ภาคผนวก ค
ตัวอย่างโปรแกรมภาษาฟอร์แทรน
สำหรับการวิเคราะห์เชิงตัวเลข

บทที่ 2

ตัวอย่างการเรียกใช้ Subroutine **NEWTON RAPHSON**

```
C*****RNR******
      EXTERNAL F,FP
10    WRITE(6,1)
1     FORMAT(1X,'ENTER NUMSIG,MAXIT,XO(INITIAL GUESs) ')
      READ(5,*)NUMSIG,MAXIT,XO
      CALL NR(F,FP,NUMSIG,MAXIT,XO,X,Y)
      WRITE(6,2)
2     FORMAT(1X,'ENTER 1 TO CONTINUE, 2 TO STOP ')
      READ(5,*)IFLAG
      IF(IFLAG .EQ. 1)GO TO 10
      STOP
      END
C*****FUNCTION F(X):f(x)******
C*****f(x) from the example in Section 2.1C******
      FUNCTION F(X)
      F=EXP(-X)-COS(X)
      RETURN
      END
C*****FUNCTION FP(X):First derivative of f(x) = f'(x)******
      FUNCTION FP(X)
      FP=SIN(X)-EXP(-X)
      RETURN
      END
C*****SUBROUTINE NEWTON RAPHSON******
C*****See Slope Method Algorithm in Section 2.1A******
      SUBROUTINE NR(F,FP,NUMSIG,MAXIT,XO,X,Y)
      RELTOL=10.**(-NUMSIG)
      XPREV=XO
      YPREV=F(XO)
      WRITE(6,6)XPREV,YPREV
6     FORMAT(1X,'K= 0 ' , '   XO=',F12.7,'YO=',E13.7)
      DO 10 K=1,MAXIT
      SLOPE=FP(XPREV)
      DELTAX=-YPREV/SLOPE
      X=XPREV+DELTAX
      Y=F(X)
      WRITE(6,2)K,X,Y,DELTAX
2     FORMAT(1X,'K=',I2,'  X =',F12.7,'  Y =',E13.7,
     t'  DELTAX= ',E14.7)
      XPREV=X
      YPREV=Y
      IF(ABS(DELTAX) .LE. RELTOL*ABS(X))GO TO 9
10    CONTINUE
      WRITE(6,4)MAXIT
4     FORMAT(1X,'DESIRED ACCURACY IS NOT EVIDENT IN',I3,
     + 'ITERATIONS')
      RETURN
9     WRITE(6,3)X,NUMSIG
3     FORMAT(/1X,'X= ',F12.7,' APPROXIMATES REQUIRED ROOT TO ',I2,
     +' SIGNIFICANT DIGITS')
      RETURN
      END
```

ตัวอย่าง output จากการเรียกใช้ Subroutine NEWTON RAPHSON

สำหรับตัวอย่างในหัวข้อ 2.1C

```
ENTER NUMSIG,MAXIT,XO(INITIAL GUESS) UNIT 5? con
7 10 1.3

K= 0  XO=   1.3000000 YO= , 5032927E-02
K= 1  X =   1.2927170  Y =  .1442432E-04 DELTAX=  -.7283263E-02
K= 2  X =   1.2926960  Y =  .2980232E-07 DELTAX=  -.2099426E-04
K= 3  x =   1.2926960  Y =  .2980232E-07 DELTAX=  -.4337729E-07

X=    1.2926960 APPROXIMATES REQUIRED ROOT TO 7 SIGNIFICANT DIGITS
ENTER 3 TO CONTINUE, 2 TO STOP 1

ENTER NUMSIG,MAXIT,XO(INITIAL GUESS) 7 10 4.712389

K= 0 x01-   4.7123870   YO- .8983279E-02
K= 1  X =   4.1212920  Y =  .2663583E-06 DELTAX=   .8903298E-02
K= 2  X =   4.7212930  Y =-.2151355E-06 DELTAX=   .2640180E-06

X=    4.7212930 APPROXIMATES REQUIRED ROOT TO 7 SIGNIFICANT DIGITS
ENTER 1 TO CONTINUE, 2 TO STOP 2
Stop - Program terminated.
```

ตัวอย่างการเรียกใช้ subroutine BAIRSTOW

```
C*****INTERACTIVE CALLING PROGRAM FOR BAIRSTOW'S METHOD******
      DIMENSION   A(10),B(10),C(10)
      MAXIT=20
      WRITE(6,1)
1     FORMAT(1X,'ENTER N(DEGREE OF p(X)) A(1),....,A(N+1)')
      READ(5,*)N,(A(I),I=1,N+1)
      M=N+1
      WRITE it.21
2     FORMAT(1X,'ENTER DESIRED # OF SIGNIFICANT DIGITS')
      READ(5,*)NUMSIG
      WRITE(6,3)
3     FORMAT(1X,'ENTER INITIAL R,INITIAL S')
      READ(5,*)RO,SO
      CALL BAIRSTOW(A,B,C,M,RO,SO,NUMSIG,MAXIT)
      STOP
      END
C*****SUBROUTINE FOR BAIRSTOW'S METHOD*****
C*****See Bairstow's Method Algorithm in Section 2.2C******
      SUBROUTINE BAIRSTOW(A,B,C,M,RO,SO,NUMSIG,MAXIT)
C*****TO FIND A QUADRATIC FACTOR q(x)=x*x-rx-s OF AN Nth DEGREE******
C*****POLYNOMIAL p(x)******
C*****M=N+1******
      DIMENSION A(M),B(M),C(M)
      N=M-1
      B(1)=A(1)
      C(1)=B(1)
      R=RO
      S=SO
      WRITE(6,5)
5     FORMAT(3X,'K',7X,'dr',12X,'ds',14X,'r',12X,'s')
      WRITE(6,6)R,S
6     FORMAT(3X,'0',8X,'-',13X,'-',7X,2F13.7)
      DO 11 K=1,MAXIT
      B(2)=A(2)+B(1)*R
      C(2)=B(2)+C(1)*R
        DO 22 I=3,M
        B(I)=A(I)+R*B(I-1)+S*B(I-2)
        C(I)=B(I)+R*C(I-1)+S*C(I-2)
22      CONTINUE
      DET=C(N-2)*C(N-2)-C(N-1)*C(N-1)
      DR=(B(N)*C(N-1)-B(N+1)*C(N-2))/DET
      DS=(B(N+1)*C(N-1)-B(N)*C(N))/DET
      R=R+DR
      S=S+DS
      WRITE(6,1)K,DR,DS,R,S
1     FORMAT(I4,2E15.7,2F13.7)
      IF(ABS(DR) .LE. 10.**(-NUMSIG)*AMAX1(1.,ABS(R)) .AND.
     +   ABS(DS) .LE. 10.**(-NUMSIG)*AMAX1(1.,ABS(S)))GO TO 9
11    CONTINUE
9     WRITE(6,2)R,S
2     FORMAT(1X,'q(X)=X**X- (',F12.7,')*X- (',F12.7,')')
      WRITE(6,3)(B(I),I=1,N-1)
3     FORMAT(1X,'COEFFICIENTS OF Q(X):'/
     +5X,5F12.7/5X,5F12.7/5X,5F12.7)
      RETURN
      END
```

ตัวอย่างการเรียกใช้ Subroutine LINFIT

```
C*****CALLING PROGRAM FOR LINFIT****
C*****PROBLEM 5.3 OR 205*****
      DIMENSION X(6),Y(6),G(6),H(6)
      REAL LNY(6),LNX(6)
      DATA X,Y/1.,2.,3.,4.,5.,6.,2.3,6.1,10.7,16.0,21.9,28.3/
      OPEN(6,FILE='B:RLINFIT1.OUT',STATUS='NEW')
      N=6
      DO 10 I=1,6
      LNY(I)=ALOG(Y(I))
10    LNX(I)=ALOG(X(I))
      CALL LINFIT(N,X,Y,SQUERR,YCEPT,SLOPE)
      CALL PRINT(N,X,Y,SQUERR,YCEPT,SLOPE)
      CALL LINFIT(N,X,LNY,SQUERR,YCEPT,SLOPE)
      CALL PRINT(N,X,LNY,SQUERR,YCEPT,SLOPE)
      ALPHA=EXP(YCEPT)
      BETA=SLOPE
      WRITE(6,11)ALPHA,BETA
11    FORMAT(1X,'ALPHA = ',F10.5,'  BETA = ',F10.5)
      EG=0
      DO 20 I=1,6
      G(I)=ALPHA*EXP(BETA*X(I))
20    EG=EG+(G(I)-Y(I))**2
      WRITE(6,12)EG
12    FORMAT(1X,'E[g(x)]=',E12.7)
      CALL LINFIT(N,LNX,LNY,SQUERR,YCEPT,SLOPE)
      CALL PRINT(N,LNX,LNY,SQUERR,YCEPT,SLOPE)
      ALPHA=EXP(YCEPT)
      BETA=SLOPE
      WRITE(6,11)ALPHA,BETA
      EH=0
      DO 30 I=1,6
      H(I)=ALPHA*X(I)**BETA
30    EH=EH+(H(I)-Y(I))**2
      WRITE(6,13)EH
13    FORMAT(1X,'E[h(x)]=',E12.7)
      STOP
      END
C*****SUBROUTINE TO PRINT OUTPUT*****
      SUBROUTINE PRINT(N,X,Y,SQUERR,YCEPT,SLOPE)
      DIMENSION X(20),Y(20)
      WRITE(6,2)(X(I),I=1,N)
2     FORMAT(/' X:',6F10.5)
      WRITE(6,3)(Y(I),I=1,N)
3     FORMAT(' Y:',6F10.5)
      WRITE(6,4)SQUERR,YCEPT,SLOPE
4     FORMAT(' ERROR SUM OF SQUARE E[L ] = ',E14.7/
     +       ' Y - INTERCEPT             = ',E14.7/
     +       ' SLOPE                     = ',E14.7)
      WRITE(6,5)YCEPT,SLOPE
5     FORMAT(' PREDICTION EQUATION: Y = ',F8.5,' +',F8.5,' *X   ')
      RETURN
      END
```

ตัวอย่าง Output จากการเรียกใช้ Subroutine LINFIT สำหรับแบบฝึกหัดข้อ 5.3

```
X:    1.00000    2.00000    3.00000    4.00000    5.00000    6.00000
Y:    2.30000    6.10000   10.70000   16.00000   21.90000   28.30000
ERROR SUM OF SQUARE E[L ] =    .3961378E+01
Y - INTERCEPT             =   -.4053333E+01
SLOPE                     =    .5220000E+01
PREDICTION EQUATION: Y = -4.05333 + 5.22000 *X

X:    1.00000    2.00000    3.00000    4.00000    5.00000    6.00000
Y:     .83291    1.80829    2.37024    2.77259    3.08649    3.34286
ERROR SUM OF SQUARE E[L ] =    .2744317E+00
Y - INTERCEPT             =    .6902263E+00
SLOPE                     =    .4796201E+00
PREDICTION EQUATION: Y^ =    .69023 +  .47962 *X
ALPHA =    1.99417  BETA =      .47962
E[g(x)]=.6379251E+02

X:     .00000     .69315    1.09861    1.38629    1.60944    1.79176
Y:     .83291    1.80829    2.37024    2.77259    3.08649    3.34286
ERROR SUM OF SQUARE E[L ] =    .2519730E-04
Y - INTERCEPT             =    .8342940E+00
SLOPE                     =    .1399493E+01
PREDICTION EQUATION: Y^ =    .83429 + 1.39949 *X
ALPHA =    2.30319  BETA =     1.39949
E[h(x)]=.2546936E-02
```

ตัวอย่างการเรียกใช้ FUNCTION POFZ

```
C*****RPOFZ*****
C*****INTERACTIVE CALLING PROGRAM FOR FUNCTION POFZ*****
      DIMENSION X(20).Y(20)
      WRITE(6,1)
1     FORMAT(1X,'ENTER NUMBER OF POINT-; ')
      READ(5,*)NP
      WRITE(6,11)
11    FORMAT(JX,'ENTER X(I)''S 'ı
      READ(5,*)(X(I),I=1,NP)
      WRITE(6,33)
33    FORMAT(1X,'ENTER Y(I)''S ')
      READ(5,*)(Y(I),I=1,NP)
      WRITE(6,6)(X(I),I=1.NP)
6     FORMAT(1X,'X(I): '.10F10.3)
      WRITE(6,7)(Y(I).I=1,NP)
7     FORMAT(1X,'Y(I): ',10F10.3)
12    WR1TE(6,2)
2     FORMAT(1X,'ENTER K,M,Z FOR p(K,K+M) OF Z ')
      READ(5,*)K,M,Z
      YY=POFZ(X,Y,NP,K,M,Z)
      KM=K+M
      WRITE(6,3)K,KM,Z,YY
3     FORMAT(1X,'p(',I2,',',I2,') OF `,F10 3, ''.I 10.3)
      WRITE(o.4)
4     FORMAT(1X,'ENTER 1 TO CONTINUE, 2 TO STOP
      READ(5,*)IFLAG
      IF(IFLAG .EQ. 1)GO TO 12
      STOP
      END
C*****FUNCTION FOR LAGRANGE INTERPOLATION *****
C*****See Lagrange Interpolation Algorithm*****
C*****in Section 6.1C                     *****
      FUNCTION POFZ(X,Y,NP,K,M.Z)
C*****THIS FUNCTION EVALUATES THE LAGRANGE FORM OF*****
C*****P[K,K+M] OF (Z)*****
      DIMENSION X(NP),Y(NP)
      POFZ=0
      DO 10 J=K,K+M
         TERMJ=Y(J)
         DO 11 I=K,K+M
            IF(I .NE. J)THEN
            TERMJ=TERMJ*(Z-X(I))/(X(J)-X(I))
            ENDIF
11       CONTINUE
         POFZ=POFZ+TERMJ
10    CONTINUE
      RETURN
      END
```

ตัวอย่าง Output จากการเรียกใช้ Subroutine POFZ

สำหรับตัวอย่างในหัวข้อ 6.1D

```
UNIT 5? con

ENTER NUMBER OF POINTS UNIT 5? con
5

ENTER X(I)'S -2 0 1 4 5

ENTER Y(I)'S -8 0 1 64 125

X(I):      -2.000       .000      1.000      4.000      5.000
Y(I):      -8.000       .000      1.000     64.000    125.000
ENTER K,M,Z FOR p(K,K+M) OF Z 2 2 3

p( 2, 4) OF       3.000=     33.000
ENTER 1 TO CONTINUE, 2 TO STOP. 1

ENTER K,M,Z FOR p(K,K+M) OF Z 1 3 2

p( 1, 4) OF       2.000=      8.000
ENTER 1 TO CONTINUE, 2 TO STOP. 1

ENTER K,M,Z FOR p(K,K+M) OF Z 1 4 2

p( 1, 5) OF       2.000=      8.000
ENTER 1 TO CONTINUE, 2 TO STOP.
```

ตัวอย่างการเรียกใช้ Subroutine FORMDD

```
C*****RFORMDD*****
C*****CALLING PROGRAM FOR SUBROUTINE F ORMDD*****
      DIMENSION  X(10),Y(10),DD(10,10)
      WRITE(6,1)
1     FORMAT(1X,'ENTER NUMBER OF POINTS (LESS THAN 10) ')
      READ(5,*)NP
      WRITE(6,2)
2     FORMAT(1X,'ENTER X''S : ')
      READ(5,*)(X(I),I=1,NP)
      WRITE(6,3)
3     FORMAT(1X,'ENTER Y''S : ')
      READ(5,*)(Y(I),I=1,NP)
      WRITE(6,4)(X(I),I=1,NP)
4     FORMAT(1X,'X(I): ',5F10.3)
      WRITE(6,5)(Y(I),I=1,NP)
5     FORMAT(1X,'Y(I): ',5F10.3)
      CALL FORMDD(X,Y,NP,DD)
      STOP
      END
C*****FORMING A DIVIDED DIFFERENCE TABI E DD******
C*****See FORMDD ALGORITHM in Section 6.2E******
      SUBROUTINE FORMDD(X,Y,N,DD)
      DIMENSION  X(N),Y(N),DD(N,N)
      DO 10 K-l.N
10       DD(K,1)=Y(K)
      DO 12 M=1,N-1
         DO 1.3 K= 1 ,N-M
         DD(K,M+1)=(DD(K+1,M)-DD(K,M))/(X(K+M) X(K )
13       WRITE(6,1)K,M+1,DD(K,M+1),DD(K,M+1)
1        FORMAT(1X,'DD(',I2,',',I2,') =',F12.7,E14. 'I
12    CONTINUE
      RETURN
      END
```

ตัวอย่าง Output จากการเรียกใช้ Subroutine **FORMDD**

```
ENTER NUMBER OF POINTS (LESS THAN 10) UNIT 5? con
6

ENTER X'S : 0 .2 .4 .6 .8 1.0

ENTER Y'S : .5 .5793 .6554 .7257 .7881 .8413

X(I):        .000       .200       .400       .600       .800
X(I):       1.000
Y(I):         500       .579       .655       .726       788
Y(I):         841
DD( 1, 2) =     .3964999   .3964999E too
DD( 2, 2) =     . 3805000  .3805000E+00
DD( 3, 2) =     .3515002   .3515002E+00
DD( 4, 2) =     .3119999   .3119999E+00
DD( 5, 2) =     .2660001   .2660001E+00
DD( 1, 3) =    -. 0399999 -.3999993E -01
DD( 2, 3) =    -.0724994  -.7249943E-01
DD( 3, 3) =    -.0987506  -.9875059E-01
DD( 4, 3) =    -.1149997  -.1149997E+00
DD( 1, 4) =    -.0541658  -.5416583E-01
DD( 2, 4) =    -.0437519  -.4375193E-01
DD( 3, 4) =    -.0270819  -.2708188E-01
DD( 1, 5) =     .0130174   .1301737E-01
DD( 2, 5) =     .0208376   .2083757E-01
DD( 1, 6) =     .0078202   .7820197E-02Stop - Program terminated.
```

บทที่ 7

ตัวอย่างการเรียกใช้ Subroutine **ROMBERG**

```
C*****RROMBERG*****
C*****CALLING PROGRAM FOR SUBROUTINE ROMBERG*****
      DIMENSION X(100),Y(100),T(20,20)
      EXTERNAL FUNC
5     WRITE(6,1)
1     FORMAT(1X,'ENTER a,b,N,MAXROWS,NUMSIG ')
      READ(5,*)A,B,N,MAXROWS,NUMSIG
      CALL ROMBERG(A,B,N,MAXROWS,NUMSIG,FUNC,X,Y,T)
      WRITE(6,2)
2     FORMAT(1X,'ENTER 1 TO CONTINUE,2 TO STOP ')
      READ(5,*)IFLAG
      IF(IFLAG .EQ. 1)GO TO 5
      STOP
      END
C*****FUNCTION TO BE INTEGRATED******
      FUNCTION FUNC(X)
C*****THIS FUNCTION IS TO BE CHANGED******
      FUNC=ALOG(X)
      RETURN
      END
C*****ROMBERG INTEGRATION******
C*****See ROMBERG INTEGRATION ALGORITHM in Section 7.40******
      SUBROUTINE ROMBERG(A,B,N,MAXROWS,NUMSIG,FUNC,X,Y,T)
      DIMENSION X(100),Y(100),T(MAXROWS,MAXROWS)
      EXTERNAL FUNC
      H=(B-A)/N
      CULL SUMOE( X,Y,N,A,B,H,SUMO, SUME,FUNC)
      SUM=SUMO+SUME
      T(1,1)=H*((FUNC(A)+FUNC(B))/2.+SUM)
      WRITE(6,9)T(1,1)
9     FORMAT(1X,'T( 1, 1)= ' F12.6)
      RELTOL=10**(-NUMSIG)
      D0 19 K=2, MAX R 0 W S
      H=H/2
      N=2*N
      CALL SUMOE(X,Y,N,A,B,H,SUMO,SUME,FUNC)
      T(K,1)=.5*(T(K 1,1)+2*H*SUMO)
      WRITE(6,7)K,T(K,1)
7     FORMAT(1X,'T(',I2, , 1)=',F12.6)
      00 18 I=2,K
      T(K,I)=(4**(I-1)*T(K,I-1) T(K-1,I 1))/(4**(I-1)-1)
18    WRITE(6,8)K,I,T(K,I)
8     FORMAT(1X,'T(',I2,',',I2,')= ',F12.6)
      IF(ABS(T(K,K)-T(K,K-1)) .LE. RELTOL*ABS(T(K,K)))GO TO 99
19    CONT INUE
      WRITE(6,2)MAXROWS
2     FORMAT(1X,I2,' ITERATIONS DID NOT YIELD THE DESIRED ',
     +'ACCURACY.')
      RETURN
99    WRITE(6,3)K,K,T(K,K)
3     FORMAT(1X,'ESTIMATED INTEGRAL IS T(',I2,',',I2,')',F12.6)
      RETURN
      END
```

```
C*****SUMOE******
C*****Points in ROMBERG ALGORITHM:y(0),y(1),y(2),...,y(n-1),y(n)*****
C*****SUMO = y(1) t y(3) t ...
C*****SUME = y(2) t y(4) t ...
C*****Points in this subroutine :y(1),y(2),y(3),..y(n) ,y(n+1)*****
      SUBROUTINE SUMOE(X,Y,N,A,B,H,SUMO,SUME,FUNC)
      DIMENSION X(N),Y(N)
      SUMO=0
      SUME=0
      DO 10 I=2,N
      X(I)=A+(I-1)*H
10    Y(I)=FUNC(X(I))
      D O 11 I=2,N,2
11    SUMO=SUMO+Y(I)
      DO 12 I=3,N,2
12    SUME=SUME+Y(I)
      RE TURN
      END
```

ตัวอย่าง Output จากการเรียกใช้ Subroutine **ROMBERG**

```
C:\FORTRAN>rromberg
File name missing or blank - Please enter name
UNIT 6? con

ENTER a,b,N,MAXROWS,NUMSIG UNIT 5? con
1 2.2 3 5 5

T( 1, 1)=       .527395
T( 2, 1)=      .532792
T( 2, 2)=       .534591
T( 3, 1)=      .534152
T( 3, 2)=       .534605
T( 3, 3)=       .534606
T( 4, 1)=      .534493
T( 4, 2)=       .534606
T( 4, 3)=       .534606
T( 4, 4)=       .534606
ESTIMATED INTEGRAL IS T( 4, 4)      .534606
ENTER 1 TO CONTINUE,2 TO STOP
```

```
C*****PR713A*****
C*****Problem 7.13a*****
C*****3-point GAUSS QUADRATURE for integral of SQRT(X)*****
C*****over [0,4]*****
      DIMENSION XI(3),X(3),GAMMA(3)
      FUNC(X)=SQRT(X)
      DATA A,B/0.0,4.0/
C*****GAUSS SAMPLE POINTS (XI's)*****
C*****and GAUSS SAMPLE WEIGHTS (GAMMA's)*****
C*****from TABLE 7.5-1*****
      XI(1)=SQRT(.6)
      XI(2)=-XI(1)
      XI(3)=0
      GAMMA(1)=5./9.
      GAMMA(2)=GAMMA(1)
      GAMMA(3)=8./9.
      OPEN(6,FILE='B:PR713A.OUT',STATUS='NEW')
      WRITE(6,5)
5     FORMAT(1X,'Problem 7.13a, 3-Point Gauss Quadrature'/)
      DO 11 I=1,3
11    X(I)=A+(B-A)/2*(XI(I)+1)
      PINT-0
      DO 10 I=1,3
      Y=FUNC(X(I))
      Z=GAMMA(I)*Y
      WRITE(6,1)I,X(I),Y,Z
1     FORMAT(1X,'X(',I1,')=',F12.7,' F(X)=',F12.7,
     +' GAMMA(I)*F(X(I))=',F12.7)
10    PINT=PINT+Z
      PINT=(B-A)/2.*PINT
      WRITE(6,2)PINT
2     FORMAT(/1X,'ESTIMATED DEFINITE INTEGRAL = ',F12.7)
      STOP
      END
```

```
 Problem 7.13a, 3-Point Gauss Quadrature

X(1)=   3.5471530  F(X)=   1.8839300  GAMMA(I)*F(X(I))=   1.0466280
X(,2)-    .4508066 F(X)=    .6714213  GAMMA(I)*F(X(I))=   .3730119
X(3)=   2.0000000  F(X)=   1.4142140  GAMMA(I)*F(X(I))=   1.2570790

ESTIMATED DEFINITE INTEGRAL =    5.3534370
```

```
C*****PR713B*****
C*****Problem 7.13b*****
C*****5-point GAUSS QUADRATURE for INTEGRAL of SQRT(X)*****
C*****over [0,4]*****
      DIMENSION XI(5),X(5),GAMMA(5)
      FUNC(X)=SQRT(X)
      DATA A,B/0.0,4.0/
C*****GAUSS SAMPLE POINTS (XI's)*****
C*****and GAUSS SAMPLE WEIGHTS (GAMMA's)*****
C*****from TABLE 7.5-1*****
      DATA XI/-.9061798,.9061798,-.5384693,.5384693,0./
      DATA GAMMA/.2369269,.2369269,.4786287,.4786287,.5688889/
      OPEN(6,FILE='B:PR713B.OUT',STATUS='NEW')
      WRITE(6,5)
5     FORMAT(1X,'Problem 7.13b, 5-Point Gauss Quadrature'/)
      DO 11 I=1,5
11    X(I)=A+(B-A)/2*(XI(I)+1)
      PINT=0
      DO 10 I=1,5
      Y=FUNC(X(I))
      Z=GAMMA(I)*Y
      WRITE(6,1)I,X(I),Y,Z
1     FORMAT(1X,'X(',I1,')=',F12.7,' F(X)=',F12.7,
     t' GAMMA(I)*F(X(I))=',F12.7)
10    PINT=PINT+Z
      PINT=(B-A)/2.*PINT
      WRITE(6,2)PINT
2     FORMAT(/1X,'ESTIMATED DEFINITE INTEGRAL = ',F12.7)
      STOP
      END
```

```
 Problem 7.13b, 5-Point Gauss Quadrature

 X(1)=     .1876404 F(X)=     .4331748 GAMMA(I)*F(X(I))=     .1026308
 X(2)=    3.8123600 F(X)=    1.9525260 GAMMA(I)*F(X(I))=     .4626060
 X(3)=     .9230614 F(X)=     .9607608 GAMMA(I)*F(X(I))=     .4598477
 X(4)=    3.0769390 F(X)=    1.7541200 GAMMA(I)*F(X(I))=     .8395724
 X(5)=    2.0000000 F(X)=    1.4142140 GAMMA(I)*F(X(I))=     .8045304

 ESTIMATED DEFINITE INTEGRAL =     5.3383750
```

บทที่ 8

ตัวอย่างการเรียกใช้ Subroutine RK4

```
C*****RRK4******
C*****CALLING PROGRAM FOR SUBROUTINE RK4******
C*****RK4 : 4-TH ORDER RUNGE-KUTTA METHOD******
      DOUBLE PRECISION YO
      EXTERNAL FTEST
      WRITE(6,1)
1     FORMAT(1X,'INPUT TO,TF,YO ')
      READ(5,*)TO,TF,YO
      WRITE(6,2)
2     FORMAT(1X,'INPUT # STEPS NPRINT ')
      READ(5,*)NSTEPS,NPRINT
      WRITE(6,3)
3     FORMAT(6X,'T',12X,'Y')
      CALL RK4(FTEST,TO,TF,NSTEPS,NPRINT,YO)
      STOP
      END
C*****FUNCTIDN FTEST******
C*****This FUNCTION is to be changed upon IVP*****
      FUNCTION FTEST(T,Y)
      DOUBLE PRECISION Y
C*****Example in Section 8.2C*****
         FTEST=-T*Y*Y
      RETURN
      END
C*****SUBROUTINE RK4******
      SUBROUTINE RK4(F,T,TF,NSTEPS,NPRINT,Y)
      REAL M1,M2,M3,M4
      DOUBLE PRECISION Y
C*****************************************************************
C     THIS SUBROUTINE INTEGRATES FROM TO TO TF THE 1ST ORDER IVP
C          Y' = F(T,Y)     Y(TO)=YO (INITIAL T,Y)
C     USING NSTEPS STEPS OF THE 4TH ORDER RUNGE-KUTTA METHOD.
C     IF NPRINT>0, IT PRINTS T AND Y EVERY NPRINT STEPS.
C     NOTE: F MUST BE DECLARED EXTERNALLY IN THE CALLING PROGRAM
C*****************************************************************
      IF (NPRINT .GT. 0)WRITE(6,1)T,Y
1     FORMAT(F10.3,3X,D14.7)
      H=(TF-T)/NSTEPS
      DO 10 J=1,NSTEPS
         M1=F(T,Y)
         M2=F(T+0.5*H,Y+0.5*H*M1)
         M3=F(T+0.5*H,Y+0.5*H*M2)
         M4=F(T+H,Y+H*M3)
         T=T+H
         Y=Y+H*(M1+2*(M2+M3)+M4)/6
         IF(NPRINT .GT. 0 .AND. MOD(J,NPRINT) .EQ. 0)
     t      WRITE(6,1)T,Y
10    CONTINUE
      RETURN
      END
```

```
C*****RRK43******
C*****CALLING PROGRAM FOR SUBROUTINE RK4*****
C*****RK4 : 4-TH ORDER RUNGE-KUTTA METHOD******
      DOUBLE PRECISION YO
      EXTERNAL FTEST
      WRITE(6,1)
1     FORMAT(1X,'INPUT TO,TF,YO ')
      READ(5,*)TO,TF,YO
      WRITE(6,2)
2     FORMAT(1X,'INPUT # STEPS NPRINT ')
      READ(5,*)NSTEPS,NPRINT
      WRITE(6,5)
5     FORMAT(7X,'Output for Problem 8.6b')
      WRITE(6,3)
3     FORMAT(7X,'T',12X,'Y')
      CALL RK4(FTEST,TO,TF,NSTEPS,NPRINT,YO)
      STOP
      END
C*****FUNCTION FTEST******
C*****This FUNCTION is to be changed upon IVP*****
      FUNCTION FTEST(T,Y)
      DOUBLE PRECISION Y
C*****IVP from Problem 8.6b*****
        FTEST=.5*(Y/T+T/Y)
      RETURN
      END
******SUBROUTINE RK4******
      SUBROUTINE RK4(F,T,TF,NSTEPS,NPRINT,Y)
      REAL M1,M2,M3,M4
      DOUBLE PRECISION Y
C**********************************************************************
C     THIS SUBROUTINE INTEGRATES FROM TO TO TF THE 1ST ORDER IVP
C          Y' = F(T,Y)    Y(TO)=YO (INITIAL T,Y)
C     USING NSTEPS STEPS OF THE 4TH ORDER RUNGE-KUTTA METHOD.
C     IF NPRINT>0, IT PRINTS T AND Y EVERY NPRINT STEPS.
C     NOTE: F MUST BE DECLARED EXTERNALLY IN THE CALLING PROGRAM
C**********************************************************************
      IF (NPRINT .GT. 0)WRITE(6,1)T,Y
1     FORMAT(F10.3,3X,D14.7)
      H=(TF-T)/NSTEPS
      DO 10 J=1,NSTEPS
         M1=F(T,Y)
         M2=F(T+0.5*H,Y+0.5*H*M1)
         M3=F(T+0.5*H,Y+0.5*H*M2)
         M4=F(T+H,Y+H*M3)
         T=T+H
         Y=Y+H*(M1+2*(M2+M3)+M4)/6
         IF(NPRINT .GT. 0 .AND. MOD(J,NPRINT) .EQ. 0)
     +     WRITE(6,1)T,Y
10    CONTINUE
      RETURN
      END
```

ตัวอย่าง Output จากการเรียกใช้ Subroutine RK4

สำหรับ IVP จากตัวอย่างในหัวข้อ 8 . x

```
C:\FORTRAN>rrk4
File name missing or blank - Please enter name
UNIT 6? con

INPUT TO,TF,YO UNIT 5? con
2 3 1

INPUT # STEPS NPRINT 10 1

     T              Y
 2.000       .1000000D+01
 2.100       .8298852D+00
 2.200       .7042368D+00
 2.300       .6079135D+00
 2.400       .5319244D+00
 2.500       .4705964D+00
 2.600       .4201750D+00
 2.700       .3780778D+00
 2.800       .3424708D+00
 2.900       .3120168D+00
 3.000       .2857180D+00Stop    Program   terminated.
```

ตัวอย่าง Output จากการเรียกใช้ Subroutine RK4

สำหรับ IVP จากแบบฝึกหัดข้อ 8.6b

```
C:\FORTRAN>rrk43
File name missing or blank - Please enter name
UNIT 6'1' con

INPUT TO,TF,YO UNIT 5? con
1 3 3

INPUT # STEPS NPRINT 10 1

      Output for Problem 8.6b
      T              Y
 1.000       .3000000D+01
 1.200       .3322651D+01
 1.400       .3627674D+01
 1.600       .3919187D+01
 1.800       .4200004D+01
 2.000       .4472140D+01
 2.200       .4737092D+01
 2.400       .4996003D+01
 2.600       .5249767D+01
 2.800       .5499096D+01
 3 . 000     .5744568D+01Stop   Program terminated.
```

ตัวอย่างการเรียกใช้ Subroutine RKF4

```
C*****RRKF4******
C*****CALLING PROGRAM FOR SUBROUTINE RKF4*****
C*****RKF4 : 4 TH ORDER RUNGE-KUTTA-FEHLBERG METHOD******
      DOUBLE PRECISION YO(10)
      COMMON N, TO, TF, YO
      DATA SCAMIN, SCAMAX, RMAX/0.1, 4.0, 1.E-4/
      WRITE(6,1)
1     FORMAT(1X,'INPUT N, TO ,TF ,NPRINT ')
      READ(5,*)N, TO, TF, NPRINT
      WRITE(6,2)N
2     FORMAT(1X,'INPUT',I2,' COMPONENTS OF YO ')
      READ(5,*)(YO(I),I=1,N)
      WRITE(6,3)TO, TF, NPRINT,
3     FORMAT(1X,'INTEGRATING FROM',F8.4,' TO',F8.4,
     +          ', PRINTING EVERY',I5,' STEPS'//5X,
     +          'I',6X,'T',10X,'H',8X,'------Y------>')
      CALL RKF4(RMAX, SCAMIN, SCAMAX, NPRINT)
      STOP
      END
C*****SUBROUTINE EVALF*****
      SUBROUTINE EVALF(T,Y,F)
      DIMENSION F(1)
      DOUBLE PRECISION Y(1)
C*****F(1) is to be changed upon IVP*****
C*****Example in Section 8.2C*****
      F(1) = -T*Y(1)*Y(1)
*
      RETURN
      END
C*****SUBROUTINE SUMK*****
      SUBROUTINE SUMK(J, F, Q1, Q2, Q3, Q4, Q5)
      REAL K(10,6), F(10)
      DOUBLE PRECISION Y(10), SUM(10)
      COMMON N, T, TF, Y
      COMMON /KCALC/ H, K
C*****************************************************************
C     THIS SUBROUTINE EVALUATES THE N-VECTOR
C
C          SUM = Y + Q1*ROW1(K) +...+ Q5*ROW5(K)
C
C     IF J = 0, IT SETS Y = SUM.
C     IF 1 <= J <= 6, IT PUTS F(T+P*H,SUM) IN JTH COLUMN OF K
C*****************************************************************
      DO 10 I = 1, N
         SUM(I) = Y(I) + DBLE(Q1)*K(I,1) + DBLE(Q2)*K(I,2) +
     +            DBLE(Q3)*K(I,3) + DBLE(Q4)*K(I,4) + DBLE(Q5)*K(I,5)
         IF(J .EQ. 0)Y(I) = SUM(I)
   10 CONTINUE
      IF(J .EQ. 0)RETURN
      CALL EVALF(T+P*H, SUM, F)
      DO 20 I = 1,N
         K(I,J) = H*F(I)
   20 CONTINUE
      RETURN
      END
```

```
C*****SUBROUTINE RKF4******
      SUBROUTINE RKF4(RMAX,SCAMIN,SCAMAX,NPRINT)
      REAL K(10,6)
      DOUBLE PRECISION Y(10)
      COMMON N, T, TF, Y
      COMMON /KCALC/ H, K
C*************************************************************************
C     THIS SUBROUTINE USES THE 4TH ORDER RUNGE-KUTTA-FEHLBERG
C     ALGORITHM TO INTEGRATE N (N < 11) COUPLED FIRST ORDER IVPS.
C     I.E.      Y' = F(T,Y)   Y(TO) = YO  (TO, YO ARE INITIAL T, Y)
C     FROM TO TO TF, WHERE TF CAN BE TO THE LEFT FO TO.
C     VALUES OF T, H, AND VECTOR Y ARE PRINTED EVERY NPRINT ITERATIONS.
C     IF NPRINT = 0, NOTHING IS PRINTED (UNLESS H BECOMES TOO SMALL).
C*************************************************************************
C     INITIALIZE STEPSIZE, MINIMUM ALLOWABLE STEPSIZE, AND COUNTER:
      H = SIGN(RMAX**.25,TF-T)
      HMIN=0.5E-4*H
      ITER=0
      IF(NPRINT .GT. 0) WRITE(6,2)ITER,T,H,(Y(I),I=1,N)
2     FORMAT(I6,F9.4,11E15.7)

C     ITERATE:
C*****BEGIN LOOP BY SETTING H TO TF-T IF T+H PASSES TF
10    IF(H*(T+H-TF) .GT. 0)H=TF-T
C
C*****PUT VECTOR KJ IN JTH COLUMN OF MATRIX K, J=1,....6
      CALL SUMK(1,0.,0.,0.,0.,0.,0.)
      CALL SUMK(2,.25, .25, 0., 0., 0., 0.)
      CALL SUMK(3,.375, 3./32, 9./32, 0., 0., 0.)
      CALL SUMK(4,12./13, 1932./2197, -7200./2197, 7296./2197, 0., 0.)
      CALL SUMK(5,1., 439./216, -8., 3680./513, -845./4104, 0.)
      CALL SUMK(6,.5,-8./27, 2., -3544./2565, 1859./4104, -11./40)
C
C*****FORM ERREST = ESTIMATE OF ERROR OF NEXT Y(I) FOR I = 1,....N
C*****AND FIND RATIO = THE LARGEST OF THE ERREST/H RATIOS
      RATIO = 0.
      DO 20 I=1,N
      ERREST=K(I,1)/360-128.*K(I,3)/4275-2197.*K(I,4)/75240+
     +   K(I,5)/50+2.*K(I,6)/55
      RATIO=AMAX1(RATIO,ABS(ERREST/H))
20    CONTINUE
C
C*****TEST ACCURACY OF NEXT Y. IF OK, UPDATE T, Y AND ITER
      IF(RATIO .GT. RMAX) GO TO 30
        T=T+H
        CALL SUMK(0, 0., 25./216, 0., 1408./2565, 2197./4104, -0.2)
        ITER=ITER+1
C
        IF(NPRINT .GT. 0 .AND. (MOD(ITER,NPRINT) .EQ. 0.OR.T.EQ.TF))
     +    WRITE(6,2)ITER,T,H,(Y(I),I=1,N)
```

```
C
C*****SET SCALE(BETWEEN SCAMIN AND SCAMAX) AND UPDATE H
30     SCALE=.84*(RMAX/RATIO)**.25
       IF(SCALE .LT. SCAMIN)SCALE=SCAMIN
       IF(SCALE .GT. SCAMAX)SCALE=SCAMAX
       H=SCALE*H
C
C*****TERMINATION TESTS
       IF(T .EQ. TF)RETURN
       IF(ABS(H) .GT. HHIN)GO TO 10
C*****END OF LOOP
C
       WRITE(6,3)T
3      FORMAT(1X,'APPARENT SINGURARITY NEAR T= ',E10.3)
       RETURN
       END
```

ตัวอย่าง Output จากการเรียกใช้ Subroutine **RKF4**

```
C:\FORTRAN>rrkf4
File name missing or blank - Please enter name
UNIT 6? con

INPUT N, TO ,TF ,NPRINT UNIT 5? con
1 2 4 1

INPUT 1 COMPONENTS OF YO 1

INTEGRATING FROM 2.0000 TO 4.0000, PRINTING EVERY     1 STEPS

    3       T           H          -----Y----->
    0    2.0000     .1000000E+00    .1000000E+01
    1    2.1000     .1000000E+00    .8298735E+00
    2    2.2115     .1114867E+00    .6918773E+00
    3    2.3496     .1381161E+00    .5680757E+00
    4    2.5204     .1708163E+00    .4595003E+00
    5    2.7342     .2137788E+00    .3652362E+00
    6    3.0050     .2707816E+00    .2844945E+00
    7    3.3529     .3478962E+00    .2164052E+00
    8    3.8077     .4547834E+00    .1600214E+00
    9    4.0000     .1923409E+00    .1428565E+00Stop - Program terminated.
```

บทที่ 9

ตัวอย่างการเรียกใช้ Subroutine ROTATE

```
C*****RJACOBI
C*****CALLING PROGRAM FOR SUBROUTINE ROTATE*****
C*****SEE JACOBI'S METHOD ALGORITHM        *****
C*****IN SECTION 9.3C                      *****
      DIMENSION A(5,5),U(5,5)
      COMMON  N,NROTAT,A,U
      OPEN(6,FILE='B:JACOBI.OUT',STATUS='NEW')
      WRITE(7,1)
1     FORMAT(1X,'ENTER N(LESS THAN 5):',
     t ' ORDER OF SYMMETRIC MATRIX A')
      READ(5,*)N
      WRITE(6,51)N
51    FORMAT(1X,'N = ',I3)
      WRITE(7,2)
2     FORMAT(1X,'ENTER N*N ELEMENTS OF A:',
     t ' COLUMN BY COLUMN')
      READ(5,*)((A(I,J),I=1,N),J=1,N)
      WRITE(6,22)((A(I,J),I=1,N),J=1,N)
22    FORMAT(/1X,'A(I,J) BY COLUMN:',3F12.7,(/18X,3F12.7))
      WRITE(7,3)
3     FORMAT(1X,'ENTER NUMDEC, MAXIT ')
      READ(5,*)NUMDEC,MAXIT
      WRITE(6,77)NUMDEC,MAXIT
77    FORMAT(/1X,'NUMDEC = ',I3,' ,MAXIT = ',I3/)
      DO 10 I = 1,N
      DO 10 3 = l,N
      IF(I .EQ. J)THEN
        U(I,J) = 1
      ELSE
        U(I,J) = 0
      ENDIF
10    CONTINUE
      ABSTOL = 10.0**(-NUMDEC)
      WRITE(6,88)ABSTOL
88    FORMAT(1X,'ABSTOL = ',F12.7/)
      NROTAT = 0
      DO 11 K = l,MAXIT
         AMAXOD = ABS(A(1,2))
         II = 1
         JJ = 2
         DO 12 I = 1,N-1
         DO 12 J = 2,N
         IF(I .LT. J)THEN
           IF(ABS(A(I,J)) .GT. AMAXOD)THEN
             AMAXOD = ABS(A(I,J))
             II = I
             JJ = J
           ENDIF
         ENDIF
12       CONTINUE
      WRITE(6,66)
66    FORMAT(/1X,' K II JJ      A(II,JJ)      AMAXOD')
      WRITE(6,6)K,II,JJ,A(II,JJ),AMAXOD
6     FORMAT(/3I3,2F12.7)
```

```
      CALL ROTATE(II,JJ)
      WRITE(6,67)((A(I,J),I=1,N),J=1,N)
67    FORMAT(/1X,'A(I,J) BY COLUMN:',3F12.7,(/18X,3F12.7))
      WRITE(6,68)((U(I,J),I=1,N),J=1,N)
68    FORMAT(/1X,'U(I,J) BY COLUMN:',3F12.7,(/18X,3F12.7))
      IF(AMAXOD .LT. ABSTOL)GO TO 99
11    CONTINUE
      WRITE(6,4)MAXIT
4     FORMAT(/1X,'DESIRED ACCURACY IS NOT EVIDENT IN',
     +I3,' ITERATIONS')
      STOP
99    WRITE(6,7)NROTAT
7     FORMAT(/1X,'NO. OF ROTATIONS PERFORMED =',I3)
      DO 13 J = l,N
      WRITE(6,5)A(J,J),(U(I,J),I=1,N)
5     FORMAT(/1X,'EIGEN VALUE',F12.7/
     t       1X,'EIGEN VECTOR:',5F12.7/)
13    CONTINUE
      STOP
      END
C*****SUBROUTINE ROTATE*****
      SUBROUTINE ROTATE(I,J)
      DIMENSION A(5,5), U(5,5)
      COMMON N,NROTAT,A,U
C*******************************************************
C THIS SUBROUTINE PERFORMS THE ROTATION REPLACEMENTS
C       A <-- RTRANSPOSE*A*R   AND U <-- U*R
C WHERE R = R[I,J](THETA) MAKES A(1,J) = A(J,I) = 0
C*******************************************************
      THETA = ATAN(l.O)
      IF(ABS(A(I,I)-A(J,J)) .GT. 1.E-6*ABS(A(I,I)))
     t THETA = .5*ATAN(2.*A(I,J)/(A(I,I)-A(J,J)))
      SINT = SIN(THETA)
      COST = COS(THETA)
C*****ROTATE:   A <-- RTRANS*A*R AND U <-- U*R
      DO 10 K = l,N
         UKI = U(K,I)
         U(K,I) = UKI*COST t U(K,J)*SINT
         U(K,J) =-UKI*SINT t U(K,J)*COST
         IF(K .EQ. I .OR. K .EQ. J)GO TO 10
           A(K,I) = A(I,K)*COST t A(J,K)*SINT
           A(K,J) =-A(I,K)*SINT + A(J,K)*COST
           A(J,K) = A(K,J)
           A(I,K) = A(K,I)
10    CONTINUE
      AII = A(I,I)
      A(I,I) = AII*COST**2 t A(J,J)*SINT**2 t
     +2.*A(I,J)*SINT*COST
      A(J,J) = AII*SINT**2 t A(J,J)*COST**2 -
     +2.*A(I,J)*SINT*COST
      A(I,J) = 0.
      A(J,I) = 0.
      NROTAT = NROTAT+1
      RETURN
      END
```

ตัวอย่าง Output จากการเรียกใช้ Subroutine ROTATE

สำหรับ JACOBI'S METHOD ALGORITHM

```
N=    3

A(I,J) BY COLUMN:    3.0000000     .0100000     .0200000
                      .0100000    2.0000000     .1000000

                      .0200000     .1000000    1.0000000

NUMDEC =    7 ,MAXIT =  10

ABSTOL =      .0000001


 K II JJ      A(II,JJ)      AMAXOD

 1  2  3     .1000000     .1000000

A(I,J) BY COLUMN:    3.0000000  .0119221     .0189173
                      .0119221  2.0099020     .0000000

                      .0189173     .0000000     .9900981

U(I,J) BY COLUMN:    1.0000000     .0000000     .0000000
                      .0000000     .9951333     .0985376

                      .0000000    -.0985376     .9951333

 K II JJ      A(II,JJ)      AMAXOD

 2    13     .0189173     .0189173

A(I,J) BY COLUMN:    3.0001780   .0119216     .0000000
                      .0119216  2.0099020    -.0001122

                      .0000000  -.0001122     .9899200

U(I,J) BY COLUMN:  , 9999557  -.0009273     .0093650
                      .0000000  .9951333     .0985376

                     -.0094108   -.0985333     .9950893

 K II JJ      A(II,JJ)      AMAXOD

 3  1  2     .0119216     .0119216

A(I,J) BY COLUMN:    3.0003220  ,  0000000    -.0000014
                      .0000000    2.0097580    -.0001122

                     -.0000014   -.0001122     .9899200

U(I,J) BY COLUMN:    .9998833  .0110502     .0105503
                     -.0120355     .9950724     .0984178

                     -.0094108   -.0985333     .9950893
```

```
 K II JJ      A(II,JJ)       AMAXOD

 4  2  3   -.0001122     .0001122

A(I,J) BY COLUMN:    3.0003220       .0000000     -.0000014
                       .0000000     2.0097580       .0000000

                                   -.0000014      .0000000     .9899200

U(I,J) BY COLUMN:      .9998833     .0110502     .0105503
                      -60120344     .9950833     .0983083

                                   -.0094121    -.0984238     .9951001

 K II JJ      A(II,JJ)       AMAXOD

 5  1  3   -.0000014     .0000014

A(I,J) BY COLUMN:    3.0003220       .0000000     , 0000000
                       .0000000     2.0097580      .0000000

                                    .0000000      , 0000000    .9899200

U(I,J) BY COLUMN:      .9998833     .0110502     .0105497
                      -.0120344     .9950833     .0983083

                                   -.0094114    -.0984238     .9951001

 K II JJ      A(II,JJ)       AMAXOD

 6  1  2    , 0000000     .0000000

A(I,J) BY COLUMN:    3.0003220       .0000000      .0000000
                       .0000000     2.0097580      .0000000

                                    .0000000      .0000000     .9899200

U(I,J) BY COLUMN:      .9998833     .0110502     .0105497
                      -.0120344     .9950833     .0983083

                                   -.0094114    -.0984238     .9951001

NO.  OF  ROTATIONS  PERFORMED  =  6

EIGEN  VALUE    3.0003220
EIGEN  VECTOR:     .9998833     .0110502     .0105497

EIGEN  VALUE    2.0097580
EIGEN  VECTOR:    -.0120344     .9950833     .0983083

EIGEN  VALUE     .9899200
EIGEN VECTOR:     -.0094114   -.0984238     .9951001
```