Decomposition Algorithm

ตัวอย่าง การใช้การกำจัดของเกาส์ โดย Full Pivoting และไม่ใช้ 1-SCALEs เพื่อหาผลเฉลยของ

$$\begin{aligned} -x_1 + x_2 - 4x_3 &= 0 \\ 2x_1 + 2x_2 &= 1 \\ 3x_1 + 3x_2 + 2x_3 &= 1/2 \end{aligned} \qquad \text{ผลเฉลยแม่นตรง คือ } \bar{x} = \begin{bmatrix} 5/4 \\ -3/4 \\ -1/2 \end{bmatrix} \qquad \ldots(1)$$

Solution ทำขั้น (upper triangulation) ดังต่อไปนี้ pivots ถูกวงรอบไว้)

$$\underset{|-4| \text{ is max; } p_1 = -4}{\overset{\text{coefficients}}{\left[\begin{array}{ccccc} -1 & 1 & -4 & : & 0 \\ 2 & 2 & 0 & : & 1 \\ 3 & 3 & 2 & : & \frac{1}{2} \end{array}\right]}} \xrightarrow[\rho_3 + \frac{1}{2}\rho_1]{\rho_2 - 0\rho_1} \underset{|\frac{7}{2}| \text{ is max; } p_2 = \frac{7}{2}}{\overset{\text{zeros below } p_1}{\left[\begin{array}{ccccc} -1 & 1 & -4 & : & 0 \\ 2 & 2 & 0 & : & 1 \\ \frac{5}{2} & \frac{7}{2} & 0 & : & \frac{1}{2} \end{array}\right]}} \xrightarrow[\rho_3 - \frac{4}{7}\rho_2]{\rho_2 \rightleftarrows \rho_3} \underset{\frac{4}{7} \text{ is } p_3}{\overset{\text{zeros below } p_2}{\left[\begin{array}{ccccc} -1 & 1 & -4 & : & 0 \\ \frac{5}{2} & \frac{7}{2} & 0 & : & \frac{1}{2} \\ \frac{4}{7} & 0 & 0 & : & \frac{5}{7} \end{array}\right]}} \qquad (2)$$

ใช้การแทนค่าย้อนหลัง เพื่อแก้สมการ (E_3) หาค่า $\bar{x}_1$ แล้ว แก้สมการ (E_2) หาค่า $\bar{x}_2$ และ (E_1) หาค่า $\bar{x}_3$ ดังต่อไปนี้

$$\bar{x}_1 = (5/7)/(4/7) = 5/4$$

$$\bar{x}_2 = (2/7)\{1/2-(5/2)\bar{x}_1\} = -3/4$$

$$\bar{x}_3 = (-1/4)\{0-\bar{x}_1+\bar{x}_2\} = -1/2$$

ถึงแม้ว่าการหา p_i ที่ใหญ่ที่สุดนั้นเราทำได้ง่ายในการคำนวณด้วยมือ แต่เมื่อใช้เครื่องคอมพิวเตอร์จะเสียเวลามากขึ้นอย่างมาก ดังนั้น Full Pivoting strategy นั้นมักไม่นิยมใช้นอกเสียจากว่าจำเป็นต้องใช้ผลเฉลย $\bar{x}$ ซึ่งมีความแม่นยำสูงสุดเท่าที่จะเป็นไปได้

รูป 4.2-1 และรูป 4.2-2 คือซับรูทีน DECOMP (ซึ่งใช้ Scaled Partial Pivoting (SPP) และ Partial Extended Precision (PEP) Strategy) และซับรูทีน FORBAK สำหรับการหาผลเฉลยของ AX = B

```
      SUBROUTINE DECOMP(N,EPS,A,LUD,IROW,SNGULR,CONDIT,KPVT,S)
      REAL A(N,N), S(N), LUD(N,N), MTHPVT
      INTEGER IROW(N)
      DOUBLE PRECISION ACCUM
      LOGICAL SNGULR
C ----------------------------------------------------------------- C
C GIVEN AN N BY N MATRIX A, THIS SUBROUTINE RETURNS:                C
C       LUD(N,N) = LU-DECOMPOSITION OF A- = L^*U^                   C
C       IROW(N)  = ROW ORDER VECTOR                                 C
C       CONDIT   = CONDITION INDICATOR (0 IF SINGULAR)              C
C                                                                   C
C NOTE: ROW K OF [A^:B^:S^:L^\U^] IS ROW IROW(K) OF [A:B:S:LUD]     C
C                                                                   C
C SNGULR IS SET TO TRUE IF FOR SOME M ALL L^/S^ RATIOS ARE < EPS.   C
C IF A IS NOT SINGULAR, CONDIT = CP(A) = L^(N,N)/S^(N).             C
C ------------------------------------------- VERSION 1: 5/1/81 -   C
C INITIALIZE SNGULR, IROW AND S
      SNGULR = .FALSE.
      DO 10 I=1,N
         IROW(I) = I
C       **SET S(I) = LARGEST ABS VALUE IN ROW I OF A
         S(I) = 0.
         DO 5 J=1,N
            S(I) = AMAX1(S(I),ABS(A(I,J)))
    5    CONTINUE
```

รูป 4.2-1 ซับรูทีน DECOMP (หา LUD และ IROW)

```
   10 CONTINUE
C
C ITERATE:  FORM LUD USING SPP AND PEP STRATEGIES
      DO 100 M=1,N
C       **GET ENTRIES OF COLUMN  M OF L^
         KSTOP = M-l
         DO 25 I=M,N
            II = IROW(I)
            ACCUM = DBLE(A(II,M))
            IF (M.EQ.1) GOTO 20
              DO 15 K=1,KSTOP
                 ACCUM = ACCUN - DBLE(LUD(II,K))*LUD(IROW(K),M)
   15         CONTINUE
   20       LUD(II,M) = SNGL(ACCUM)
   25    CONTINUE
C       **SET CONDIT = LARGEST L^/S^ RATIO OF COLUMN  M Of L^
         CONDIT = 0.
         DO 30 K=M,N
            RATIO= ABS(LUD(IROW(K),M))/S(IROW(K))
            IF (RATIO. LE. CONDIT) GOTO 30
               CONDIT = RATIO
               KPVT = K
   30    CONTINUE
         IM = IROW(KPVT)
         IF (KPVT.EQ. M) GOTO 35
C       **INTERCHANGE ROWS M AND KPVT OF IROW
         IROW(KPVT) = IROW(M)
         IROW(M) = IM
C       **EXIT  CHECK:  EXIT IF A IS NEARLY SINGULAR OR M = N
   35    MTHPVT = LUD(IM,M)
         IF (CONDIT.GT.EPS .AND. ABS(MTHPVT).GT.EPS) GOTO 40
            SNGULR = .TRUE.
   40    IF (SNGULR .OR. CM .EQ. N)) RETURN
C       **GET ENTRIES OF ROW M OF U^(IN ROW IM OF LUD)
         JSTART = M+1
         DO 55 J=JSTART,N
            ACCUM = DBLE(A(IM,J))
            IF CM .EQ.1) GOTO 50
              DO 45 K=1,KSTOP
                 ACCUM = ACCUM - DBLE(LUD(IM,K))*LUD(IROW(K),J)
   45         CONTINUE
   50       LUD(IM,J) = SNGL(ACCUM)/MTHPVT
   55    CONTINUE
  100 CONTINUE
      END
```

รูป 4.2-1(ต่อ)

```
      SUBROUTINE FORBAK(N, P, LUD, IROW, B, XBAR)
      INTEGER IROW(N), P
      REAL B(N,P), XBAR(N,P), LUD(N,N), ITHPVT
      DOUBLE PRECISION ACCUM
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - C
C THIS SUBROUTINE SOLVES AX = B BY SOLVING FIRST L-C = B^            C
C (FORWARD SUBSTN) THEN U-X = CBAR (BACKWARD SUBSTN) USING           C
C LUD AND IROU RETURNED BY SUBROUTINE DECOHP.                        C
C - - - - - - - - - - - - - - - - - - - - - - - - VERSION 1: 5/1/81 C
C  FORWARD SUBSTITUTION: SOLVE L-C = B^ FOR CRAR (PUT IN XBAR)
      DO 40 I=1,N
         II = IROW(I)
         ITHPVT = LUD(II,I)
         KSTOP = I-1
         DO 30 J=1,P
            ACCUM = DBLE(B(II,J))
            IF (I .EQ. 1) GOTO 20
               DO 10 K=1,KSTOP
                  ACCUM = ACCUM - DBLE(LUD(II,K))*XBAR(K,J)
   10          CONTINUE
   20       XBAR(I,J) = SNGL(ACCUM)/ITHPVT
   30    CONTINUE
   40 CONTINUE
      IF (N .EQ. 1) RETURN
C
C  BACKWARD SUBSTITUTION: SOLVE U-X = CBAR FOR DESIRED SOLN XBAR
      DO 70 INDEX=2,N
         I = N-INDEX+1
         KSTART = I+1
         DO 60 J=1,P
            ACCUM = DBLE(XBAR(I,J))
            DO 50 K=KSTART,N
               ACCUN = ACCUM - DBLE(LUD(IROW(I),K))*XBAR(K,J)
   50       CONTINUE
            XBAR(I,J) = SNGL(ACCUM)
   60    CONTINUE
   70 CONTINUE
      RETURN
      END
```

รูป 4.2-2 ซับรูทีน FORBAK (FORWARD และ BACKWARD SUBSTITUTION)

4.3 วิธีการทำซ้ำ (Iterative Methods) สำหรับการหาผลเฉลยของระบบเชิงเส้น

วิธีการตรงต้องการการคำนวณทางคณิตศาสตร์ประมาณ $2n^3/3$ ครั้ง เพื่อหาผลเฉลยของระบบ **Ax** = **b** ขนาด (n x n) นั่นคือ

$$(E_1) \quad a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1$$

$$(E_2) \quad a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2$$

$$\vdots$$

$$(E_n) \quad a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \qquad \ldots(1)$$

จำนวนครั้งดังกล่าวเป็นเหตุให้ n ที่จะปฏิบัติได้นั้นมีขีดจำกัด เราจึงต้องหาวิธีการทำซ้ำที่จะทำให้เราได้คำตอบที่แม่นยำสูงแม้ว่า n จะมีขนาดใหญ่

4.3A วิธียาโคบิ (Jacobi Method) และ วิธีเกาส์-ไซเดล (Gauss-Seidel Method)

ถ้าสมาชิกบน main diagonal ของ A ทุกตัวไม่เป็นศูนย์ แล้ว (E_i) จะถูกแก้เพื่อหา $\bar{x}_i$

$$\bar{x}_i = (1/a_{ii})\{b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}x_j\} \quad ,i=1,\ldots,n \qquad \ldots(2)$$

$\bar{\mathbf{x}}$ satisfies Ax = b <--> $\bar{\mathbf{x}}$ satisfies "fixed-point equation" $x = g(\mathbf{x})$ ...(3a)

โดยที่ $g(\mathbf{x})$ คือ vector function ซึ่งสมาชิกตัวที่ i คือ

$$g_i(x) = (1/a_{ii})\{b_i - \sum_{j \neq i} a_{ij}x_j\} \quad ,i=1,\ldots,n \qquad \ldots(3b)$$

วิธียาโคบิ สำหรับการหาผลเฉลย **Ax** = **b** คือการแก้สมการ (3a) โดยวิธีการแทนค่าซ้ำ (Repeated Substitution) นั่นคือ เริ่มต้นด้วยค่าเดาเริ่มต้น **x** แล้วแทนที่ **x** ด้วย

$$x^{(new)} = g(x) \qquad \ldots(4a)$$

ทำซ้ำไปเรื่อย ๆ จนกระทั่ง $x^{(new)}$ นั้นมีค่าใกล้ $\bar{x}$ อย่างเพียงพอ

วิธีนี้อาจไม่ลู่เข้าเลยก็ได้ แม้ว่าจะใช้ ค่าเดาเริ่มต้นที่ดี (good) และถ้าลู่เข้า โดยทั่ว ๆ ไปจะลู่เข้าแบบเชิงเส้น

วิธีที่ปรับปรุงให้ดีกว่า วิธียาโคบิ คือ **วิธีเกาส์-ไซเดล (Gauss-Seidel)** [หรือเรียกว่า วิธีการแทนค่าสืบเนื่อง (Method of Successive Displacement)] ซึ่งทำโดยการใช้ผลของสมการที่ i ของ (4a) นั่นคือ

$$x_i^{(new)} = (1/a_{ii})\{b_i - \sum_{j \neq i} a_{ij}x_j\} \qquad \ldots(4b)$$

แทนที่ x_i ใน array x ทันทีที่หาค่าได้ ทั้งนี้ $i = 1,\ldots,n$

x ตัวใหม่นี้ คาดว่าจะมีค่าใกล้เคียงกับผลเฉลย $\bar{x}$ มากกว่า x ที่ใช้เพื่อหา $x_i^{(new)}$ ใน (4b) โดยทั่ว ๆ ไปวิธีเกาส์-ไซเดลนั้นคาดว่าจะลู่เข้าเร็วกว่าวิธียาโคบิ และ วิธีเกาส์-ไซเดลนั้นถ้าลู่เข้าจะลู่เข้าแบบเชิงเส้น

ตัวอย่าง การใช้วิธีเกาส์-ไซเดล เพื่อหาผลเฉลยของ Ax = b โดยที่

$$[A:b] = \left|\begin{array}{cccc} 1 & 0 & 3 & : & 2 \\ 5 & 1 & 2 & : & -5 \\ 1 & 6 & 2 & : & -11 \end{array}\right| \quad \text{ผลเฉลยแม่นตรง คือ} \quad \left|\begin{array}{c} -1 \\ -2 \\ 1 \end{array}\right| \qquad \ldots(5)$$

Solution เนื่องจากว่าสมาชิกทุกตัวบน diagonal ไม่เป็นศูนย์ เราอาจแก้สมการ (E_i) for x_i, $i=1,\ldots,n$ ทำให้ได้สมการต่อไปนี้

$$x_1^{(new)} = (1/1)\{2-0x_2-3x_3\} = 2-3x_3$$

$$x_2^{(new)} = (1/1)\{-5-5x_1-2x_3\} = -[5+5x_1+2x_3]$$

$$x_3^{(new)} = (1/2)\{-11-1x_1-6x_2\} = (-1/2)[11+x_1+6x_2] \qquad \ldots(6)$$

เริ่มต้นด้วย $x = x_o$ = CO 0 01' ใช้ (6) ครั้งแรก (first iteration) ได้

$x_1^{(new)} = 2-3(0) = 2$

$x_2^{(new)} = -\{5+5(2)+2(0)\} = -15$

$x_3^{(new)} = (-1/2)\{11+2+6(-15)\} = 36.5$

ดังนั้น $x = x_1$ = [2 -15 38.51'

ใช้ (6) ครั้งที่สอง (second iteration) โดยใช้ $x = x_2$ ได้

$x_1^{(new)} = 2-3(38.5) = -113.5$

$x_2^{(new)} = -\{5+5(-113.5)+2(38.5)\} = 485.5$

$x_3^{(new)} = (-1/2)\{11+(-113.5)+6(485.5)\} = 1405.25$

ดังนั้น $x = x_2$ = C-113.5 485.5 1405.251'

จะเห็นได้ชัดว่า x_k ลู่ออก อย่างไรก็ดี แทนที่จะเพียงแก้สมการ (E_1) สมมุติเราจะทำดังต่อไปนี้

<u>การทำซ้ำของเกาส์-ไซเดล (Gauss-Seidel Iteration) เริ่มต้นที่นี่</u>

แก้สมการ (E_1) เพื่อหา x_3: $x_3^{(new)} = (1/3)\{2-1x_1-0x_2\}$

$= (1/3)\{2-x_1\}$

แล้วแก้ (E_3) เพื่อหา x , : $x_2^{(new)} = (1/6)\{-11-1x_1-2x_3\}$

$= (-1/6)\{11+x_1+2x_3\}$

แล้วแก้ (E_2) เพื่อหา x , : $x_1^{(new)} = (1/5)\{-5-1x_2-2x_3\}$

$= (-1/5)\{5+x_2+2x_3\}$ ...(7)

เริ่มต้นด้วย $x = x_o = [0 \quad 0 \quad 0]'$ ใช้ (7) ในการทำครั้งแรกได้

$$x_3^{(new)} = (1/3)\{2-0\} = 0.66667$$
$$x_2^{(new)} = (-1/6)\{11+0+2(0.66667)\} = -2.0556$$
$$x_1^{(new)} = (-1/5)\{5+(-2.0556)+2(0.66667)\} = -0.85556$$

ดังนั้น $x = x_1 \doteq [-0.65556 \quad -2.0556 \quad 0.66667]'$

ใช้ (7) ในการทำครั้งที่สองได้

$$x_3^{(new)} = (1/3)\{2-(-0.85556)\} \doteq 0.95185$$
$$x_2^{(new)} = (-1/6)\{11-0.85556+2(0.95185)\} \doteq -2.0080$$
$$x_1^{(new)} = (-1/5)\{5+(-2.0080)+2(0.95185)\} \doteq -.97914$$

ตารางต่อไปแสดงผลของการทำซ้ำ 4 ครั้งแรก ซึ่งจะเห็นได้ว่า x_k ลู่เข้าหาผลเฉลยแม่นตรง $\bar{x} = [-1 \quad -2 \quad 1]'$

ตาราง 4.3-1 การทำซ้ำ 4 ครั้งโดยวิธีเกาส์-ไซเดิล

k	สัมประสิทธิ์ของ x_k x_1	x_2	x_3
1	-0.85556	-2.0556	0.66667
2	-0.97914	-2.0080	0.95185
3	-0.99699	-2.0012	0.99305
4	-0.99956	-2.0002	0.99900

สมการ (7) แสดงให้เห็นว่า เราอาจแก้สมการในลำดับใด ๆ สำหรับตัวแปรใด ๆ ก็ได้ แต่ตัวแปรแต่ละตัวจะต้องถูกหาค่าจากการแก้สมการหนึ่งเพียงครั้งเดียว

หมายเหตุ ใน (7) แต่ละสมการ (E_1), (E_2), (E_3) ถูกแก้เพื่อหาค่าตัวแปรซึ่งมีสัมประสิทธิ์ใหญ่ที่สุด การลู่เข้าในตารางที่แล้วนั้นเป็นผลมาจากสิ่งที่จะพิจารณาในหัวข้อถัดไป

Algorithm: Gauss-Scidel Iteration (Method of Successive Displacements)'

Purpose: To solve $Ax = \mathbf{b}$ for $\bar{\mathbf{x}} = A^{-1}\mathbf{b}$ to *NumSig* significant digits. One iteration, starting with $\mathbf{x} = [x_1\ x_2\ .\ x_n]^T$ goes as follows:

(*) Solve equation ($\hat{E}_i$) for variable $\hat{x}_i^{(new)}$ for $i = 1, 2, , n$, replacing $\hat{x}_i$ by $\hat{x}_i^{(new)}$ he/tire calculating $\hat{x}_{i+1}^{(new)}$

In (*), (E,), , (E_n) and $\hat{x}_1$, , $\hat{x}_n$ are preselected rearrangements of the given equations (Et), , (E_n) and the variables x_1, , x_n, respectively. The algorithm terminates after *MaxIt* iterations, or sooner if

$dx_i = x_i^{(new)} - x_i$ is sufficiently small for $i = l, , n$

{*initialize*}
GET n, A, $\mathbf{b}$, [equation parameters)
MaxIt, *NumSig*, {termination parameters)
$\mathbf{x}_0$ {an initial guess of $\bar{\mathbf{x}}$}
RelTol ← $10^{-NumSig}$
$\mathbf{x} \leftarrow \mathbf{x}_0$ {$\mathbf{x} = [x_1\ \ x_2\ \cdots\ x_n]^T$ is the current approximation of $\bar{\mathbf{x}}$}

{*iterate*}
DO FOR $k = 1$ to *MaxIt* UNTIL **termination test** is satisfied
BEGIN {form $\mathbf{x}_k$ and $\mathbf{dx} = \mathbf{x}_k - \mathbf{x}_{k-1}$ in rearranged component order}
DO FOR $i = 1$ TO n
BEGIN
{get $\hat{x}_i^{(new)}$} Solve ($\hat{E}_i$) for $\hat{x}_i^{(new)}$ using components of current $\mathbf{x}$
$d\hat{x}_i \leftarrow \hat{x}_i^{(new)} - \hat{x}_i$
{**update** $\mathbf{x}$} $\hat{x}_i \leftarrow \hat{x}_i^{(new)}$
END
{$x = [x_1\ x_2 \cdots x_n]^T$ is now $\mathbf{x}_k$; and $\mathbf{dx} = [dx_1\ dx_2 \cdots dx_n]^T$}
OUTPUT (k, x_1, x_2, . . . , x_n)
{**termination test:** $|dx_i| \leq RelTol$*max(1, $|x_i|$) for $i = 1, 2, . . . , n$}
END

IF termination **test** was satisfied
THEN OUTPUT ($\mathbf{x}$ approximates $\bar{\mathbf{x}}$ to *NumSig* significant digits)
ELSE OUTPUT (Convergence not evident in *MaxIt* iterations)

รูป 4.3-1 รหัสเทียมสำหรับขั้นตอนวิธี Gauss-Seidel Iteration

4.3B ยุทธวิธีสำหรับการปรับปรุงการลู่เข้า (Convergence)

จาก ith สมการของระบบเชิงเส้น $Ax = b$ คือ

$$(E_i)\quad a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i \qquad \dots(8)$$

เราเรียก x_k ว่า "strictly dominant variable" ของ (E_i) และเรียก a_{ik} [ส.ป.ส.ของ x_k ใน (E_i)] ว่า "strictly dominant entry" ของ $\text{row}_i A$ ถ้า

$$|a_{ik}| > \sum_{j \neq k} |a_{ij}|$$

[นั่นคือ ขนาดของ a_{ik} มากกว่าผลบวกของขนาดของส.ป.ส.ตัวอื่น ๆ ที่เหลืออยู่ใน (E_i)]

และเรียก A ว่าเป็น "strictly dominant matrix" ถ้าแต่ละ row ของ A มี strictly dominant entry อยู่ในคอลัมน์ที่ต่างไปจาก row อื่น ๆ ในกรณีนี้ แต่ละตัวแปร $x_1, \dots, x_n$ คือ strictly dominant variable ของ (E_i) ที่ต่างกัน

ตัวอย่าง เมตริกซ์ A ซึ่งเป็น strictly dominant matrix

$$A = \begin{bmatrix} 1 & 0 & 3 \\ 5 & 1 & 2 \\ 1 & 6 & 2 \end{bmatrix} \qquad \begin{array}{ll} \text{row 1} & |a_{13}| = 3 > |1| + |0| \\ \text{row 2} & |a_{21}| = 5 > |1| + |2| \\ \text{row 3} & |a_{32}| = 6 > |1| + |2| \end{array}$$

ตัวอย่าง เมตริกซ์ที่ไม่เป็น strictly dominant matrix

$$B = \begin{vmatrix} 2 & -4 & 3 \\ 0 & 2 & 5 \\ 1 & -1 & 0 \end{vmatrix} \qquad |b_{12}| = 4 \not> |2| + |3|$$

Dominance Theorem **(D.T.)**

Let Ax = b be a linear system for which A is strictly dominant. If each equation is solve for its strictly dominant variable then the x_k's generated by Gauss-Seidel iteration will converge to $\bar{x} = A^{-1}b$ for any choice of x,.

เมตริกซ์ส่วนมากไม่เป็น strictly dominant matrix อย่างไรก็ดี D.T. แนะว่าเราควรใช้ข้อแนะนำต่อไปนี้ เมื่อตัดสินใจว่าจะหาค่าของตัวแปรตัวใดจาก (E_i), i=1,...,n เมื่อใช้วิธีเกาส์-ไซเดล

กุศโลบายในการเลือกตัวแปร

Try to solve as many equations as possible for the variable having the largest (in magnitude, coefficient.

ถ้า A เป็น strictly diagonally dominant (นั่นคือ a_{ii} เป็น strictly dominant entry ของ $row_i A$, i=1,. . .,n) แล้วรับประกันการลู่เข้าของวิธีเกาส์-ไซเดล ถ้าเรา

แก้สมการ (E_i) เพื่อหา x_i เมื่อ i = 1,..., n ...(9)

ถ้า A เป็นเมตริกซ์สมมาตร ซึ่งเป็น positive definite ($x'Ax > 0$ for all nonzero x) แล้ว (9) จะให้ผลเฉลยที่ลู่เข้าหา $\bar{x}$ สำหรับค่าใด ๆ ของ x_0 (ค่าเดาเริ่มต้น)

4.4 <u>การหาผลเฉลยของระบบไม่เป็นเชิงเส้น</u>

สมการซึ่งประกอบด้วยนิพจน์เช่น

$$x^3,\ y^{-2},\ xy,\ \sqrt{y/x},\ (2y-z)^2,\ \sin x,\ e^{yz},\ z\sqrt{x+y}$$

ถูกเรียกว่าไม่เป็นเชิงเส้น (nonlinear) ใน x, y, z,... เพราะว่าเราไม่อาจเขียนสมการเหล่านั้นในรูป $ax + by + cz + \ldots = \text{constant}$ ซึ่งคือสมการเชิงเส้นในตัวแปร x, y, z,... ได้

ระบบของ n สมการ ในตัวแปร n ตัว $x_1, x_2, \ldots, x_n$ ระบบหนึ่งถูกเรียกว่า **ระบบไม่เป็นเชิงเส้น (nonlinear system)** ถ้ามีสมการหนึ่งหรือมากกว่าหนึ่งสมการไม่เป็นเชิงเส้น

โดยการเอาทุกเทอมที่ไม่เป็นศูนย์ไปไว้ทางซ้ายของสมการ เราอาจเขียนรูปทั่วไปของระบบไม่เป็นเชิงเส้นขนาด (n x n) ดังนี้

$$\begin{array}{l} (E_1)\ f_1(x_1, x_2, \ldots, x_n) = 0 \\ (E_2)\ f_2(x_1, x_2, \ldots, x_n) = 0 \\ \quad\quad \vdots \\ (E_n)\ f_n(x_1, x_2, \ldots, x_n) = 0 \end{array} \quad \text{หรือ} \quad \begin{bmatrix} f_1(x) = 0 \\ f_2(x) = 0 \\ \vdots \\ f_n(x) = 0 \end{bmatrix} \quad \text{โดยที่ } x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \ldots(1)$$

4.4A <u>การใช้วิธีการทางกราฟฟิคเพื่อประมาณค่าผลเฉลยของระบบไม่เป็นเชิงเส้นขนาด (2 x 2)</u>

พิจารณาระบบไม่เป็นเชิงเส้นขนาด (2 x 2)

(E_1) ye- -2 = 0 {นั่นคือ $f_1(x) = 0$ โดยที่ x = [x y]'}

(E_2) $x^2+y-4 = 0$ {นั่นคือ $f_2(x) = 0$ โดยที่ x = cx y]'}

...(2)

ทำการวิเคราะห์ระบบนี้ทางกราฟฟิกโดยพิจารณาว่า

x satisfies (E_1) <--> x อยู่บน exponential curve $y = 2e^{-x}$

x satisfies (E_2) <--> x อยู่บน parabola $y = 4-x^2$

ดังนั้น $\bar{x} = [\bar{x}\ \ \bar{y}]'$ สมนัยกับจุดซึ่ง curve ทั้งสองตัดกัน ดังแสดงในรูปในหัวข้อ 4.4B ซึ่งจะเห็นว่า (2) มีสองผลเฉลย คือ

$\bar{x}_1 \approx$ [-0.6 3 . 7 1 ' และ $\bar{x}_2 \approx$ [1.9 0 . 4 3

ในการหาผลเฉลยของระบบไม่เป็นเชิงเส้น (2) ด้วยความแม่นยำสูงกว่า เราขจัด y จากระบบข้างต้นจะได้

$(4-x^2)e^x-2 = 0$ ...(3)

แล้วเราอาจจะแก้สมการด้วยวิธีในบทที่ 2 โดยการใช้ค่าเดาเริ่มต้น $x_0 = -0.6$ และ $x_0 = 1.9$ ซึ่งได้จากรูปในหัวข้อ 4.4B อย่างไรก็ดี เราไม่อาจขจัด x หรือ y จากระบบไม่เป็นเชิงเส้น ต่อไปนี้ได้

(E_1) $xe^y-x^5+y+3 = 0$

(E_2) $x+y+\tan x-\sin y = 0$ ...(4)

ดังนั้นจึงสำคัญมากที่จะต้องมีวิธีสำหรับการหาผลเฉลยของระบบไม่เป็นเชิงเส้น ในรูป (1) เราจะได้กล่าวถึงวิธีหนึ่งในหัวข้อถัดไป

4.4B วิธีของนิวตัน-ราฟสันสำหรับการหาผลเฉลยของระบบไม่เป็นเชิงเส้น (Newton-Raphson Method for Nonlinear Systems :NRSYS)

สมมุติว่า $\bar{x} = [\bar{x}_1\ \ \bar{x}_2\ \ldots\ \bar{x}_n]'$ ...(5a)

เป็นผลเฉลยของระบบไม่เป็นเชิงเส้น (1) นั่นคือหมายความว่า

$$f_1(\bar{x}) = f_2(\bar{x}) = \ldots = f_n(\bar{x}) = 0 \qquad \ldots(5b)$$

ถ้า x ประมาณค่า $\bar{x}$ แล้ว increment จาก x to $\bar{x}$ จะเขียนแทนด้วย

$$\Delta x = \bar{x} - x = \begin{bmatrix} \bar{x}_1 - x_1 \\ \bar{x}_2 - x_2 \\ \vdots \\ \bar{x}_n - x_n \end{bmatrix} = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{bmatrix} \qquad \ldots(6a)$$

โดยที่ Δx_j คือ increment จาก x_j ไป $\bar{x}_j$ สำหรับ $j=1,\ldots,n$

ปัญหาของเราคือหา Δx ซึ่งจะทำให้

$$\bar{x} = x + \Delta x \qquad \ldots(6b)$$

แทนที่เราจะหา exact increment Δx ซึ่งสอดคล้องกับ

$$f_i(x + \Delta x) = 0, i = 1,\ldots,n \qquad \ldots(7a)$$

เราจะหาค่าประมาณของ Δx นั่นคือหา dx

$$dx = [dx_1\ dx_2\ \ldots\ dx_n]'$$

ซึ่งสอดคล้องกับระบบซึ่งแก้ได้ง่ายกว่า

$$\{\text{linear approximation of } f_i(x+dx)\} = 0, \ i=1,\ldots,n \qquad \ldots(7b)$$

นั่นคือ dx สอดคล้องกับ approximating system

$$f_1(x)+(\partial f_1/\partial x_1)dx_1+(\partial f_1/\partial x_2)dx_2+\ldots+(\partial f_1/\partial x_n)dx_n = 0$$
$$f_2(x)+(\partial f_2/\partial x_1)dx_1+(\partial f_2/\partial x_2)dx_2+\ldots+(\partial f_2/\partial x_n)dx_n = 0$$
$$\vdots$$
$$f_n(x)+(\partial f_n/\partial x_1)dx_1+(\partial f_n/\partial x_2)dx_2+\ldots+(\partial f_n/\partial x_n)dx_n = 0$$
$$\ldots(8a)$$

โดยที่ partial derivatives $\partial f_i/\partial x_j$ ถูกหาค่าจากค่า x ที่ทราบค่าอยู่แล้ว อาจเขียน (8a) ในรูปเมตริกซ์ดังนี้

$$
\underbrace{\begin{bmatrix} \partial f_1(x)/\partial x_1 & \partial f_1(x)/\partial x_2 & \cdots & \partial f_1(x)/\partial x_n \\ \partial f_2(x)/\partial x_1 & \partial f_2(x)/\partial x_2 & \cdots & \partial f_2(x)/\partial x_n \\ \vdots & & & \\ \partial f_n(x)/\partial x_1 & \partial f_n(x)/\partial x_2 & \cdots & \partial f_n(x)/\partial x_n \end{bmatrix}}_{J = f'(x)} \underbrace{\begin{bmatrix} dx_1 \\ dx_2 \\ \vdots \\ dx_n \end{bmatrix}}_{dx} \quad \underbrace{\begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix}}_{f(x)}
$$

(**J = เมตริกซ์จาโคเบียน (Jacobian matrix) ของ f ที่ x**) ...(8b)

เมื่อ $f'(x)$ เป็น nonsingular แล้ว $dx = -f'(x)^{-1}f(x) = -J^{-1}f(x)$

ถ้า x ใกล้ $\bar{x}$ มากพอ ดังนั้น linear approximation

$$f_i(x+\Delta x) \approx f_i(x)+(\partial f_i(x)/\partial x_1)dx_1+\ldots+(\partial f_i(x)/\partial x_n)dx_n$$

แม่นยำสำหรับ $i = 1,\ldots,n$ แล้ว Δx จะถูกประมาณด้วย dx

ดังนั้นโดย (8b)

$$\bar{x} = x + \Delta x \approx x + dx = x - f'(x)^{-1}f(x) \qquad \ldots(9)$$

และสูตรสำหรับการทำซ้ำ คือ

$$\boxed{x_{k+1} = x_k + dx_k \text{ โดยที่ } dx_k = -f'(x_k)^{-1}f(x_k)} \qquad (10)$$

เราเรียกสูตร (10) ว่า **วิธีของนิวตัน-ราฟสันสำหรับการหาผลเฉลยของระบบไม่เป็นเชิงเส้น (Newton-Raphson Method for Nonlinear Systems : NRSYS)**

ในหน้าถัดไปแสดง NRSYS Algorithm

Algorithm: NRSYS (Newton-Raphson Method for **Nonlinear** System:.,

Purpose: To solve a nonlinear system of n equations in n unknowns:

$$f_i(x_1, \ldots, x_n) = 0, \qquad i = 1, 2, \ldots, n$$

The method uses the arrays

$\mathbf{x} = [x_1 \quad x_2 \ldots \quad x_n]^T$ {current approximation of $\mathbf{x}$}
$\mathbf{f} = [f_1(\mathbf{x}) \quad f_2(\mathbf{x}) \ldots f_n(\mathbf{x})]^T$ {array of $f_i(\mathbf{x})$ values, i.e., $\mathbf{f}(\mathbf{x})$}
$\mathbf{dx} = [dx_1 \quad dx_2 \ldots \quad dx_n]^T$ {$\mathbf{dx} = \mathbf{x}_k - \mathbf{x}_{k-1}$}

and the matrix

$J = (\partial f_i(\mathbf{x})/\partial x_j)_{n \times n}$ {the Jacobian matrix at x, i.e., $\mathbf{f}'(\mathbf{x})$}

to find a solution $\bar{\mathbf{x}}$ to *NumSig* significant digits in *MaxIt* iterations.

{*initialize*}
GET n, *MaxIt*, *NumSig*, $\mathbf{x}_0$ {initial guess}
RelTol ← $10^{-NumSig}$; $\mathbf{x} \leftarrow \mathbf{x}_0$

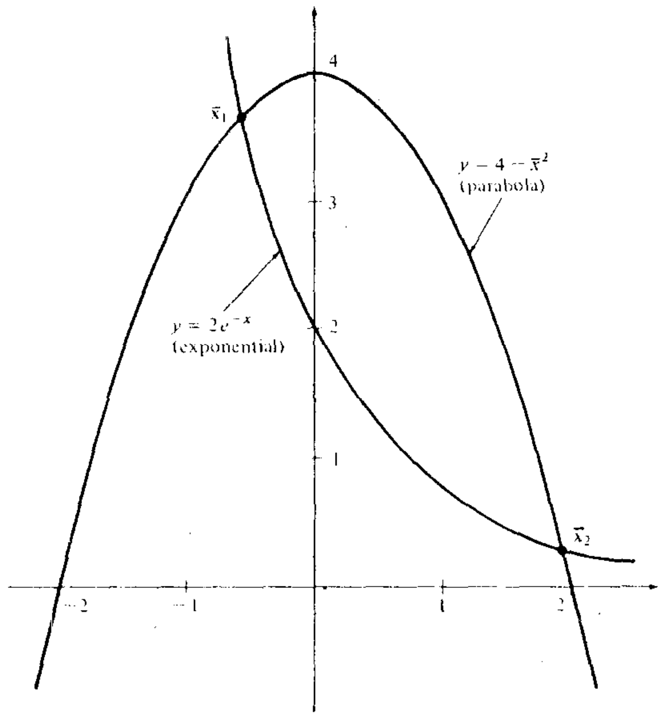
{*iterate*}
DO FOR k = 1 TO *MaxIt* UNTIL termination test is satisfied
BEGIN
f←f(x); $J \leftarrow \mathbf{f}'(\mathbf{x})$ {Evaluate f and J at the current $\mathbf{x}$}
(get dx} Solve the linear system J **dx** = $-\mathbf{f}$ **for dx**
(get x} $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{dx}$ {$\mathbf{x} = [x_1 \cdots x_n]^T$ **is now** $\mathbf{x}_k$}
OUTPUT $(k, x_1, \ldots, x_n)$
(termination test: $|dx_i| \leqslant RelTol * \max(1, |x_i|)$ for $i = 1, 2, \ldots, n$}
END

IF **termination test was satisfied**
THEN OUTPUT ($\mathbf{x}$ approximates $\bar{\mathbf{x}}$ to *NumSig* significant digits)
ELSE OUTPUT (Convergence not apparent in *MaxIt* Iterations)

{*Note:* In the **get dx** step, **dx** should be obtained as $-J^{-1}\mathbf{f}$ using the formula for J^{-1} if $n = 2$, and by solving $J\,\mathbf{dx} = -\mathbf{f}$ directly (*without finding* J^{-1}) if $n \geqslant 3$}

รูป 4.4-1 รหัสเทียมสำหรับขั้นตอนวิธี Newton-Raphson สำหรับระบบไม่เป็นเชิงเส้น

<u>ตัวอย่าง</u> จงใช้ NRSYS เพื่อหา 5s ผลเฉลย $\bar{x}_1$ และ $\bar{x}_2$ ของรูปต่อไปนี้



รูป 4.4-2 รูปแสดงการประมาณค่าผลเฉลยของ $ye^x - 2 = 0,\ x^2 + y - 4 = 0$

<u>Solution</u> ระบบไม่เป็นเชิงเส้น $f(x) = 0$

โดยที่ $x' = [x \quad y]$

$$f(x) = [ye^x - 2 \quad x^2 + y - 4]'$$

เมตริกซ์จาโคเบียน (Jacobian matrix) คือ

$$J = f'(x) = \begin{bmatrix} \partial(ye^x-2)/\partial x & \partial(ye^x-2)/\partial y \\ \partial(x^2+y-4)/\partial x & \partial(x^2+y-4)/\partial y \end{bmatrix} = \begin{bmatrix} ye^x & e^x \\ 2x & 1 \end{bmatrix}$$

เพราะว่า $\det[f'(x)] = (y-2x)e^x$ เราควรพยายามให้ x_k ห่างจากเส้น $y = 2x$

สำหรับ $\bar{x}_1$: เริ่มจาก $x_0 = [-0.6 \quad 3.7]'$ ซึ่งได้จากรูป NRSYS เริ่มต้นดังนี้

$$k = 0:\ f(x_o) = \begin{bmatrix} (3.7)e^{-0.6}-2 \\ (-0.6)^2+3.7-4 \end{bmatrix} = \begin{bmatrix} 0.0306031 \\ -0.6 \end{bmatrix}$$

$$f'(x_o) = \begin{bmatrix} 3.7e^{-0.6} & e^{-0.6} \\ 2(-0.6) & 1 \end{bmatrix} = \begin{bmatrix} 2.03060 & 0.548812 \\ -1.2 & 1 \end{bmatrix}$$

เพราะว่า $n = 2$ วิธีที่ง่ายที่สุดในการหาผลเฉลยของ $f'(x_o)dx_o = -f(x_o)$ คือการใช้สูตรเพื่อหา $f'(x_o)^{-1}$ แล้วหา dx_o โดยผลคูณ $-f'(x_o)^{-1}f(x_o)$ ดังนั้น

$$dx_o = -(1/2.68917) \begin{bmatrix} 1 & -0.548812 \\ 1.2 & 2.03060 \end{bmatrix} \begin{bmatrix} 0.0306031 \\ 0.06 \end{bmatrix} = \begin{bmatrix} 0.000865 \\ -0.05896 \end{bmatrix}$$

$$x_1 = x_o + dx_o = \begin{bmatrix} -0.6 \\ 3.7 \end{bmatrix} + \begin{bmatrix} 0.000865 \\ -0.05896 \end{bmatrix} = \begin{bmatrix} -0.599135 \\ 3.64104 \end{bmatrix}$$

การทำซ้ำครั้งที่สองให้ผลดังนี้

$$k = 1:\ f(x_1) = \begin{bmatrix} -2.73173E-5 \\ 5.91353E-7 \end{bmatrix},\ f'(x_1) = \begin{bmatrix} 1.99997 & 0.549286 \\ -1.19827 & 1 \end{bmatrix}$$

$$dx_1 = (-1/2.65816)\begin{bmatrix} 1 & -0.549286 \\ 1.19827 & 1.99997 \end{bmatrix}\begin{bmatrix} -2.73173E-5 \\ 5.91353E-7 \end{bmatrix} = \begin{bmatrix} 0.000010 \\ 0.000012 \end{bmatrix}$$

$$x_2 = x_1 + dx_1 = \begin{bmatrix} -0.599135 \\ 3.64104 \end{bmatrix} + \begin{bmatrix} 0.000010 \\ 0.000012 \end{bmatrix} = \begin{bmatrix} -0.599125 \\ 3.64105 \end{bmatrix}$$

dx_1 มีสมาชิกที่มีค่าต่ำ แสดงว่า x_1 มีความแม่นยำประมาณ 5d และเนื่องจากการลู่เข้าเป็นแบบอันดับสอง x_2 ควรมีความแม่นยำอย่างน้อย 5s ตามต้องการ ซึ่งจะเห็นจริงในการทำการทำซ้ำครั้งต่อไป

สำหรับ $\bar{x}_2$: เริ่มจาก x_0 = [1.9 0.4]' และทำการคำนวณในทำนองเดียวกันจะได้

$$dx_0 = -f'(x_0)^{-1}f(x_0) = \begin{bmatrix} 0.02672 \\ -0.111553 \end{bmatrix}, \quad x_1 = x_0 + dx_0 = \begin{bmatrix} 1.92672 \\ 0.288447 \end{bmatrix}$$

$$dx_1 = -f'(x_1)^{-1}f(x_1) = \begin{bmatrix} -0.00098 \\ 0.003086 \end{bmatrix}, \quad x_2 = x_1 + dx_1 = \begin{bmatrix} 1.92574 \\ 0.291533 \end{bmatrix}$$

$$dx_2 = -f'(x_2)^{-1}f(x_2) = \begin{bmatrix} 0.00000 \\ 0.00000 \end{bmatrix}, \quad x_3 = x_2 \; t \; dx_2 = \begin{bmatrix} 1.92574 \\ 0.291536 \end{bmatrix}$$

x_2 มีความแม่นยำถึง 5s ด้วย

4.4C ข้อควรพิจารณาเมื่อใช้ NRSYS

พิจารณาในทางปฏิบัติ NRSYS มีข้อจำกัด 3 ประการคือ

1) ความจำเป็นที่จะต้องใช้ค่าเดาเริ่มต้นที่ดี

2) เสียเวลาในการหา n^2 Partial Derivative Functions $\partial f_i/\partial x_j$ สำหรับ Jacobian matrix **f'(x)** อย่างมาก

3) เสียเวลาในการแทนค่า x ใน **f'** แล้วหาผลเฉลยของระบบเชิงเส้น **f'(x)dx** = -f(x) สำหรับแต่ละครั้งของการทำซ้ำอย่างมาก

แบบฝึกหัดบทที่ 4

4.1 In a) and b), solve the equation so that you are assured that Gauss-Seidel iteration will converge. Then, with $x_0 = 0$, do four iterations.

a) $2x, \ -6x, - x_3 = -3$

$-8x,, +3x_2 \ t \ x_3 = -4$

$x_1 \ t \ x_2 -3x_3 = 5$

b)
$$
\begin{array}{rrrrrcr}
x_1 & - x_2 & & +3x_4 & & & 1 \\
5x_1 & & + x_3 & & - x_5 & = & 3 \\
 & 4x_2 & + x_3 & - x_4 & & = & -\ 1 \\
 & & 3x_3 & t\ x_4 & - x_5 & = & -4 \\
x_{\|} & + x_2 & & & -4x_5 & = & -3
\end{array}
$$

Are coefficient matrices in a) and b) strictly dominant matrices?

4.2 Let

$$
A = \begin{bmatrix} 5 & 3 & -2 \\ 3 & 2 & -2 \\ -2 & -2 & 5 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 2 & 3 & -1 \\ 3 & 4 & -2 \\ -1 & -2 & 1 \end{bmatrix}
$$

a) Show that $\bar{\bar{x}}$ = [1 -1 11' is the solution of both Ax = CO -1 5]' and Bx = C-2 -3 21'.

b) For both systems in part a), solve (E_i) for x_i, i=1,2,3 and perform two iterations of Gauss-Seidel iteration starting with $x_0 = 0$. Why should you expect the iteration for Ax = b to converge if continued ?

4.3 Use graphical methods to estimate all roots of the given nonlinear systems.

a) $2 \ln y - x = 0$

$xy - y - 1 = 0$

b) $x + y^2 - 9 = 0$

$y - \ln x = 0$

c) $x^2 - y^2 - 4 = 0$

$y + 1 - 2 \sin x = 0$

Hint a) Sketch $y = e^{(x/2)}$ and $y = 1/(x-1)$

4.4 a) one root = Cl.4777 2.09351

b) Sketch $x = 9-y'$ and $y = \ln x$

4.4 b) two roots = C5.8665 1.76961 and

C0.050206 -2.99161

ci Sketch $(x/2)^2 - (y/2)^2 = 1$ and $y \sin x - 1$

4.4 c) two roots = [2.1207 0.70517] and

C-2.7124 -1.83231

4.4 Use NRSYS with your answers to Exercise 4.3 as initial guesses to find to 5s all roots of the systems a)-c) of Exercise 4.3.

4.5 For the system $x^2 - 3 \sin y - z^2 = 0$

$z - 2xy + 1 = 0$

$e^{x+y} + z^2 = 0$

a) Find the Jacobian matrix (J).

b) Use NRSYS with $x_0 = 0$ to find x,.

(Note: You can solve $f'(x_0)dx_0 = -f(x_0)$ almost by inspection.)