รายการอ้างอิง

ภาษาไทย

จงดี โรจนประศาสน์. การเปรียบเทียบวิธีการประมาณค่าพารามิเตอร์ในสมการถดถอยเชิงเส้นอย่างง่ายเมื่อตัวแปรตามมีค่าที่ถูกตัดทิ้ง วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย

ชำเลือง ชาติสุวรรณ และ พรพรรณ วิบูลย์ศิริรัตน์. การสร้างตารางมรณะไทย 2529 กองประกันชีวิต กรมประกันภัย.

ธีระพร วีระถาวร. ความน่าจะเป็นกับการประยุกต์ กรุงเทพมหานคร : สำนักพิมพ์อักษรกราฟฟิค , 2537.

_____. การอนุมานเชิงสถิติขั้นกลาง : โครงสร้างและความหมาย กรุงเทพมหานคร : พิทักษ์การพิมพ์ , 2531.

ภาษาอังกฤษ

Batten, R.W. Mortality Table Construction Prentice-Hall, Inc. England Cliffs, Newjersy, 1978.

Bowers, N.L. Actuarial Mathematics The Society of Actuaries, 1986.

Elandt-Johnson, R.C. and Johnson, N.L. Survival Models and Data Analysis New York : John Wiley & Sons, 1981.

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

Jordan, C.W. Life Contingencies The Society of Actuaries, 1975.

Miller, R.G. Survival Analysis New York : John Wiley & Sons, 1981.

Nelson, W. Applied Life Data Analysis New York : John Wiley & Sons, 1982.

ภาคผนวก

ภาคผนวก ก

ตาราง ก. แสดงค่า q_x และค่าพารามิเตอร์ k และ n สำหรับการแจกแจงแบบไวบูลล์ B และ c เป็นค่าพารามิเตอร์ สำหรับการแจกแจงแบบกอมเพิรตซ์ ซึ่งนำค่าพารามิเตอร์นี้ไปจำลองข้อมูลเกี่ยวกับระยะเวลาที่จะมีชีวิตอยู่ต่อไปในอนาคต

x	q_x	กำหนด n = 1.0 และค่า k เป็นดังนี้	กำหนด c = 1.08 และค่า B เป็นดังนี้
25	0.0012330	0.002467522	0.001186894
26	0.0012735	0.002548623	0.001225904
27	0.0013288	0.002659367	0.001279173
28	0.0013895	0.002780933	0.001337647
29	0.0014560	0.002914122	0.001401712
30	0.0015289	0.003060140	0.001471947
31	0.0016089	0.003220391	0.001549029
32	0.0016965	0.003395881	0.001633441
33	0.0017927	0.003588618	0.001726148
34	0.0018980	0.003799607	0.001827636
35	0.0020136	0.004031260	0.001939062
36	0.0021402	0.004284987	0.002061107
37	0.0022791	0.004563402	0.002195026
38	0.0024313	0.004868521	0.002341790
39	0.0025982	0.005203162	0.002502755
40	0.0027812	0.005570149	0.002679278
41	0.0029818	0.005972509	0.002872816
42	0.0032017	0.006413673	0.003085018
43	0.0034427	0.006897279	0.003317636
44	0.0037070	0.007427776	0.003572809

ตาราง ก. (ต่อ) แสดงค่า q_x และค่าพารามิเตอร์ k และ n สำหรับการแจกแจงแบบไวบูลล์ B และ c เป็นค่าพารามิเตอร์ สำหรับการแจกแจงแบบกอมเพิรตซ์ ซึ่งนำค่าพารามิเตอร์นี้ไปจำลองข้อมูลเกี่ยวกับระยะเวลาที่จะมีชีวิตอยู่ต่อไปในอนาคต

x	q_x	กำหนด n = 1.0 และค่า k เป็นดังนี้	กำหนด c = 1.08 และค่า B เป็นดังนี้
45	0.0039966	0.008009215	0.003852485
46	0.0043141	0.008646865	0.004159198
47	0.0046621	0.009346003	0.004495488
48	0.0050436	0.010112723	0.004864286
49	0.0054617	0.010953339	0.005268627
50	0.0059199	0.011874984	0.005711945
51	0.0064221	0.012885620	0.006198067
52	0.0069724	0.013993641	0.006731033
53	0.0075755	0.015208679	0.007315474
54	0.0082364	0.016541013	0.007956335
55	0.0089605	0.018001773	0.008658970
56	0.0097538	0.019603359	0.009429344
57	0.0106230	0.021359653	0.010274132
58	0.0115752	0.023285428	0.011200442
59	0.0126181	0.025396768	0.012216010
60	0.0137604	0.027711903	0.013329606
61	0.0150114	0.030250422	0.014550650
62	0.0163813	0.033033914	0.015889527
63	0.0178812	0.036086000	0.017357601
64	0.0195231	0.039432386	0.018967234
65	0.0213203	0.043101721	0.020732208

<u>หมายเหตุ</u> ค่า q_x ที่ใช้ในการจำลองข้อมูลนี้ นำมาจากภาคผนวก 2A ในหน้า 560-561 ของหนังสือ Actuarial Mathematics

ภาคผนวก ข

```
C*************************************************************************C
C***                          MAIN PROGRAM                             ***C
C***                ESTIMATING MORTALITY PROBABILITY                   ***C
C***                             FOR                                   ***C
C***                INCOMPLETE LIFE INSURANCE DATA                     ***C
C*************************************************************************C

      COMMON/ DOME/ T(200) , T1(1000)
     *          / SEED/ IX
     *          / PARA1/ AK , AN , B , C
     *          / PARA2/ XK0 , XN0 , B0 , C0
     *          / DTA1/ NM(8), IE(8)
     *          / DTA2/ M, K
     *          / DTA3/ SUMT
     *          / MORT/ TQX , QX1 , QX2 , QX3
       IX  =  881
       IA  =  25
       TQX  =  0.0012330
      NM(1) = 30
      NM(2) = 50
      NM(3) = 70
      NM(4) = 100
      NM(5) = 300
      NM(6) = 500
      NM(7) = 700
      NM(8) = 1000
```

```
      DO  5    ID  =  1 , 2
          IF  (ID .EQ. 1)  THEN
              AK  =  0.002467522
              AN  =  1.0
              WRITE (6,10)
10            FORMAT (25X ,' ***  WEIBULL  DISTRIBUTION  *** ')
          ELSE
              B  =  0.001186894
              C  =  1.08
              WRITE (6,15)
15            FORMAT (25X ,' ***  GOMPERTZ  DISTRIBUTION  *** ')
          ENDIF
          WRITE (6,20)  IA , IX , TQX
20        FORMAT (23X , 'AGE' , I2 , 'SEED' , I5 , ' QX  =  ' , F10.7)
          WRITE (6,25)
25        FORMAT (80('-')
     *          /3X , 'N' , 2X , ':' , 10X , 'CLS' , 10X , ':' , 10X , 'MLE' , 10X , ':' ,
     *          9X , 'BAYES'/6X , 3(':' , 4X , 'QX' , 5X , ':' , 3X , 'RAPE' , 4X)/80('-'))
            M = 0
            SUMT = 0.0
            DO 29  J = 1,8
              IF ( J .EQ. 1 ) .AND. ( J .EQ. 4 ) THEN
                  IE(J) =  30
              ELSE IF ( J .EQ. 2 ) .OR. ( J .EQ. 3 ) THEN
                  IE(J) = 20
              ELSE IF ( J .LE. 7 ) .OR. ( J .GE. 5 ) THEN
                  IE(J) = 200
              ELSE
                  IE(J) = 300
```

```
            ENDIF
29        CONTINUE
          DO 7   J  = 1,8
           SUM1(J)  =  0.0
          SUM2(J)  =  0.0
          SUM3(J)  =  0.0
7         CONTINUE
        DO  30    I =  1,500
          DO 35   K = 1,8
             IF  (ID .EQ. 1)  THEN
                 CALL  GENTWEI
                 CALL  FINDQXWB
             ELSE
                 CALL  GENTGOM
                 CALL  FINDQXGP
             ENDIF
               SUM1(K)  =  SUM1(K) + QX1
               SUM2(K)  =  SUM2(K) + QX2
               SUM3(K)  =  SUM3(K) + QX3
35        CONTINUE
30      CONTINUE
        DO 40   K = 1,8
          SBK1(K)  =  SUM1(K)/500.
          SBK2(K)  =  SUM2(K)/500.
          SBK3(K)  =  SUM3(K)/500.
          E1(K)  =  SBK1(k) - TQX
          E2(K)  =  SBK2(k) - TQX
          E3(K)  =  SBK3(k) - TQX
          RAPE1(K)  =  ABS(E1(K))*100/TQX
```

```
                RAPE2(k)  =  ABS(E2(K))*100/TQX
                RAPE3(k)  =  ABS(E3(K))*100/TQX
                WRITE (6,45)  NM(K) , SBK1(K) , RAPE1(K) , SBK2(K) , RAPE2(K) , SBK3(K)
                RAPE3(K)
   45          FORMAT (IX , I4 , IX , 3( ':' , F10.7 , IX , ':' , F11.4) )
   40       CONTINUE
            WRITE (6,50)
   50       FORMAT ( 80('-') )
    5   CONTINUE
        STOP
        END

C*************************************************************************C
C***                        STOP  MAIN  PROGRAM                        ***C
C*************************************************************************C
C***      FIND  THE  FUTURE  LIFETIME  UNDER  WIEBULL  DISTRIBUTION      ***C
C*************************************************************************C

        SUBROUTINE  GENTWEI
        COMMON/ DOME/ T(200) , T1(1000)
      *           / SEED/ IX
      *           / PARA1/ AK , AN , B , C
      *           / PARA2/ XK0 , XN0 , B0 , C0
      *           / DTA1/ NM(8), IE(8)
      *           / DTA2/ M, K
      *           / DTA3/ SUMT
      *           / MORT/ TQX , QX1 , QX2 , QX3
        IE1 = IE(K)
        DO  50    I  =  1 , IE1
```

```
            U2  =  RAND(IX)
            UU1  =  ALOG(U2)
            ANK  =  (AN+1)/AK
            AN1  =  1/(AN+1)
            TT1  =  (-ANK*UU1)**AN1
            IF  (TT1 .GE. 1.0)  THEN
                TT1  =  1.0
                T1(I)  =  1.0
            ELSE
                M  =  M+1
                T(M)  =  TT1
                T1(I)  =  TT1
            ENDIF
            SUMT  =  SUMT + TT1
   50   CONTINUE
        RETURN
        END

C*************************************************************************C
C***     FIND THE FUTURE LIFETIME UNDER GOMPERTZ DISTRIBUTION      ***C
C*************************************************************************C

        SUBROUTINE  GENTGOM
        COMMON/ DOME/ T(200) , T1(1000)
     *          / SEED/ IX
     *          / PARA1/ AK , AN , B , C
     *          / PARA2/ XK0 , XN0 , B0 , C0
     *          / DTA1/ NM(8), IE(8)
     *          / DTA2/ M, K
```

```
*           / DTA3/ SUMT
*           / MORT/ TQX , QX1 , QX2 , QX3
      IE1 = IE(K)
      DO 55   I = 1 , IE1
         U2 = RAND(IX)
         UU1 = ALOG(U2)
         UC = ALOG(C)
         UU2 = UC*UU1/B
         UUU1 = 1 - UU2
         UU3 = ALOG(UUU1)
         TT1 = UU3/UC
         IF (TT1 .GE. 1.0) THEN
            TT1 = 1.0
            T1(I) = 1.0
         ELSE
            M = M+1
            T(M) = TT1
            T1(M) = TT1
         ENDIF
         SUMT = SUMT + TT1
   50 CONTINUE
      RETURN
      END

C*************************************************************************C
C***      FIND MORTALITY PROBABILITY UNDER WEIBULL DISTRIBUTION     ***C
C*************************************************************************C

      SUBROUTINE FINDQXWB
```

```
      COMMON/ DOME/ T(200) , T1(1000)
     *            / SEED/ IX
     *            / PARA1/ AK , AN , B , C
     *            / PARA2/ XK0 , XN0 , B0 , C0
     *            / DTA1/ NM(8), IE(8)
     *            / DTA2/ M, K
     *            / DTA3/ SUMT
     *            / MORT/ TQX , QX1 , QX2 , QX3
      IJK = 0
 1    IJ = 0
11    CALL DETERWEI
21    IJ = IJ + 1
      IF ( M .EQ. 0 ) THEN
         QX1 = 0.
         QX2 = 0.
         QX3 = 1 - ( FLOAT ( 5000 + NM(K) ) / FLOAT ( 5001 + NM(K) ) )
         GOTO 61
      ENDIF
      IF ( ( XN0 .GT. 50 ) .OR. ( XN0 .LT. -50 ) ) THEN
         IJK = IJK + 1
         IF ( IJK .EQ. 50 ) THEN
            WRITE (6,31)
31          FORMAT (' *** TOO LONG EXECUTION *** ')
            STOP
            ENDIF
         CALL GENTWEI
         GOTO 1
      ENDIF
      SUM1 = 0.
```

```
      SUM2 = 0.
      SUM3 = 0.
      SUM4 = 0.
      DO 41  I = 1 , M
         SUM1 = SUM1 + T(I)**(XN0 + 1)
         SUM2 = SUM2 + ALOG( T(I) )*( T(I)**(XN0 + 1) )
         SUM3 = SUM3 + ALOG( T(I) )
         SUM4 = SUM4 + ( (ALOG( T(I) ) )**2 )*( T(I)**(XN0 + 1) )
41    CONTINUE
      X1 = (NM - M) + SUM1
      G1 = FLOAT (M)/XK0 - X1/(XN0 + 1)
      G11 = -FLOAT (M)/XK0**2
      G12 = X1/(XN0 + 1)**2 - SUM2/(XN0 + 1)
      G2 = SUM3 + XK0*X1/(XN0 + 1)**2 - XK0*SUM2/(XN0 + 1)
      G22 = -2*XK0*X1/(XN0+1)**3 + 2*XK0*SUM2/(XN0+1)**2 - XK0*SUM4/(XN0+1)
      BB1 = (G2*G12) - (G1*G22)
      BC = (G11*G22) - G12**2
      IF (BC .EQ. 0.0) THEN
         WRITE (6,51)
51       FORMAT (' *** THE DENOMINATOR EQUALS TO ZERO. *** ')
         STOP
      ENDIF
      BB = BB1/BC
      CC1 = (G1*G12) - (G2*G11)
      CC = CC1/BC
      XK = BB + XK0
      XN = CC + XN0
      IF ( (ABS (BB) .LE. 0.0001) .AND. (ABS (CC) .LE. 0.0001) ) THEN
         QX1 = FLOAT (M)/NM(K)
```

```
            QX2  =  1 - EXP ( -XK/(XN + 1) )
            BETA  =  -1/ALOG(1 - QX1)
            QX3  =  1 - ( (BETA + SUMT) / (BETA + 1 + SUMT) )**(M+1)
            GOTO  61
         ELSE
            IF  (ABS (BB) .LE. 0.0001)  THEN
               XK0  =  XK0
            ELSE
               XK0  =  XK
            ENDIF
            IF  ( ABS (CC) .LE. 0.0001)  THEN
               XN0  =  XN0
            ELSE
               XN0  =  XN
            ENDIF
            GOTO  21
         ENDIF
   61    RETURN
         END

C*************************************************************************C
C***     FIND MORTALITY PROBABILITY UNDER GOMPERTZ DISTRIBUTION     ***C
C*************************************************************************C

         SUBROUTINE  FINDQXGP
         COMMON/ DOME/ T(200) , T1(1000)
        *           / SEED/ IX
        *           / PARA1/ AK , AN , B , C
        *           / PARA2/ XK0 , XN0 , B0 , C0
```

```
*           / DTA1/ NM(8), IE(8)
*           / DTA2/ M, K
*           / DTA3/ SUMT
*           / MORT/ TQX , QX1 , QX2 , QX3
    IJK = 0
 2  IJ = 0
22  CALL DETERWEI
32  IJ = IJ + 1
    IF ( M .EQ. 0 ) THEN
        QX1 = 0.
        QX2 = 0.
        QX3 = 1 - ( FLOAT ( 5000 + NM(K) ) / FLOAT ( 5001 + NM(K) ) )
        GOTO 72
    ENDIF
    IF ( ( C0 .GT. 10 ) .OR. ( C0 .LT. 0 ) .OR. ( C0 .EQ. 1 ) ) THEN
        CALL GENTGOM
        GOTO 22
    ENDIF
    IF ( IJ .EQ. 50 ) THEN
        IJK = IJK + 1
        IF ( IJK .EQ. 50 ) THEN
            WRITE (6,42)
42          FORMAT (' *** TOO LONG EXECUTION *** ')
            STOP
        ENDIF
        CALL GENTGOM
        GOTO 2
    ENDIF
    SUM1 = 0.
```

```
      SUM2  =  0.
      SUM3  =  0.
      SUM4  =  0.
      DO  52    I  =  1 , M
          SUM1  =  SUM1 + C0**T(I) - 1
          SUM2  =  SUM2 + T(I)*(C0**( T(I) - 1) )
          SUM3  =  SUM3 + T(I)*( T(I) - 1 )*( C0**( T(I) - 2 ) )
          SUM4  =  SUM4 + T(I)
52    CONTINUE
      X1  =  (NM - M)*(C0 - 1) + SUM1
      X2  =  (NM - M) + SUM2
      G1  =  FLOAT (M)/B0 - X1/ALOG (C0)
      G11  =  -FLOAT (M)/B0**2
      AG12  =  C0*ALOG (C0)**2
      G12  =  X1/AG12 - X2/ALOG (C0)
      G2  =  SUM4/C0 + B0*X1/AG12 - B0*X2/ALOG (C0)
      AG22  =  B0/(C0*ALOG (C0))**2
      BG22  =  2*B0/( (C0**2)*ALOG (C0)**3 )
      G22  =  -SUM4/C0**2 - (AG22+BG22)*X1 + 2*B0*X2/AG12 - B0*SUM3/ALOG(C0)
      BB1  =  (G2 * G12) - (G1 * G22)
      BC  =  (G11 * G22) - G12**2
      IF  (BC .EQ. 0.0)  THEN
          WRITE (6,62)
62        FORMAT (' ***  THE  DENOMINATOR  EQUALS  TO  ZERO.  *** ')
          STOP
      ENDIF
      BB  =  BB1/BC
      CC1  =  (G1*G12) - (G2*G11)
      CC  = CC1/BC
```

```
      XB  =  BB + B0
      XC  =  CC + C0
      IF  ( (ABS (BB) .LE. 0.0001) .AND. (ABS (CC) .LE. 0.0001) )  THEN
          QX1  =  FLOAT (M)/NM(K)
          QX2  =  1 - EXP ( -XB*(XC - 1)/ALOG(XC) )
          BETA  =  -1 / ALOG(1 - QX1)
          QX3  =  1 - ( (BETA + SUMT) / (BETA + 1 + SUMT) ) ** (M+1)
          GOTO  72
      ELSE
          IF  (ABS (BB) .LE. 0.0001)  THEN
              B0  =  B0
          ELSE
              B0  =  XB
          ENDIF
          IF  ( ABS (CC) .LE. 0.0001)  THEN
              C0  =  C0
          ELSE
              C0  =  XC
          ENDIF
          GOTO  32
      ENDIF
   72 RETURN
      END

C*************************************************************************C
C***              DETERMINE THE INITIAL VALUES N0 AND K0              ***C
C***                  FOR NEWTON - RAPHSON METHOD                     ***C
C*************************************************************************C
```

```
      SUBROUTINE  DETERWEI
      COMMON/ DOME/ T(200) , T1(1000)
     *            / SEED/ IX
     *            / PARA1/ AK , AN , B , C
     *            / PARA2/ XK0 , XN0 , B0 , C0
     *            / DTA1/ NM(8), IE(8)
     *            / DTA2/ M, K
     *            / DTA3/ SUMT
     *            / MORT/ TQX , QX1 , QX2 , QX3
      NM1  =  NM(K) - 1
      DO  3    I  =  1 , NM1
          IJ  =  I + 1
          NM2 = NM(K)
          DO  13    J  =  IJ , NM2
              IF  (T1(I) .GT. T1(J))  THEN
                    TEMP1  =  T1(I)
                    T1(I)  =  T1(J)
                    T1(J)  =  TEMP1
              ENDIF
13        CONTINUE
 3    CONTINUE
      JK  =  0
23    JK  =  JK + 1
      IF  (JK .EQ. 1000)  THEN
          WRITE (6,33)
33        FORMAT (' THE  DATERMINATION  IS  TOO  LONG ')
      ENDIF
      IF  (NM(K) .GE. 500)  THEN
          AMUL  =  1000
```

```
      ELSE
         AMUL = 10000
      ENDIF
   43 Z1 = RAND(IX)
      N1 = INT ( Z1*AMUL)
      IF ( ( N1 .GT. NM(K) ) .OR. ( N1 .EQ. 0) GOTO 43
   53 Z2 = RAND(IX)
      N2 = INT (Z2*AMUL)
      IF ( ( N2 .GT. NM(K) ) .OR. ( N2 .EQ. 0) .OR. ( N2 .EQ. N1) GOTO 53
      FTN1 = FLOAT(N1)/FLOAT(NM(K))
      FTN2 = FLOAT(N2)/FLOAT(NM(K))
      XD1 = ALOG( 1 - FTN1)
      XD2 = ALOG(1 - FTN2)
      XD = XD1/XD2
      GXD = ALOG(XD)
      ST = T1(N1)/T1(N2)
      XT = ALOG(ST)
      XN0 = GXD/XT - 1
      IF ( ( XN0 .LE. 0) .OR. ( XN0 .GE. 100) GOTO 23
      TN = T1(N2)**(XN0 + 1)
      XK0 = -XD2*(XN0 + 1)/TN
      IF ( (XK0 .LE. 0 ) .OR. (XK0 .GE. 100) GOTO 23
      RETURN
      END

C*************************************************************************C
C***              DETERMINE THE INITIAL VALUES B0 AND C0              ***C
C***                  FOR NEWTON - RAPHSON METHOD                     ***C
C*************************************************************************C
```

```
      SUBROUTINE  DETERGOM
      COMMON/ DOME/ T(200) , T1(1000)
     *            / SEED/ IX
     *            / PARA1/ AK , AN , B , C
     *            / PARA2/ XK0 , XN0 , B0 , C0
     *            / DTA1/ NM(8), IE(8)
     *            / DTA2/ M, K
     *            / DTA3/ SUMT
     *            / MORT/ TQX , QX1 , QX2 , QX3
      MAX  =  500
      NM1  =  NM(K) - 1
      DO  4    I  =  1 , NM1
          IJ  =  I + 1
          NM2 = NM(K)
          DO  14    J  =  IJ , NM2
              IF  ( T1(I) .GT. T1(J) )  THEN
                    TEMP1  =  T1(I)
                    T1(I)  =  T1(J)
                    T1(J)  =  TEMP1
              ENDIF
14        CONTINUE
 4    CONTINUE
      JK  =  0
      JK  =  JK + 1
      IF  ( JK .EQ. 1000 )  THEN
          WRITE (6,34)
34        FORMAT (' THE  DETERMINATION  IS  TOO  LONG ')
          STOP
```

```
      ENDIF
      IF  (NM(K) .GE. 500)  THEN
          AMUL  =  1000
      ELSE
          AMUL  =  10000
      ENDIF
44    Z1  =  RAND (IX)
      N1  =  INT (Z1*AMUL)
      IF  ( ( N1 .GT. NM(K) ) .OR. ( N1 .EQ. 0 ) .OR. ( N1 .LT. 50 ) )  GOTO  44
54    Z2  =  RAND (IX)
      N2  =  INT (Z2*AMUL)
      IF  ( ( N2 .GT. NM(K) ) .OR. ( N2 .EQ. 0 ) .OR. ( N2 .GE. N1 ) )  GOTO  54
      R  =  T1(N2)/T1(N1)
      IF  ( R .GE. 1.)  GOTO  44
      FTN1  =  FLOAT(N1)/FLOAT(NM(K))
      FTN2  =  FLOAT(N2)/FLOAT(NM(K))
      XK1  =  ALOG (1 - FTN1)
      XK2  =  ALOG (1 - FTN2)
      XK  =  XK1/XK2
      SR  =  1/R
      SR1  =  1/(1 - R)
      X0  =  (XK*R)**(SR1)
      TOL  =  0.0001
      IF  ( (XK .GT. 1.) .AND. (XK .LT. SR) )  THEN
          A1  =  0.0
          A2  =  X0
          DO  64    I  =  1 , MAX
              P  =  A1 + (A2 - A1)/2
              FP  =  XK*(P**R) - P + 1 - XK
```

```
            D  =  (A2 - A1)/2
            IF  ( (FP .EQ. 0.) .OR. (D .LT. TOL) )  THEN
                  H1  =  ALOG (P) / T(N1)
                  C0  =  EXP (H1)
                  IF  ( (C0 .LE. 1.) .OR. (C0 .GE. 10.) )  GOTO  24
                  H2  =  C0**T1(N2) - 1
                  B0  =  -XK2*H1/H2
                  IF  ( (B0 .LE. 0.) .OR. (B0 .GE. 10.) )  GOTO  24
                  GOTO  114
            ENDIF
            FA1  =  XK*(A1**R) - A1 + 1 - XK
            F  =  FA1*FP
            IF  (F .GT. 0)  THEN
                  A1  =  P
            ELSE
                  A2  =  P
            ENDIF
64          CONTINUE
            WRITE (6,74)  MAX
74          FORMAT (' METHOD FAILED AFTER NO ITERATIONS, N0 = ', I5 )
         ELSE  IF  (XK .GT. SR)  THEN
            X1  =  X0 + 1
84          FX1  =  XK*(X1**R) - X1 + 1 - XK
            IF  (FX1 .GT.0.)  THEN
               X1  =  X1 + 1
               GOTO  84
            ENDIF
            A1  =  X0
            A2  =  X1
```

```
      DO  94    I  =  1 , MAX
          P  =  A1 + (A2 - A1)/2
          FP  =  XK*(P**R) - P + 1 - XK
          D  =  (A2 - A1)/2
          IF  ( ( FP .EQ. 0.) .OR. ( D .LT. TOL) )  THEN
                H1  =  ALOG(P) /T1(N1)
                C0  =  EXP(H1)
                IF  ( (C0 .LE. 1.) .OR. (C0 .GE. 10.) )  GOTO  24
                H2  =  C0**T1(N2) - 1
                B0  =  -XK2*H1/H2
                IF  ( ( B0 .LE. 0.) .OR. (B0 .GE. 10.)  GOTO  24
                GOTO  114
          ENDIF
          FA1  =  XK*(A1**R) - A1 + 1 - XK
          F  =  FA1*FP
          IF  (F .GT. 0.)  THEN
                A1  =  P
          ELSE
                A2  =  P
          ENDIF
94        CONTINUE
          WRITE (6,104)  MAX
104       FORMAT (' METHOD  FAILED  AFTER  NO  ITERATIONS , N0  =  ' , I5 )
      ENDIF
114   RETURN
      END
```

```
C*************************************************************************C
C***                        FUNCTION  RANDOM (0,1)                     ***C
C*************************************************************************C

      FUNCTION  RAND (IX)
      IX  =  IX*16807
      IF  (IX .LT. 0)    IX  =  IX +2147483647 + 1
      RAND  =  IX
      RAND  =  RAND*0.465661E - 9
      RETURN
      END
```

ประวัติผู้เขียน

นายสมบัติ กุลวุฒิ เกิดวันที่ 28 ธันวาคม พ.ศ.2512 สำเร็จการศึกษา ปริญญาวิทยาศาสตรบัณฑิต จากภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยมหิดล ในปีการศึกษา 2534 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาการประกันภัย ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2535