รายการอ้างอิง

ภาษาไทย

ธีระพร วีรถาวร. การอนุมานเชิงสถิติขั้นกลาง. พิมพ์ครั้งที่ 2. กรุงเทพมหานคร: สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2536.

ประชุม สุวัตถี. ทฤษฎีการอนุมานเชิงสถิติ. กรุงเทพมหานคร: สำนักพิมพ์ไอเดียนสโตร์, 2527.

ศิริจันทร์ ทองประเสริฐ. การจำลองแบบปัญหา. กรุงเทพมหานคร: สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2529.

ศิริรัตน์ วงศ์ประกรณ์กุล. การทดสอบการแจกแจงไวบูลย์และการแจกแจงกอมเพิร์ตด้วยวิธีทดสอบเทียบความกลมกลืนเมื่อข้อมูลถูกตัดทิ้งอย่างมาก. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2539.

ภาษาอังกฤษ

Harder, Melinda H. Minimum Distance Estimators of Scale with Censored Data. Journal of the American Statistical Association 87, No.419 (September 1992): 832-843.

Hogg, Robert V. and Klugman, Stuart A. Loss Distributions. New York: John Wiley & Sons, 1984.

Hossack, I.B., Pollard, J.H., and Zehnwirth, B. Introductory statistics with applications in general insurance. New York: Cambridge University Press, 1983.

Johnson, Norman L. and Kotz, Samuel. Continuous univariate distributions-1. New York: John Wiley & Sons, 1970.

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

การจำลองสถานการณ์ที่ต้องการศึกษา โดยใช้เทคนิคการจำลองแบบมอนติคาร์โล ซึ่งมีรายละเอียดและขั้นตอนในการจำลองดังนี้

เทคนิคการจำลองแบบมอนติคาร์โล

เทคนิคที่ใช้ในการแก้ปัญหาทางคณิตศาสตร์มีอยู่หลายวิธี วิธีการจำลองโดยใช้การจำลองแบบมอนติคาร์โล เป็นวิธีการหนึ่งที่นิยมใช้แก้ปัญหากันอย่างแพร่หลายในปัจจุบัน ซึ่งหลักการของ เทคนิคการจำลองแบบมอนติคาร์โลเป็นการจำลองตัวเลขสุ่ม(Random Number) มาช่วยในการหาคำตอบของปัญหาที่ต้องการศึกษา

ขั้นตอนของวิธีการจำลองแบบมอนติคาร์โล ที่ใช้กันในปัจจุบันแบ่งเป็น 3 ขั้นตอน ดังนี้

1. การสร้างตัวเลขสุ่ม การใช้ตัวเลขสุ่มเป็นสิ่งที่สำคัญมากในเทคนิคนี้ เพราะว่าหลักการของการจำลองแบบมอนติคาร์โลนั้น จะใช้ตัวเลขสุ่มมาช่วยในการหาคำตอบของปัญหา โดยลักษณะของตัวเลขสุ่มที่นำมาใช้ จะมีการแจกแจงแบบสม่ำเสมอ(Uniform Distribution) ในช่วง(0,1) สำหรับวิธีการสร้างตัวเลขสุ่ม มีผู้เสนอไว้หลายวิธี แต่วิธีที่ดีนั้นลักษณะของเลขสุ่มที่ถูกสร้างขึ้นจะต้องมีการแจกแจงแบบสม่ำเสมอในช่วง(0,1) ตัวเลขสุ่มแต่ละตัวต้องเป็นอิสระต่างกันและช่วงยาวก่อนจะเกิดเลขสุ่มซ้ำ (มีวัฏจักรยาว)

2. การนำตัวเลขสุ่มมาประยุกต์ใช้กับปัญหาที่ต้องการศึกษา ซึ่งขั้นตอนนี้ขึ้นอยู่กับลักษณะของปัญหา บางปัญหาอาจจะไม่ใช้ตัวเลขสุ่มโดยตรงแต่จะนำไปผลิตเลขสุ่มที่มีการแจกแจงแบบอื่นต่อไป

3. การทดลองกระทำเมื่อนำตัวเลขสุ่มมาประยุกต์ให้เข้ากับปัญหาที่ต้องการศึกษาได้แล้วขั้นต่อไปคือ การทดลองโดยใช้กระบวนของการสุ่ม(Random Process) มากระทำในลักษณะซ้ำๆกันหลาย ๆ ครั้ง เพื่อหาคำตอบที่ต้องการ

การผลิตเลขสุ่มที่มีการแจกแจงสม่ำเสมอในช่วง(0,1)

การผลิตเลขสุ่มตัวเลขสุ่มที่ผลิตขึ้นต้องมีลักษณะความเป็นอิสระซึ่งกันและกัน และมีการแจกแจงแบบสม่ำเสมอในช่วง(0,1)

วิธีการผลิตเลขสุ่มแบบ Multiplicative Congruential Method จะผลิตตัวเลขสุ่มจากสมการ

$$X_i = (a\ X_{i-1}) \bmod M \quad , i=1,2,3,...$$

เมื่อ X_i เป็นเลขสุ่มตัวที่ i

X_0 เป็นตัวเลขค่าเริ่มต้น

a เป็นค่าคงที่

mod M หมายถึง ค่า $(a\ X_{i-1})$ หารด้วย M จนกระทั่งเหลือเศษน้อยกว่าค่า M เลขที่เหลือเศษจึงเป็นเลขสุ่มของเลขสุ่มตัวถัดไปคือ X_i จากสมการนี้ X เมื่อเริ่มค่า X_0 เป็นค่าเริ่มต้น(initial value หรือ seed) จะได้ตัวเลขสุ่ม $X_1, X_2, X_3, ...$ ตามลำดับ เป็นเลขจำนวนเต็มที่มีค่าอยู่ระหว่าง 0 ถึง M-1 ค่าตัวเลขสุ่มที่ได้เป็นค่าที่ไม่ต่อเนื่อง ซึ่งการกำหนดค่า M , a และ X_0 จึงมีความสำคัญในการผลิตเลขสุ่ม โดยการที่จะผลิตเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอในช่วง(0,1) ต้องกำหนดค่า M ให้มีค่าของจำนวนเต็มที่ใหญ่ที่สุดและเป็นเลขคี่ ที่สามารถคำนวณได้จากเครื่องคอมพิวเตอร์ โดยที่ $M = 2^b$ เมื่อ b เป็นค่าความยาว 1 word หรือจำนวน bit ใน 1 word ของเครื่องคอมพิวเตอร์ 32 bit ซึ่ง bit สุดท้าย 1 bit ใช้สำหรับแสดงเครื่องหมาย ดังนั้นเลขจำนวนเต็มที่ใหญ่ที่สุดใน 1 word และเป็นเลขคี่ที่คอมพิวเตอร์ได้รับคือ 2^{b-1}-1 เท่ากับ 2^{31}-1 =2147483647 นั่นคือค่า M ควรมีค่าเท่ากับ 2147483647 การผลิตตัวเลขสุ่มของวิธีนี้ที่ได้ผ่านการทดสอบแล้วอย่างมากจะกำหนดค่า a เท่ากับ 7^5 = 16807 ซึ่งเป็นค่าคงที่ และค่า X_0 มีค่าเป็นเลขจำนวนเต็มบวกที่เป็นเลขคี่

จากวิธีการผลิตเลขสุ่มดังกล่าวเขียนโปรแกรมฟังก์ชันการสร้างเลขสุ่ม โดยใช้ภาษาฟอร์แทรน ดังนี้

```
      FUNTION RAND(IX)
      IX=IX*16807
      IF (IX) 1,2,2
1     IX=IX+2147483647+1
2     RAND=RAND/2147483647
      RETURN
      END
```

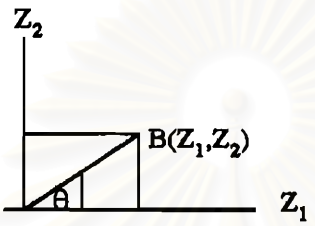
โปรแกรมฟังก์ชัน RAND(IX) โดย IX เป็นตัวเลขค่าเริ่มต้นที่รับเข้ามาในฟังก์ชันเพื่อจะได้ตัวเลขสุ่ม RAND ที่มีค่าในช่วง(0,1)

การผลิตเลขสุ่มที่มีการแจกแจงปกติ

จะใช้เทคนิคการแปลงโดยตรงจาก

$$\phi(x) = \int_{-\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx \qquad ,-\infty < x < \infty$$

โดย Box และ Muller(ค.ศ.1958) สร้างเลขสุ่มที่มีการแจกแจงแบบปกติมาตรฐานที่มีค่าเฉลี่ย 0 และค่าความแปรปรวน 1 พร้อมๆ กัน 2 ค่าดังนี้



เมื่อ $Z_1 = B \cos\theta$

$Z_2 = B \sin\theta$

ดังนั้น $B = Z_1^2 + Z_2^2$ จะมีการแจกแจงแบบไคสแควร์(Chi-square Distribution) ระดับของความเป็นอิสระ 2 ซึ่งก็คือการแจกแจงแบบเอ็กซ์โปเนนเชียล(Exponential Distribution) ที่มีค่าเฉลี่ยเท่ากับ 2 แล้วจะได้ค่ารัศมี B ดังนี้

$$B = (-2 \ln R)^{1/2}$$

เมื่อ R เป็นเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอในช่วง(0,1)

พิจารณาการสมมาตรของการแจกแจงแบบปกติ จะได้ θ มีการแจกแจงแบบสม่ำเสมอ ระหว่าง 0 กับ 2π เรเดียน ซึ่งค่า B และ θ เป็นอิสระกัน(Mutually Independent)

$$Z_1 = (-2 \ln R_1)^{1/2} \cos(2\pi R_2)$$

$$Z_2 = (-2 \ln R_1)^{1/2} \sin(2\pi R_2)$$

เมื่อได้เลขสุ่มที่มีการแจกแจงปกติมาตรฐาน(Standard-Normal Distribution) แล้วทำการแปลงค่าเลขสุ่มดังกล่าวให้เป็นเลขสุ่มที่มีการแจกแจงปกติ มีค่าเฉลี่ย μ และค่าเบี่ยงเบนมาตรฐาน σ ดังนี้

$$X_1 = Z_1 \sigma + \mu$$

$$X_2 = Z_2 \sigma + \mu$$

โปรแกรมย่อยที่ใช้ในการสร้างเลขสุ่มที่มีการแจกแจงปกติ มีค่าเฉลี่ย μ และค่าเบี่ยงเบนมาตรฐาน σ คือ

```
      SUBROUTINE NORMAL(AMEW,SIGMA,ANORM)
      COMMON/SEED/IX,KK
      PI = 3.1415927
      IF (KK.EQ.1) GOTO 10
      RONE = RAND(IX)
      RTWO = RAND(IX)
      ZONE = SQRT(-2*ALOG(RONE))*COS(2*PI*RTWO)
      ZTWO = SQRT(-2*ALOG(RONE))*SIN(2*PI*RTWO)
      ANORM = ZONE*SIGMA+AMEW
      KK =1
      GOTO 20
10    ANORM = ZTWO*SIGMA+AMEW
      KK = 0
20    RETURN
      END
```

การผลิตเลขสุ่มที่มีการแจกแจงลอกนอร์มอล

ขั้นแรกทำการผลิตเลขสุ่มที่มีการแจกแจงปกติ เมื่อได้เลขสุ่มดังกล่าวแล้วทำการแปลงค่าเลขสุ่มให้เป็นเลขสุ่มที่มีการแจกแจงลอกนอร์มอล ที่มีค่าเฉลี่ยของ ln X และค่าเบี่ยงเบนมาตรฐานของ ln X เป็น μ และ σ ตามลำดับ จากฟังก์ชันดังนี้

$$X = EXP(Z\,\sigma + \mu)$$

ในโปรแกรมที่ใช้สร้างเลขสุ่มมีฟังก์ชันดังนี้

```
ALOGN = EXP(ANORM)
```

การผลิตเลขสุ่มที่มีการแจกแจงโลจิสติค

จะใช้เทคนิคการแปลงผกผันโดยตรงจาก

$$F(x) = \frac{1}{1+e^{-(x-\alpha)/\beta}}$$

โดยที่ F(x) เป็นเลขสุ่มที่มีการแจกแจงสม่ำเสมอช่วง(0,1) ในที่นี้ใช้สัญลักษณ์เป็น R แทน F(x)

ทำการแปลงค่าให้เป็นเลขสุ่มที่มีการแจกแจงโลจิสติค ที่มีพารามิเตอร์ α และ β

$$X = -\beta\,(\ln(1-R)-\ln R) + \alpha$$

โปรแกรมย่อยที่ใช้ในการสร้างเลขสุ่มที่มีการแจกแจงโลจิสติค ที่มีพารามิเตอร์ α และ β คือ

```
      SUBROUTINE LOGIST(ALPHA,BETA,GISTIC)
      COMMON/SEED/IX
111   R = RAND(IX)
      IF ((R.LE.0).OR.(R.GE.1.0)) GOTO 111
      GISTIC = -BETA*(ALOG(1-R)-ALOG(R))+ALPHA
      RETURN
      END
```

การประมาณค่าพารามิเตอร์เริ่มต้นด้วยวิธีโมเมนต์

วิธีประมาณค่าด้วยโมเมนต์

X เป็นตัวแปรสุ่มของการแจกแจง

μ_r เป็นโมเมนต์ที่ r ของ X

M_r เป็นโมเมนต์ที่ r ของตัวอย่าง

ซึ่ง $\mu_r = E(X^r)$

และ $M_r = \sum_{i=1}^{n} \frac{X_i^r}{n}$ เมื่อ n เป็นขนาดตัวอย่าง

ให้ $\mu_r = M_r$ เมื่อ r=1,2,3,...

จากหลักการนี้ประมาณค่าพารามิเตอร์เริ่มต้น 2 ค่าจะใช้

$$\mu_1 = M_1$$

และ $$\mu_2 = M_2$$

แก้สมการหาค่าประมาณพารามิเตอร์ซึ่งใช้เป็นค่าเริ่มต้นในการประมาณค่าต่อไป

โปรแกรมย่อยที่ใช้ในการประมาณค่าเริ่มต้นสำหรับการแจกแจงปกติ

```
      SUBROUTINE MOMENT(N,XMEW0,XSIG0)
      COMMON/SAMP/X(500)
      SUMX=0.0
      SUMX2=0.0
```

```
      DO 10 I=1,N
      SUMX=SUMX+X(I)
      SUMX2=SUMX2+(X(I)**2)
10    CONTINUE
      EX = SUMX/N
      VARX = (SUMX2-(SUMX**2/N))/(N-1)
      XMEW0 = EX
      XSIG0 = SQRT(VARX)
      RETURN
      END
```

โปรแกรมย่อยที่ใช้ในการประมาณค่าเริ่มต้นสำหรับการแจกแจงลอกนอร์มอล

```
      SUBROUTINE MOMENT(N,XMEW0,XSIG0)
      COMMON/SAMP/X(500)
      SUMX=0.0
      SUMX2=0.0
      DO 10 I=1,N
            SUMX=SUMX+X(I)
            SUMX2=SUMX2+(X(I)**2)
10    CONTINUE
      EX=SUMX/N
      EX2=SUMX2/N
      BSIG0 = ALOG(EX2)-2*ALOG(EX)
      XSIG0 = SQRT(BSIG0)
      XMEW0 = ALOG(EX)-(BSIG0/2)
      RETURN
      END
```

โปรแกรมย่อยที่ใช้ในการประมาณค่าเริ่มต้นสำหรับการแจกแจงโลจิสติค

```
      SUBROUTINE MOMENT(N,ALPHA0,BETA0)
      COMMON/SAMP/X(500)
      PI = 3.1415927
```

```
      SUMX = 0.0
      SUMX2 = 0.0
      DO 10 I=1,N
          SUMX = SUMX+X(I)
          SUMX2 = SUMX2+(X(I)**2)
10    CONTINUE
      EX=SUMX/N
      EX2=SUMX2/N
      BBETA = 3*(EX2-(EX**2))/(PI**2)
      ALPHA0 = EX
      BETA0 = SQRT(BBETA)
      RETURN
      END
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

โปรแกรมในการประมาณค่าพารามิเตอร์ของการแจกแจงความเสียหายแบบปกติ เมื่อข้อมูลความเสียหายถูกตัดปลายทั้งสองข้าง ด้วยวิธีการประมาณค่า 3 วิธีคือ ML, MD และ MWD ดังนี้

```
C**********************
C   MAIN PROGRAM-1
C**********************
      COMMON/SEED/IX,KK
     *      /PAR1/AMEW,SIGMA,XMEW0,XSIG0
     *      /FACT/XD,XM,N
     *      /SAMP/X(500)
     *      /SIM/FQ(500),F(500)
     *      /SUMM/SUMMW(3),SUMSG(3),SUMMW1(3),SUMSG1(3),
     *           SUMMW2(3),SUMSG2(3)
     *      /VAR1/VARMW(3),VARSG(3),BIASMW(3),BIASSG(3),
     *           VMSEMW(3),VMSESG(3),BIASM(3),BIASS(3)
     *      /VAR2/AMSE(3),RMSE(3),RELMW(3),RELSG(3),REL(3)
     *      /INDEX/DIFG,NOTG
      CHARACTER DIFG*1,NOTG*1
      N=70
      XD=0.2
      XM=15.0
      PC=0.2
      AMEW=10.0
      SIGMA=2.0
      IX=103
      IR=1000
      KK=0
      DO 710 IKK=1,3
         SUMMW(IKK) = 0.0
         SUMSG(IKK) = 0.0
         SUMMW1(IKK) = 0.0
         SUMSG1(IKK) = 0.0
         SUMMW2(IKK) = 0.0
```

```
      SUMSG2(IKK) = 0.0
710   CONTINUE
      WRITE(6,25)
      WRITE(6,515) AMEW,SIGMA
      WRITE(6,51) N,PC,XD,XM
      DO 100 IZ=1,IR
         NUM1=0
         NUM2=0
         NPC=INT(N*PC)
         NN=N-NPC
         IDIFG=0
1        DO 15 I=1,N
5         CALL NORMAL(AMEW,SIGMA,ANORM)
          ANOR=ANORM
          IF (ANOR.LE.XD) GOTO 5
             IF (ANOR.LE.XM) THEN
                IF (NUM2.GE.NN) GOTO 5
                   NUM2=NUM2+1
                   X(I)=ANOR
                ELSE
                  IF (NUM1.GE.NPC) GOTO 5
                     NUM1=NUM1+1
                     X(I)=XM
                END IF
15    CONTINUE
C**********************WRITE HEAD***************
      CALL SORT(N)
      CALL CDFEMP(N)
      CALL MOMENT(N,XMEW0,XSIG0)
      IF ((XMEW0.LE.0).OR.(XSIG0.LE.0)) THEN
         NUM1=0
         NUM2=0
         GOTO 1
      END IF
      MAX=50
      DIFG='N'
      NOTG='N'
```

```
      CALL MLE(MAX,XMEW1,XSIG1)
      IF ((DIFG.EQ.'Y').OR.(NOTG.EQ.'Y')) THEN
          NUM1=0
          NUM2=0
          GOTO 1
       END IF
      CALL MDE(MAX,XMEW2,XSIG2)
      IF (DIFG.EQ.'Y').OR.(NOTG.EQ.'Y')) THEN
          NUM1=0
          NUM2=0
          GOTO 1
      END IF
      CALL MWD(MAX,XMEW3,XSIG3)
      IF ((DIFG.EQ.'Y').OR.(NOTG.EQ.'Y')) THEN
          NUM1=0
          NUM2=0
          GOTO 1
      END IF
      SUMMW(1) = SUMMW(1)+XMEW1
      SUMMW1(1) = SUMMW1(1)+(XMEW1**2)
      SUMSG(1) = SUMSG(1)+XSIG1
      SUMSG1(1) = SUMSG1(1)+(XSIG1**2)
      DIFER1 = ABS(XMEW1-AMEW)
      DIFER2 = ABS(XSIG1-SIGMA)
      SUMMW2(1) = SUMMW2(1)+DIFER1
      SUMSG2(1) = SUMSG2(1)+DIFER2
C*****************************************
      SUMMW(2) = SUMMW(2)+XMEW2
      SUMMW1(2) = SUMMW1(2)+(XMEW2**2)
      SUMSG(2) = SUMSG(2)+XSIG2
      SUMSG1(2) = SUMSG1(2)+(XSIG2**2)
      DIFER3 = ABS(XMEW2-AMEW)
      DIFER4 = ABS(XSIG2-SIGMA)
      SUMMW2(2) = SUMMW2(2)+DIFER3
      SUMSG2(2) = SUMSG2(2)+DIFER4
C*****************************************
      SUMMW(3) = SUMMW(3)+XMEW3
```

```
      SUMMW1(3) = SUMMW1(3)+(XMEW3**2)
      SUMSG(3) = SUMSG(3)+XSIG3
      SUMSG1(3) = SUMSG1(3)+(XSIG3**2)
      DIFER5 = ABS(XMEW3-AMEW)
      DIFER6 = ABS(XSIG3-SIGMA)
      SUMMW2(3) = SUMMW2(3)+DIFER5
      SUMSG2(3) = SUMSG2(3)+DIFER6
C*****************************************
100   CONTINUE
      DO 711 IJ=1,3
       VARMW(IJ) = (SUMMW1(IJ)-(SUMMW(IJ)**2)/IR)/IR
       VARSG(IJ) = (SUMSG1(IJ)-(SUMSG(IJ)**2)/IR)/IR
       BIASM(IJ) = SUMMW(IJ)/IR-AMEW
       BIASMW(IJ) = BIASM(IJ)**2
       BIASS(IJ) = SUMSG(IJ)/IR-SIGMA
      BIASSG(IJ) = BIASS(IJ)**2
       VMSEMW(IJ) = VARMW(IJ)+BIASMW(IJ)
       VMSESG(IJ) = VARSG(IJ)+BIASSG(IJ)
       AMSE(IJ) = (VMSEMW(IJ)+VMSESG(IJ))/2
       RMSE(IJ) = SQRT(AMSE(IJ))
       RELMW(IJ) = SUMMW2(IJ)/AMEW
       RELSG(IJ) = SUMSG2(IJ)/SIGMA
       REL(IJ) = (RELMW(IJ)+RELSG(IJ))/(2*IR)
711   CONTINUE
C*****WRITE OUTPUT OF NORMAL DIST.**********
      WRITE(6,52)
      WRITE(6,65) VARMW(1),VARSG(1),BIASM(1),BIASS(1)
     *   ,VMSEMW(1),VMSESG(1),AMSE(1),RMSE(1)
      WRITE(6,655) SUMMW2(1),SUMSG2(1),RELMW(1)
     *   ,RELSG(1),REL(1)
C*****************************************************
      WRITE(6,521)
      WRITE(6,65) VARMW(2),VARSG(2),BIASM(2),BIASS(2)
     *   ,VMSEMW(2),VMSESG(2),AMSE(2),RMSE(2)
      WRITE(6,655) SUMMW2(2),SUMSG2(2),RELMW(2)
     *   ,RELSG(2),REL(2)
C*****************************************************
```

```
      WRITE(6,522)
      WRITE(6,65) VARMW(3),VARSG(3),BIASM(3),BIASS(3)
     *    ,VMSEMW(3),VMSESG(3),AMSE(3),RMSE(3)
      WRITE(6,655) SUMMW2(3),SUMSG2(3),RELMW(3)
     *    ,RELSG(3),REL(3)
C*****************************************************
25    FORMAT(14X,'** NORMAL DIST.**')
51    FORMAT(7X,'SAMPLE SIZE=',I4,2X,'PC.=',F4.2,2X,'DEDUC.=',F4.1
     *       ,2X,'MAX=',F5.1)
515   FORMAT(7X,'PAR.MEW ',F9.6,2X,'SIGMA ',F9.6)
52    FORMAT(10X,'**MAXIMUM LIKELIHOOD**')
521   FORMAT(11X,'**MINIMUM DISTANCE**')
522   FORMAT(8X,'**MINIMUM WEIGHT DISTANCE**')
65    FORMAT(5X,'VAR(MEW) =',E15.6,3X,'VAR(SIGMA) =',E15.6/
     * 5X,'BIAS(MEW)=',E15.6,3X,'BIAS(SIGMA)=',E15.6/
     * 5X,'MSE(MEW)=',E15.6,3X,'MSE(SIGMA)=',E15.6/
     * 5X,'MSE=',E15.6,3X,'RMSE=',E15.6)
655   FORMAT(5X,'ABS(MEW)=',E15.6,3X,'ABS(SIGMA)=',E15.6/
     *  5X,'REL(MEW)=',E15.6,3X,'REL(SIGMA)=',E15.6/
     *  5X,'REL=',E15.6)
200   STOP
      END
C**********************
C     RANDOM NUMBER
C**********************
      FUNCTION RAND(IX)
      IX=IX*16807
      IF (IX) 1,2,2
1     IX=IX+2147483647+1
2     RAND=IX
      RAND=RAND/2147483647
      RETURN
      END
C*********************************
C     NORMAL DISTRIBUTION
C*********************************
      SUBROUTINE NORMAL(AMEW,SIGMA,ANORM)
```

```
      COMMON/SEED/IX,KK
      PI=3.1415927
      IF (KK.EQ.1) GOTO 10
      RONE=RAND(IX)
      RTWO=RAND(IX)
      ZONE=SQRT(-2*ALOG(RONE))*COS(2*PI*RTWO)
      ZTWO=SQRT(-2*ALOG(RONE))*SIN(2*PI*RTWO)
      ANORM=ZONE*SIGMA+AMEW
      KK=1
      GOTO 20
10    ANORM=ZTWO*SIGMA+AMEW
      KK=0
20    RETURN
      END
C*************************
C     SHELL SORT
C*************************
      SUBROUTINE SORT(N)
      COMMON/SAMP/X(500)
      JUMP=N
10    JUMP=JUMP/2
      IF (JUMP.EQ.0) GOTO 90
        J2=N-JUMP
        DO 30 J=1,J2
          I=J
20        J3=I+JUMP
          IF (X(I).LE.X(J3)) GOTO 30
            HOLD=X(I)
            X(I)=X(J3)
            X(J3)=HOLD
            I=I-JUMP
              IF (I.GT.0) GOTO 20
30      CONTINUE
      GOTO 10
90    RETURN
      END
C*************************
```

```
C     CDF.OF EMPIRICAL
C**************************
      SUBROUTINE CDFEMP(N)
      COMMON/SAMP/X(500)
     *      /SIM/FQ(500),F(500)
      NUM = 1
5     FQ(NUM) = 1
      DO 10 I=NUM,N
         J = I+1
        IF (J.GT.N) GOTO 20
           IF (X(I).EQ.X(J)) THEN
               FQ(NUM) = FQ(NUM)+1
           ELSE
               GOTO 4
           END IF
10    CONTINUE
4     NUM = J
      GOTO 5
20    M = 1
      JUMP = 1
30    K = JUMP
       IF (JUMP.GT.N) GOTO 40
       L = JUMP-M
      IF (L.EQ.0) THEN
         F(K) = FQ(K)/N+L
      ELSE
        F(K) = FQ(K)/N+F(L)
      END IF
      M = FQ(K)
      JUMP = JUMP+M
      DO 50 I = 1,N
         IF ((I.GE.K).AND.(I.LT.JUMP)) THEN
               F(I) = F(K)
               FQ(I) = FQ(K)
         END IF
50    CONTINUE
      GOTO 30
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
40    RETURN
      END
C*********************************************
C     INITIAL VALUE OF MEW AND SIGMA BY USE
C       METHOD OF MOMENT FOR NORMAL DIST.
C*********************************************
      SUBROUTINE MOMENT(N,XMEW0,XSIG0)
      COMMON/SAMP/X(500)
      SUMX=0.0
      SUMX2=0.0
      DO 10 I=1,N
          SUMX=SUMX+X(I)
          SUMX2=SUMX2+(X(I)**2)
10    CONTINUE
      EX = SUMX/N
      VARX = (SUMX2-(SUMX**2/N))/(N-1)
      XMEW0 = EX
      XSIG0 = SQRT(VARX)
      RETURN
      END
C*********************************************
C       ESTIMATION OF THE PARAMETERS BY
C        METHOD OF MAXIMUM LIKELIHOOD
C             FOR NORMAL DIST.
C*********************************************
      SUBROUTINE MLE(MAX,XMEW1,XSIG1)
      COMMON/PAR1/AMEW,SIGMA,XMEW0,XSIG0
     *      /FACT/XD,XM,N
     *      /SAMP/X(500)
     *      /INDEX/DIFG,NOTG
      CHARACTER DIFG*1,NOTG*1
      P0 = XMEW0
      R0 = XSIG0
      IC = 0
      DO 20 J=1,MAX
          IF ((P0.LE.0).OR.(R0.LE.0)) THEN
             DIFG = 'Y'
```

```
          GOTO 90
        END IF
      CALL COMPG1(P0,R0,BEG1,BEG2,BEG11,BEG12,BEG22)
          IF (DIFG.EQ.'Y') GOTO 90
        G1=0.0
        G2=0.0
        G11=0.0
        G12=0.0
        G21=0.0
        G22=0.0
        SDIF1=0.0
        SDIF2=0.0
        SDIF11=0.0
        SDIF12=0.0
        SDIF21=0.0
        SDIF22=0.0
           IC=IC+1
         DO 10 I=1,N
           ZX = (X(I)-P0)/R0
           DIF1 = ZX/R0
      DIF2 = (ZX**2-1)/R0
           DIF12 = (-2*ZX)/(R0**2)
           DIF22 = (-1*(3*(ZX**2)-1))/(R0**2)
           SDIF1 = SDIF1+DIF1
           SDIF2 = SDIF2+DIF2
           SDIF12 = SDIF12+DIF12
           SDIF22 = SDIF22+DIF22
10         CONTINUE
          SDIF11 = -1*N/(R0**2)
           SDIF21 = SDIF12
              G1 = SDIF1+BEG1
              G2 = SDIF2+BEG2
              G11 = SDIF11+BEG11
              G12 = SDIF12+BEG12
              G21 = G12
              G22 = SDIF22+BEG22
             DET = (G11*G22)-(G12*G21)
```

```
                G21 = G12
                G22 = SDIF22+BEG22
               DET = (G11*G22)-(G12*G21)
                 IF (DET.EQ.0) THEN
                    DIFG = 'Y'
                    GOTO 90
                 END IF
              PP = (G12*G2)-(G22*G1)
              RP = (G21*G1)-(G11*G2)
              P1 = PP/DET+P0
              R1 = RP/DET+R0
      IF ((ABS(P1-P0).LT.0.001).AND.(ABS(R1-R0).LT.0.001)) THEN
               GOTO 80
      ELSE
               P0 = P1
               R0 = R1
      END IF
               IF (IC.GE.MAX) THEN
                  NOTG = 'Y'
                  GOTO 90
             END IF
20    CONTINUE
80    XMEW1 = P1
     XSIG1 = R1
90    RETURN
     END
C*****************************
C     COMPUTING VALUE G
C*****************************
     SUBROUTINE COMPG1(P0,R0,BEG1,BEG2,BEG11,BEG12,BEG22)
     COMMON/FACT/XD,XM,N
    *     /INDEX/DIFG,NOTG
     CHARACTER DIFG*1,NOTG*1
      ZM = (XM-P0)/R0
      ZD = (XD-P0)/R0
      ZC1 = CDFN(ZM)
```

```
      ZC2 = CDFN(ZD)
      ZC = ZC1-ZC2
       IF (ZC.EQ.0) THEN
          DIFG = 'Y'
          GOTO 99
       END IF
      ZP1 = PDFN(ZM)
      ZP2 = PDFN(ZD)
      PG1 = ZP1-ZP2
      PG2 = ZM*ZP1-ZD*ZP2
      PG3 = R0*ZC
      PG4 = (R0**2)*ZC
       BEG1 = N*PG1/PG3
       BEG2 = N*PG2/PG3
       BEG11 = (PG2+(PG1**2)/ZC)*N/PG4
       BBEG12 = (ZM**2-1)*ZP1-(ZD**2-1)*ZP2
       BEG12 = N*(BBEG12+PG1*PG2/ZC)/PG4
       BBEG22 = ZM*(ZM**2-2)*ZP1-ZD*(ZD**2-2)*ZP2
       BEG22 = N*(BBEG22+PG2**2/ZC)/PG4
99    RETURN
      END
C*********************************************
C     ESTIMATION OF THE PARAMETERS BY
C       METHOD OF MINIMUM  DISTANCE
C          FOR NORMAL DIST.
C*********************************************
      SUBROUTINE MDE(MAX,XMEW2,XSIG2)
      COMMON/PAR1/AMEW,SIGMA,XMEW0,XSIG0
     *    /FACT/XD,XM,N
     *    /SAMP/X(500)
     *    /SIM/FQ(500),F(500)
     *    /INDEX/DIFG,NOTG
      CHARACTER DIFG*1,NOTG*1
      P10 = XMEW0
      R10 = XSIG0
      IC1 = 0
```

```
DO 20 J1=1,MAX
    IF ((P10.LE.0).OR.(R10.LE.0)) THEN
       DIFG = 'Y'
       GOTO 90
    END IF
     GG1=0.0
     GG2=0.0
     GG11=0.0
     GG12=0.0
     GG21=0.0
     GG22=0.0
     SDF1=0.0
     SDF2=0.0
     SDF11=0.0
     SDF12=0.0
     SDF21=0.0
     SDF22=0.0
        IC1=IC1+1
 Z1M = (XM-P10)/R10
 Z1D = (XD-P10)/R10
 Z1C1 = CDFN(Z1M)
 Z1C2 = CDFN(Z1D)
 Z1C = Z1C1-Z1C2
   IF (Z1C.EQ.0) THEN
      DIFG = 'Y'
      GOTO 90
   END IF
 Z1P1 = PDFN(Z1M)
 Z1P2 = PDFN(Z1D)
P1G1 = Z1P2-Z1P1
P1G2 = Z1D*Z1P2-Z1M*Z1P1
P1G3 = (Z1D**2)*Z1P2-(Z1M**2)*Z1P1
P1G4 = (Z1D**3)*Z1P2-(Z1M**3)*Z1P1
P1G55 = R10*(Z1C**2)
P1G56 = ((R10*Z1C)**2)*Z1C
    DO 10 IM=1,N
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
      P1X1 = F(IM)
          IF (P1X1.EQ.0) THEN
              DIFG = 'Y'
              GOTO 90
          END IF
      Z1X = (X(IM)-P10)/R10
      Z1C3 = CDFN(Z1X)
      Z1P3 = PDFN(Z1X)
      P1G5 = Z1P2-Z1P3
      P1G6 = Z1C3-Z1C2
      F1YX = P1G6/Z1C
      F1Y = F1YX-P1X1
      P1G7 = Z1D*Z1P2-Z1X*Z1P3
      P1G8 = (Z1D**2)*Z1P2-(Z1X**2)*Z1P3
      P1G9 = (Z1D**3)*Z1P2-(Z1X**3)*Z1P3
     DFM = (Z1C*P1G5-P1G6*P1G1)/P1G55
     DFS = (Z1C*P1G7-P1G6*P1G2)/P1G55
     DFMM = (Z1C*(Z1C*P1G7-P1G6*P1G2)-2*P1G1
    *       *(Z1C*P1G5-P1G6*P1G1))/P1G56
     DFMS = (Z1C*(Z1C*(P1G8-P1G5)+P1G1*P1G7
    *       -P1G6*(P1G3-P1G1)-P1G5*P1G2)
    *       -2*P1G1*(Z1C*P1G7-P1G6*P1G2))/P1G56
     DFSS = (Z1C*(Z1C*(P1G9-P1G7)-P1G6*(P1G4-P1G2))
    *      -(2*P1G2+Z1C)*(Z1C*P1G7-P1G6*P1G2))/P1G56
      H1G1 = 2*F1Y*DFM
      H1G2 = 2*F1Y*DFS
      H1G11 = 2*(F1Y*DFMM+DFM**2)
      H1G21 = 2*(F1Y*DFMS+DFM*DFS)
      H1G22 = 2*(F1Y*DFSS+DFS**2)
      SDF1 = SDF1+H1G1
      SDF2 = SDF2+H1G2
      SDF11 = SDF11+H1G11
      SDF21 = SDF21+H1G21
      SDF22 = SDF22+H1G22
10    CONTINUE
       GG1 = SDF1
```

```
      GG2 = SDF2
      GG11 = SDF11
      GG21 = SDF21
      GG12 = GG21
      GG22 = SDF22
      DET1 = (GG11*GG22)-(GG12*GG21)
        IF (DET1.EQ.0) THEN
          DIFG = 'Y'
          GOTO 90
        END IF
      PP1 = (GG12*GG2)-(GG22*GG1)
      RP1 = (GG21*GG1)-(GG11*GG2)
      P11 = PP1/DET1+P10
      R11 = RP1/DET1+R10
    IF ((ABS(P11-P10).LT.0.001).AND.(ABS(R11-R10).LT.0.001)) THEN
        GOTO 80
      ELSE
          P10 = P11
          R10 = R11
    END IF
      IF (IC1.GE.MAX) THEN
          NOTG = 'Y'
          GOTO 90
      END IF
20    CONTINUE
80   XMEW2 = P11
    XSIG2 = R11
90   RETURN
    END
C*********************************************
C     ESTIMATION OF THE PARAMETERS BY
C     METHOD OF MINIMUM WEIGHT DISTANCE
C          FOR NORMAL DIST.
C*********************************************
    SUBROUTINE MWD(MAX,XMEW3,XSIG3)
    COMMON/PAR1/AMEW,SIGMA,XMEW0,XSIG0
```

```
*     /FACT/XD,XM,N
*     /SAMP/X(500)
*     /SIM/FQ(500),F(500)
*     /INDEX/DIFG,NOTG
CHARACTER DIFG*1,NOTG*1
P20 = XMEW0
R20 = XSIG0
IC2 = 0
      DO 20 J2=1,MAX
     IF ((P20.LE.0).OR.(R20.LE.0)) THEN
        DIFG = 'Y'
        GOTO 90
     END IF
      GGG1=0.0
      GGG2=0.0
      GGG11=0.0
      GGG12=0.0
      GGG21=0.0
      GGG22=0.0
      SDI1=0.0
      SDI2=0.0
      SDI11=0.0
      SDI12=0.0
      SDI21=0.0
      SDI22=0.0
         IC2=IC2+1
Z2M = (XM-P20)/R20
Z2D = (XD-P20)/R20
Z2C1 = CDFN(Z2M)
Z2C2 = CDFN(Z2D)
Z2C = Z2C1-Z2C2
  IF (Z2C.EQ.0) THEN
     DIFG = 'Y'
     GOTO 90
  END IF
Z2P1 = PDFN(Z2M)
```

```
      Z2P2 = PDFN(Z2D)
      P2G1 = Z2P2-Z2P1
      P2G2 = Z2D*Z2P2-Z2M*Z2P1
      P2G3 = (Z2D**2)*Z2P2-(Z2M**2)*Z2P1
      P2G4 = (Z2D**3)*Z2P2-(Z2M**3)*Z2P1
      P2G55 = R20*(Z2C**2)
      P2G56 = ((R20*Z2C)**2)*Z2C
          DO 10 IMM=1,N
            P2X1 = F(IMM)/(N+1)
            P2X11 = N*P2X1
            P2X12 = 1.0-P2X11
             IF ((P2X11.EQ.0).OR.(P2X12.EQ.0)) THEN
                 DIFG = 'Y'
                 GOTO 90
             END IF
            P2X22 = P2X1*P2X12
       Z2X = (X(IMM)-P20)/R20
       Z2C3 = CDFN(Z2X)
       Z2P3 = PDFN(Z2X)
       P2G5 = Z2P2-Z2P3
       P2G6 = Z2C3-Z2C2
       F2YX = P2G6/Z2C
       F2Y = F2YX-P2X11
       P2G7 = Z2D*Z2P2-Z2X*Z2P3
       P2G8 = (Z2D**2)*Z2P2-(Z2X**2)*Z2P3
       P2G9 = (Z2D**3)*Z2P2-(Z2X**3)*Z2P3
      DIM = (Z2C*P2G5-P2G6*P2G1)/P2G55
      DIS = (Z2C*P2G7-P2G6*P2G2)/P2G55
      DIMM = (Z2C*(Z2C*P2G7-P2G6*P2G2)
     *      -2*P2G1*(Z2C*P2G5-P2G6*P2G1))/P2G56
      DIMS = (Z2C*(Z2C*(P2G8-P2G5)+P2G1*P2G7
     *       -P2G6*(P2G3-P2G1)-P2G5*P2G2)
     *       -2*P2G1*(Z2C*P2G7-P2G6*P2G2))/P2G56
      DISS = (Z2C*(Z2C*(P2G9-P2G7)-P2G6*(P2G4-P2G2))-(2*P2G2+Z2C)
     *       *(Z2C*P2G7-P2G6*P2G2))/P2G56
       H2G1 = 2*F2Y*DIM/P2X22
```

```
      H2G2 = 2*F2Y*DIS/P2X22
      H2G11 = 2*(F2Y*DIMM+DIM**2)/P2X22
      H2G21 = 2*(F2Y*DIMS+DIM*DIS)/P2X22
      H2G22 = 2*(F2Y*DISS+DIS**2)/P2X22
       SDI1 = SDI1+H2G1
       SDI2 = SDI2+H2G2
       SDI11 = SDI11+H2G11
       SDI21 = SDI21+H2G21
       SDI22 = SDI22+H2G22
10     CONTINUE
        GGG1 = SDI1
        GGG2 = SDI2
        GGG11 = SDI11
        GGG21 = SDI21
        GGG12 = GGG21
        GGG22 = SDI22
        DET2 = (GGG11*GGG22)-(GGG12*GGG21)
          IF (DET2.EQ.0) THEN
             DIFG = 'Y'
             GOTO 90
          END IF
        PP2 = (GGG12*GGG2)-(GGG22*GGG1)
        RP2 = (GGG21*GGG1)-(GGG11*GGG2)
        P21 = PP2/DET2+P20
        R21 = RP2/DET2+R20
     IF ((ABS(P21-P20).LT.0.001).AND.(ABS(R21-R20).LT.0.001)) THEN
                GOTO 80
       ELSE
                P20 = P21
                R20 = R21
       END IF
          IF (IC2.GE.MAX) THEN
                NOTG = 'Y'
                GOTO 90
          END IF
20     CONTINUE
80    XMEW3 = P21
```

```
      XSIG3 = R21
90    RETURN
      END
C*************************************
C     PDF. OF Z, Z=(X-MEW)/SIGMA
C*************************************
      FUNCTION PDFN(Z)
      PI=3.141593
      IF ((Z.GT.7.0).OR.(Z.LT.-7.0)) THEN
         PDFN=0.0
      ELSE
         XX=-1*(Z**2)/2
         PDFN=EXP(XX)/SQRT(2*PI)
      END IF
      RETURN
      END
C*************************************
C     CDF. OF Z, Z=(X-MEW)/SIGMA
C*************************************
      FUNCTION CDFN(ZZ)
      IF ((ZZ.GT.7.0).OR.(ZZ.LT.-7.0)) THEN
         P=1.0
      ELSE
       Z1 = ABS(ZZ)
       T9 = 1/(1.0+0.33267*Z1)
       R = (EXP(-1.0*(ZZ**2)/2))/2.5066282746
       P = 1.0-(T9*(0.4361836-0.1201676*T9+0.937298*(T9**2)))*R
      END IF
      IF (ZZ.GE.0) THEN
         CDFN=P
      ELSE
         CDFN=1.0-P
      END IF
      RETURN
      END
```

โปรแกรมย่อยวิธีการประมาณค่าพารามิเตอร์ 3 วิธีคือ ML, MD และ MWD ของการแจกแจงความเสียหายแบบลอกนอร์มอล ดังนี้

```
C*********************************************
C      ESTIMATION OF THE PARAMETERS BY
C       METHOD OF MAXIMUM LIKELIHOOD
C           FOR LOGNORMAL DIST.
C*********************************************
      SUBROUTINE MLE(MAX,XMEW1,XSIG1)
      COMMON/PAR1/AMEW,SIGMA,XMEW0,XSIG0
     *     /FACT/XD,XM,N
     *     /SAMP/X(500)
     *     /INDEX/DIFG,NOTG
      CHARACTER DIFG*1,NOTG*1
      P0 = XMEW0
      R0 = XSIG0
      IC = 0
      DO 20 J=1,MAX
         IF (R0.LE.0) THEN
            DIFG = 'Y'
            GOTO 90
         END IF
       CALL COMPG1(P0,R0,BEG1,BEG2,BEG11,BEG12,BEG22)
            IF (DIFG.EQ.'Y') GOTO 90
          G1=0.0
          G2=0.0
          G11=0.0
          G12=0.0
          G21=0.0
          G22=0.0
          SDIF1=0.0
          SDIF2=0.0
          SDIF11=0.0
          SDIF12=0.0
          SDIF21=0.0
         SDIF22=0.0
            IC=IC+1
```

```
      DO 10 I=1,N
        ZX = (ALOG(X(I))-P0)/R0
        DIF1 = ZX/R0
        DIF2 = (ZX**2-1)/R0
        DIF12 = (-2*ZX)/(R0**2)
        DIF22 = (-1*(3*(ZX**2)-1))/(R0**2)
        SDIF1 = SDIF1+DIF1
        SDIF2 = SDIF2+DIF2
        SDIF12 = SDIF12+DIF12
        SDIF22 = SDIF22+DIF22
10      CONTINUE
        SDIF11 = -1*N/(R0**2)
        SDIF21 = SDIF12
          G1 = SDIF1+BEG1
          G2 = SDIF2+BEG2
          G11 = SDIF11+BEG11
          G12 = SDIF12+BEG12
          G21 = G12
          G22 = SDIF22+BEG22
         DET = (G11*G22)-(G12*G21)
           IF (DET.EQ.0) THEN
             DIFG = 'Y'
             GOTO 90
           END IF
        PP = (G12*G2)-(G22*G1)
        RP = (G21*G1)-(G11*G2)
        P1 = PP/DET+P0
        R1 = RP/DET+R0
 IF ((ABS(P1-P0).LT.0.001).AND.(ABS(R1-R0).LT.0.001)) THEN
        GOTO 80
 ELSE
        P0 = P1
        R0 = R1
 END IF
        IF (IC.GE.MAX) THEN
          NOTG = 'Y'
```

```
                GOTO 90
              END IF
20    CONTINUE
80    XMEW1 = P1
      XSIG1 = R1
90    RETURN
      END
C*****************************
C     COMPUTING VALUE G
C*****************************
      SUBROUTINE COMPG1(P0,R0,BEG1,BEG2,BEG11,BEG12,BEG22)
      COMMON/FACT/XD,XM,N
     *     /INDEX/DIFG,NOTG
      CHARACTER DIFG*1,NOTG*1
       ZM = (ALOG(XM)-P0)/R0
       ZD = (ALOG(XD)-P0)/R0
       ZC1 = CDFN(ZM)
       ZC2 = CDFN(ZD)
       ZC = ZC1-ZC2
        IF (ZC.EQ.0) THEN
           DIFG = 'Y'
           GOTO 99
        END IF
       ZP1 = PDFN(ZM)
       ZP2 = PDFN(ZD)
       PG1 = ZP1-ZP2
       PG2 = ZM*ZP1-ZD*ZP2
       PG3 = R0*ZC
       PG4 = (R0**2)*ZC
        BEG1 = N*PG1/PG3
        BEG2 = N*PG2/PG3
       BEG11 = (PG2+(PG1**2)/ZC)*N/PG4
        BBEG12 = (ZM**2-1)*ZP1-(ZD**2-1)*ZP2
        BEG12 = N*(BBEG12+PG1*PG2/ZC)/PG4
        BBEG22 = ZM*(ZM**2-2)*ZP1-ZD*(ZD**2-2)*ZP2
        BEG22 = N*(BBEG22+PG2**2/ZC)/PG4
```

```
99    RETURN
      END
C*********************************************
C       ESTIMATION OF THE PARAMETERS BY
C         METHOD OF MINIMUM  DISTANCE
C            FOR LOGNORMAL DIST.
C*********************************************
      SUBROUTINE MDE(MAX,XMEW2,XSIG2)
      COMMON/PAR1/AMEW,SIGMA,XMEW0,XSIG0
     *      /FACT/XD,XM,N
     *      /SAMP/X(500)
     *      /SIM/FQ(500),F(500)
     *      /INDEX/DIFG,NOTG
      CHARACTER DIFG*1,NOTG*1
      P10 = XMEW0
      R10 = XSIG0
      IC1 = 0
      DO 20 J1=1,MAX
         IF (R10.LE.0) THEN
            DIFG = 'Y'
            GOTO 90
         END IF
         GG1=0.0
         GG2=0.0
         GG11=0.0
         GG12=0.0
         GG21=0.0
         GG22=0.0
         SDF1=0.0
         SDF2=0.0
         SDF11=0.0
         SDF12=0.0
         SDF21=0.0
         SDF22=0.0
            IC1=IC1+1
      Z1M = (ALOG(XM)-P10)/R10
```

```
      Z1D = (ALOG(XD)-P10)/R10
      Z1C1 = CDFN(Z1M)
      Z1C2 = CDFN(Z1D)
      Z1C = Z1C1-Z1C2
        IF (Z1C.EQ.0) THEN
           DIFG = 'Y'
           GOTO 90
        END IF
      Z1P1 = PDFN(Z1M)
      Z1P2 = PDFN(Z1D)
     P1G1 = Z1P2-Z1P1
     P1G2 = Z1D*Z1P2-Z1M*Z1P1
     P1G3 = (Z1D**2)*Z1P2-(Z1M**2)*Z1P1
     P1G4 = (Z1D**3)*Z1P2-(Z1M**3)*Z1P1
     P1G55 = R10*(Z1C**2)
     P1G56 = ((R10*Z1C)**2)*Z1C
         DO 10 IM=1,N
         P1X1 = F(IM)
            IF (P1X1.EQ.0) THEN
               DIFG = 'Y'
               GOTO 90
            END IF
       Z1X = (ALOG(X(IM))-P10)/R10
       Z1C3 = CDFN(Z1X)
       Z1P3 = PDFN(Z1X)
       P1G5 = Z1P2-Z1P3
      P1G6 = Z1C3-Z1C2
       F1YX = P1G6/Z1C
        F1Y = F1YX-P1X1
       P1G7 = Z1D*Z1P2-Z1X*Z1P3
       P1G8 = (Z1D**2)*Z1P2-(Z1X**2)*Z1P3
       P1G9 = (Z1D**3)*Z1P2-(Z1X**3)*Z1P3
     DFM = (Z1C*P1G5-P1G6*P1G1)/P1G55
     DFS = (Z1C*P1G7-P1G6*P1G2)/P1G55
     DFMM = (Z1C*(Z1C*P1G7-P1G6*P1G2)-2*P1G1
    *        *(Z1C*P1G5-P1G6*P1G1))/P1G56
```

```
      DFMS = (Z1C*(Z1C*(P1G8-P1G5)+P1G1*P1G7
     *       -P1G6*(P1G3-P1G1)-P1G5*P1G2)
     *       -2*P1G1*(Z1C*P1G7-P1G6*P1G2))/P1G56
      DFSS = (Z1C*(Z1C*(P1G9-P1G7)-P1G6*(P1G4-P1G2))
     *       -(2*P1G2+Z1C)*(Z1C*P1G7-P1G6*P1G2))/P1G56
       H1G1 = 2*F1Y*DFM
       H1G2 = 2*F1Y*DFS
       H1G11 = 2*(F1Y*DFMM+DFM**2)
       H1G21 = 2*(F1Y*DFMS+DFM*DFS)
       H1G22 = 2*(F1Y*DFSS+DFS**2)
        SDF1 = SDF1+H1G1
        SDF2 = SDF2+H1G2
        SDF11 = SDF11+H1G11
        SDF21 = SDF21+H1G21
        SDF22 = SDF22+H1G22
10     CONTINUE
         GG1 = SDF1
         GG2 = SDF2
         GG11 = SDF11
         GG21 = SDF21
         GG12 = GG21
         GG22 = SDF22
         DET1 = (GG11*GG22)-(GG12*GG21)
           IF (DET1.EQ.0) THEN
              DIFG = 'Y'
              GOTO 90
           END IF
         PP1 = (GG12*GG2)-(GG22*GG1)
         RP1 = (GG21*GG1)-(GG11*GG2)
         P11 = PP1/DET1+P10
         R11 = RP1/DET1+R10
      IF ((ABS(P11-P10).LT.0.001).AND.(ABS(R11-R10).LT.0.001)) THEN
                 GOTO 80
        ELSE
                 P10 = P11
                 R10 = R11
```

```
      END IF
          IF (IC1.GE.MAX) THEN
              NOTG = 'Y'
              GOTO 90
          END IF
20     CONTINUE
80    XMEW2 = P11
      XSIG2 = R11
90    RETURN
      END
C*********************************************
C       ESTIMATION OF THE PARAMETERS BY
C       METHOD OF MINIMUM WEIGHT DISTANCE
C            FOR LOGNORMAL DIST.
C*********************************************
      SUBROUTINE MWD(MAX,XMEW3,XSIG3)
      COMMON/PAR1/AMEW,SIGMA,XMEW0,XSIG0
     *     /FACT/XD,XM,N
     *     /SAMP/X(500)
     *     /SIM/FQ(500),F(500)
     *     /INDEX/DIFG,NOTG
      CHARACTER DIFG*1,NOTG*1
      P20 = XMEW0
      R20 = XSIG0
      IC2 = 0
      DO 20 J2=1,MAX
          IF (R20.LE.0) THEN
             DIFG = 'Y'
             GOTO 90
          END IF
          GGG1=0.0
          GGG2=0.0
          GGG11=0.0
          GGG12=0.0
          GGG21=0.0
          GGG22=0.0
```

```
      SDI1=0.0
      SDI2=0.0
      SDI11=0.0
      SDI12=0.0
      SDI21=0.0
      SDI22=0.0
         IC2=IC2+1
 Z2M = (ALOG(XM)-P20)/R20
 Z2D = (ALOG(XD)-P20)/R20
 Z2C1 = CDFN(Z2M)
 Z2C2 = CDFN(Z2D)
 Z2C = Z2C1-Z2C2
  IF (Z2C.EQ.0) THEN
     DIFG = 'Y'
     GOTO 90
  END IF
 Z2P1 = PDFN(Z2M)
 Z2P2 = PDFN(Z2D)
P2G1 = Z2P2-Z2P1
P2G2 = Z2D*Z2P2-Z2M*Z2P1
P2G3 = (Z2D**2)*Z2P2-(Z2M**2)*Z2P1
P2G4 = (Z2D**3)*Z2P2-(Z2M**3)*Z2P1
P2G55 = R20*(Z2C**2)
P2G56 = ((R20*Z2C)**2)*Z2C
    DO 10 IMM=1,N
      P2X1 = F(IMM)/(N+1)
      P2X11 = N*P2X1
      P2X12 = 1.0-P2X11
       IF ((P2X11.EQ.0).OR.(P2X12.EQ.0)) THEN
          DIFG = 'Y'
          GOTO 90
       END IF
      P2X22 = P2X1*P2X12
 Z2X = (ALOG(X(IMM))-P20)/R20
 Z2C3 = CDFN(Z2X)
 Z2P3 = PDFN(Z2X)
```

```
      P2G5 = Z2P2-Z2P3
      P2G6 = Z2C3-Z2C2
      F2YX = P2G6/Z2C
      F2Y = F2YX-P2X11
      P2G7 = Z2D*Z2P2-Z2X*Z2P3
      P2G8 = (Z2D**2)*Z2P2-(Z2X**2)*Z2P3
      P2G9 = (Z2D**3)*Z2P2-(Z2X**3)*Z2P3
     DIM = (Z2C*P2G5-P2G6*P2G1)/P2G55
     DIS = (Z2C*P2G7-P2G6*P2G2)/P2G55
     DIMM = (Z2C*(Z2C*P2G7-P2G6*P2G2)
    *       -2*P2G1*(Z2C*P2G5-P2G6*P2G1))/P2G56
     DIMS = (Z2C*(Z2C*(P2G8-P2G5)+P2G1*P2G7
    *       -P2G6*(P2G3-P2G1)-P2G5*P2G2)
    *        -2*P2G1*(Z2C*P2G7-P2G6*P2G2))/P2G56
     DISS = (Z2C*(Z2C*(P2G9-P2G7)-P2G6*(P2G4-P2G2))-(2*P2G2+Z2C)
    *       *(Z2C*P2G7-P2G6*P2G2))/P2G56
      H2G1 = 2*F2Y*DIM/P2X22
      H2G2 = 2*F2Y*DIS/P2X22
      H2G11 = 2*(F2Y*DIMM+DIM**2)/P2X22
      H2G21 = 2*(F2Y*DIMS+DIM*DIS)/P2X22
      H2G22 = 2*(F2Y*DISS+DIS**2)/P2X22
      SDI1 = SDI1+H2G1
      SDI2 = SDI2+H2G2
     SDI11 = SDI11+H2G11
      SDI21 = SDI21+H2G21
      SDI22 = SDI22+H2G22
10    CONTINUE
        GGG1 = SDI1
        GGG2 = SDI2
        GGG11 = SDI11
        GGG21 = SDI21
        GGG12 = GGG21
        GGG22 = SDI22
        DET2 = (GGG11*GGG22)-(GGG12*GGG21)
          IF (DET2.EQ.0) THEN
             DIFG = 'Y'
```

```
            GOTO 90
          END IF
        PP2 = (GGG12*GGG2)-(GGG22*GGG1)
        RP2 = (GGG21*GGG1)-(GGG11*GGG2)
        P21 = PP2/DET2+P20
        R21 = RP2/DET2+R20
    IF ((ABS(P21-P20).LT.0.001).AND.(ABS(R21-R20).LT.0.001)) THEN
              GOTO 80
       ELSE
              P20 = P21
              R20 = R21
       END IF
          IF (IC2.GE.MAX) THEN
              NOTG = 'Y'
              GOTO 90
          END IF
20     CONTINUE
80   XMEW3 = P21
     XSIG3 = R21
90   RETURN
     END
```

โปรแกรมย่อยวิธีการประมาณค่าพารามิเตอร์ 3 วิธีคือ ML; MD และ MWD ของการแจกแจงความเสียหายแบบโลจิสติค ดังนี้

```
C*********************************************
C      ESTIMATION OF THE PARAMETERS BY
C       METHOD OF MAXIMUM LIKELIHOOD
C         FOR LOGISTIC DIST.
C*********************************************
      SUBROUTINE MLE(MAX,ALPHA1,BETA1)
      COMMON/PAR3/ALPHA,BETA,ALPHA0,BETA0
     *     /FACT/XD,XM,N
     *     /SAMP/X(500)
     *     /INDEX/DIFG,NOTG,DIG
      CHARACTER DIFG*1,NOTG*1,DIG*1
```

```
P0 = ALPHA0
R0 = BETA0
IC = 0
DO 20 J=1,MAX
   CALL COMPG3(P0,R0,BEG1,BEG2,BEG11,BEG21,BEG22)
      IF ((DIFG.EQ.'Y').OR.(DIG.EQ.'Y')) GOTO 90
     G1=0.0
     G2=0.0
     G11=0.0
     G21=0.0
     G22=0.0
     SDIF1=0.0
     SDIF2=0.0
     SDIF11=0.0
     SDIF21=0.0
     SDIF22=0.0
        IC=IC+1
     DO 10 I=1,N
        C = (X(I)-P0)/R0
        C1 = -1*C
        C2 = EXP(C1)
        FX = 1/(1+C2)
        FXX = (C2*FX/R0)*FX
        PH1 = (1-C2)*FX
  ALFX = PH1/R0
  BEFX = (C*PH1-1)/R0
        A2FX = (-2*FXX+((PH1**2)/R0))/R0
  DIF11 = A2FX-ALFX**2
        PH21 = -2*C*FXX-ALFX
        PH22 = BEFX*PH1
        ABFX = (PH21+PH22)/R0
  DIF21 = ABFX-ALFX*BEFX
        PH31 = C2*FX
        PH32 = C*(-PH31-PH1*PH31)-PH1
        PH33 = C*PH32/R0
        PH34 = BEFX-(1/R0)
```

```
            PH35 = BEFX*R0*PH34
            B2FX = (PH33+PH35)/R0
          DIF22 = B2FX-BEFX**2
            SDIF1 = SDIF1+ALFX
            SDIF2 = SDIF2+BEFX
            SDIF11 = SDIF11+DIF11
            SDIF21 = SDIF21+DIF21
            SDIF22 = SDIF22+DIF22
10          CONTINUE
              G1 = SDIF1+BEG1
              G2 = SDIF2+BEG2
              G11 = SDIF11+BEG11
              G21 = SDIF21+BEG21
             G12 = G21
              G22 = SDIF22+BEG22
             DET = (G11*G22)-(G21*G12)
               IF (DET.EQ.0) THEN
                 DIFG = 'Y'
                 GOTO 90
               END IF
            PP = (G12*G2)-(G22*G1)
            RP = (G21*G1)-(G11*G2)
            P1 = PP/DET+P0
            R1 = RP/DET+R0
        IF ((ABS(P1-P0).LT.0.001).AND.(ABS(R1-R0).LT.0.001)) THEN
            GOTO 80
        ELSE
            P0 = P1
            R0 = R1
        END IF
            IF (IC.GE.MAX) THEN
              NOTG = 'Y'
              GOTO 90
            END IF
20      CONTINUE
80      ALPHA1 = P1
```

```
      BETA1 = R1
90    RETURN
      END
C*******************************
C     COMPUTING VALUE G
C*******************************
      SUBROUTINE COMPG3(P0,R0,BEG1,BEG2,BEG11,BEG21,BEG22)
      COMMON/FACT/XD,XM,N
     *     /INDEX/DIFG,NOTG,DIG
      CHARACTER DIFG*1,NOTG*1,DIG*1
       A = (XD-P0)/R0
      B = (XM-P0)/R0
       A1 = -1*A
       B1 = -1*B
       A2 = EXP(A1)
       B2 = EXP(B1)
        FD = 1/(1+A2)
        FM = 1/(1+B2)
        F1 = FM-FD
      IF (F1.EQ.0) THEN
         DIFG = 'Y'
         GOTO 99
      END IF
        ALFM = (-B2*FM/R0)*FM
        ALFD = (-A2*FD/R0)*FD
        ALF = ALFM-ALFD
        BEFM = ALFM*B
        BEFD = ALFD*A
        BEF = BEFM-BEFD
       PG1 = (1-B2)*FM
        A2FM = ALFM*PG1/R0
       PG2 = (1-A2)*FD
        A2FD = ALFD*PG2/R0
        A2F = A2FM-A2FD
       PG11 = B1*PG1
       PG21 = PG11+1
```

```
      ABFM =-1*ALFM*PG21/R0
      PG12 = A1*PG2
      PG22 = PG12+1
      ABFD = -1*ALFD*PG22/R0
      ABF = ABFM-ABFD
      PG41 = B*BEFM*PG1/R0
      PG51 = -2*BEFM/R0
      B2FM = PG41+PG51
      PG42 = A*BEFD*PG2/R0
      PG52 = -2*BEFD/R0
      B2FD = PG42+PG52
      B2F = B2FM-B2FD
        BEG1 = -N*ALF/F1
        BEG2 = -N*BEF/F1
    PG3 = (ALF/F1)*ALF
    PG4 = A2F-PG3
      BEG11 = -N*PG4/F1
    PG5 = (ALF/F1)*BEF
    PG6 = ABF-PG5
      BEG21 = -N*PG6/F1
    PG7 = (BEF/F1)*BEF
    PG8 = B2F-PG7
      BEG22 = -N*PG8/F1
99  RETURN
    END
C*********************************************
C     ESTIMATION OF THE PARAMETERS BY
C       METHOD OF MINIMUM  DISTANCE
C        FOR LOGISTIC DIST.
C*********************************************
    SUBROUTINE MDE(MAX,ALPHA2,BETA2)
    COMMON/PAR3/ALPHA,BETA,ALPHA0,BETA0
   *      /FACT/XD,XM,N
   *      /SAMP/X(500)
   *      /SIM/FQ(500),F(500)
   *      /INDEX/DIFG,NOTG,DIG
    CHARACTER DIFG*1,NOTG*1,DIG*1
```

```
P10 = ALPHA0
R10 = BETA0
IC1 = 0
DO 20 J1=1,MAX
      GG1=0.0
      GG2=0.0
      GG11=0.0
      GG12=0.0
      GG21=0.0
      GG22=0.0
      SDF1=0.0
      SDF2=0.0
      SDF11=0.0
      SDF12=0.0
      SDF21=0.0
      SDF22=0.0
         IC1=IC1+1
C1M = (XM-P10)/R10
D1M = (XD-P10)/R10
C11M = -1*C1M
D11M = -1*D1M
C21M = EXP(C11M)
 D21M = EXP(D11M)
 F1M = 1/(1+C21M)
 F1D = 1/(1+D21M)
 F11 = F1M-F1D
  IF (F11.EQ.0) THEN
     DIFG = 'Y'
     GOTO 90
  END IF
 AL1FM = (-C21M*F1M/R10)*F1M
 AL1FD = (-D21M*F1D/R10)*F1D
 AL1F = AL1FM-AL1FD
 BE1FM = AL1FM*C1M
 BE1FD = AL1FD*D1M
   BE1F = BE1FM-BE1FD
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
PJ1 = (1-C21M)*F1M
A21FM = AL1FM*PJ1/R10
PJ2 = (1-D21M)*F1D
A21FD = AL1FD*PJ2/R10
 A21F = A21FM-A21FD
PJ11 = C11M*PJ1
PJ21 = PJ11+1
AB1FM = -1*AL1FM*PJ21/R10
PJ12 = D11M*PJ2
PJ22 = PJ12+1
AB1FD = -1*AL1FD*PJ22/R10
 AB1F = AB1FM-AB1FD
PJ41 = C1M*BE1FM*PJ1/R10
PJ51 = -2*BE1FM/R10
B21FM = PJ41+PJ51
PJ42 = D1M*BE1FD*PJ2/R10
PJ52 = -2*BE1FD/R10
B21FD = PJ42+PJ52
 B21F = B21FM-B21FD
  DO 10 IM=1,N
    P1X1 = F(IM)
     IF (P1X1.EQ.0) THEN
       DIFG = 'Y'
       GOTO 90
     END IF
   Q1M = (X(IM)-P10)/R10
Q11M = -1*Q1M
Q21M = EXP(Q11M)
F1X = 1/(1+Q21M)
F21 = F1X-F1D
F1YX = F21/F11
F1Y = F1YX-P1X1
AL1FX = (-Q21M*F1X/R10)*F1X
 AL11F = AL1FX-AL1FD
BE1FX = AL1FX*Q1M
 BE11F = BE1FX-BE1FD
```

```
      PJ3 = (1-Q21M)*F1X
      A21FX = AL1FX*PJ3/R10
       A211F = A21FX-A21FD
      PJ13 = Q11M*PJ3
      PJ23 = PJ13+1
      AB1FX = -1*AL1FX*PJ23/R10
       AB11F = AB1FX-AB1FD
      PJ71 = Q1M*BE1FX*PJ3/R10
      PJ81 = -2*BE1FX/R10
      B21FX = PJ71+PJ81
        B211F = B21FX-B21FD
      DFA = ((F11*AL11F-F21*AL1F)/F11)/F11
      DFB = ((F11*BE11F-F21*BE1F)/F11)/F11
      DFAA = (((F11*A211F-F21*A21F)/F11)/F11)
     *        -2*AL1F/F11*DFA
      DFAB = ((F11*AB11F+BE11F*AL1F-F21*AB1F
     *       -BE1F*AL11F)/F11)/F11-2*AL1F/F11*DFB
      DFBB = ((F11*B211F-F21*B21F)/F11)/F11
     *        -2*BE1F/F11*DFB
     H1G1 = 2*F1Y*DFA
     H1G2 = 2*F1Y*DFB
     H1G11 = 2*(F1Y*DFAA+DFA**2)
     H1G21 = 2*(F1Y*DFAB+DFA*DFB)
     H1G22 = 2*(F1Y*DFBB+DFB**2)
     SDF1 = SDF1+H1G1
      SDF2 = SDF2+H1G2
      SDF11 = SDF11+H1G11
      SDF21 = SDF21+H1G21
     SDF22 = SDF22+H1G22
10    CONTINUE
       GG1 = SDF1
       GG2 = SDF2
       GG11 = SDF11
       GG21 = SDF21
       GG12 = GG21
       GG22 = SDF22
```

```
      DET1 = (GG11*GG22)-(GG12*GG21)
        IF (DET1.EQ.0) THEN
          DIFG = 'Y'
          GOTO 90
        END IF
      PP1 = (GG12*GG2)-(GG22*GG1)
      RP1 = (GG21*GG1)-(GG11*GG2)
      P11 = PP1/DET1+P10
      R11 = RP1/DET1+R10
    IF ((ABS(P11-P10).LT.0.001).AND.(ABS(R11-R10).LT.0.001)) THEN
            GOTO 80
     ELSE
            P10 = P11
            R10 = R11
    END IF
        IF (IC1.GE.MAX) THEN
            NOTG = 'Y'
            GOTO 90
        END IF
20    CONTINUE
80   ALPHA2 = P11
    BETA2 = R11
90   RETURN
    END
C*********************************************
C     ESTIMATION OF THE PARAMETERS BY
C     METHOD OF MINIMUM WEIGHT DISTANCE
C         FOR LOGISTIC DIST.
C*********************************************
    SUBROUTINE MWD(MAX,ALPHA3,BETA3)
    COMMON/PAR3/ALPHA,BETA,ALPHA0,BETA0
   *     /FACT/XD,XM,N
   *     /SAMP/X(500)
   *     /SIM/FQ(500),F(500)
   *     /INDEX/DIFG,NOTG,DIG
    CHARACTER DIFG*1,NOTG*1,DIG*1
```

```
P20 = ALPHA0
R20 = BETA0
IC2 = 0
DO 20 J2=1,MAX
      GGG1=0.0
      GGG2=0.0
      GGG11=0.0
      GGG12=0.0
      GGG21=0.0
      GGG22=0.0
      SDI1=0.0
      SDI2=0.0
      SDI11=0.0
      SDI12=0.0
      SDI21=0.0
      SDI22=0.0
         IC2=IC2+1
 C2M = (XM-P20)/R20
 D2M = (XD-P20)/R20
C12M = -1*C2M
 D12M = -1*D2M
 C22M = EXP(C12M)
 D22M = EXP(D12M)
 F2M = 1/(1+C22M)
 F2D = 1/(1+D22M)
 F12 = F2M-F2D
    IF (F12.EQ.0) THEN
       DIFG = 'Y'
       GOTO 90
    END IF
 AL2FM = (-C22M*F2M/R20)*F2M
 AL2FD = (-D22M*F2D/R20)*F2D
   AL2F = AL2FM-AL2FD
 BE2FM = AL2FM*C2M
 BE2FD = AL2FD*D2M
   BE2F = BE2FM-BE2FD
```

```
PK1 = (1-C22M)*F2M
A22FM = AL2FM*PK1/R20
PK2 = (1-D22M)*F2D
A22FD = AL2FD*PK2/R20
  A22F = A22FM-A22FD
PK11 = C12M*PK1
PK21 = PK11+1
AB2FM = -1*AL2FM*PK21/R20
PK12 = D12M*PK2
PK22 = PK12+1
AB2FD = -1*AL2FD*PK22/R20
  AB2F = AB2FM-AB2FD
PK41 = C2M*BE2FM*PK1/R20
PK51 = -2*BE2FM/R20
B22FM = PK41+PK51
PK42 = D2M*BE2FD*PK2/R20
PK52 = -2*BE2FD/R20
B22FD = PK42+PK52
  B22F = B22FM-B22FD
    DO 10 IMM=1,N
      P2X1 = F(IMM)/(N+1)
      P2X11 = N*P2X1
      P2X12 = 1-P2X11
       IF ((P2X11.EQ.0).OR.(P2X12.EQ.0)) THEN
          DIFG = 'Y'
          GOTO 90
       END IF
     P2X22 = P2X1*P2X12
Q2M = (X(IMM)-P20)/R20
Q12M = -1*Q2M
Q22M = EXP(Q12M)
F2X = 1/(1+Q22M)
F22 = F2X-F2D
F2YX = F22/F12
F2Y = F2YX-P2X11
AL2FX = (-Q22M*F2X/R20)*F2X
```

```
      AL22F = AL2FX-AL2FD
     BE2FX = AL2FX*Q2M
      BE22F = BE2FX-BE2FD
     PK3 = (1-Q22M)*F2X
     A22FX = AL2FX*PK3/R20
      A222F = A22FX-A22FD
     PK13 = Q12M*PK3
     PK23 = PK13+1
     AB2FX = -1*AL2FX*PK23/R20
       AB22F = AB2FX-AB2FD
     PK71 = Q2M*BE2FX*PK3/R20
     PK81 = -2*BE2FX/R20
     B22FX = PK71+PK81
        B222F = B22FX-B22FD
    DIA = ((F12*AL22F-F22*AL2F)/F12)/F12
    DIB = ((F12*BE22F-F22*BE2F)/F12)/F12
    DIAA = (((F12*A222F-F22*A22F)/F12)/F12)
   *       -2*AL2F/F12*DIA
    DIAB = ((F12*AB22F+BE22F*AL2F-F22*AB2F
   *       -BE2F*AL22F)/F12)/F12-2*AL2F/F12*DIB
    DIBB = ((F12*B222F-F22*B22F)/F12)/F12-2*BE2F/F12*DIB
     H2G1 = 2*F2Y*DIA/P2X22
     H2G2 = 2*F2Y*DIB/P2X22
     H2G11 = 2*(F2Y*DIAA+DIA**2)/P2X22
     H2G21 = 2*(F2Y*DIAB+DIA*DIB)/P2X22
     H2G22 = 2*(F2Y*DIBB+DIB**2)/P2X22
      SDI1 = SDI1+H2G1
      SDI2 = SDI2+H2G2
      SDI11 = SDI11+H2G11
      SDI21 = SDI21+H2G21
      SDI22 = SDI22+H2G22
10    CONTINUE
       GGG1 = SDI1
       GGG2 = SDI2
       GGG11 = SDI11
       GGG21 = SDI21
```

```
      GGG12 = GGG21
      GGG22 = SDI22
      DET2 = (GGG11*GGG22)-(GGG12*GGG21)
        IF (DET2.EQ.0) THEN
          DIFG = 'Y'
          GOTO 90
        END IF
      PP2 = (GGG12*GGG2)-(GGG22*GGG1)
      RP2 = (GGG21*GGG1)-(GGG11*GGG2)
      P21 = PP2/DET2+P20
     R21 = RP2/DET2+R20
    IF ((ABS(P21-P20).LT.0.001).AND.(ABS(R21-R20).LT.0.001)) THEN
              GOTO 80
     ELSE
              P20 = P21
              R20 = R21
     END IF
        IF (IC2.GE.MAX) THEN
              NOTG = 'Y'
              GOTO 90
        END IF
20     CONTINUE
80   ALPHA3 = P21
    BETA3 = R21
90   RETURN
    END
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียน

นางสาว อุไรวรรณ เจริญกีรติกุล เกิดเมื่อวันที่ 11 กรกฎาคม 2513 สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต(วท.บ.) จากภาคสถิติ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น ในปีการศึกษา 2535 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต(วท.ม.) ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2537 ปัจจุบันรับราชการในตำแหน่งอาจารย์ ภาควิชาสถิติประยุกต์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย