```
C*********************************************************
      PROGRAM RIEMANN
C     this is the PHASE ONE algorithm
C     for holomorphic cusp-forms (March 1994)
C     CRAY VERSION -- SINGLE PRECISION
C     for arbitrary Hecke groups:  G(NGRP)
C*********************************************************
C     Prelim. version by:   Ze Wang (wangz@msi.umn.edu)
C     Advised by:           Prof. Dennis A. Hejhal
C     05/1994, D. Hejhal, Z. Wang, University of Minnesota
C*********************************************************
C
      IMPLICIT REAL(A-B,D-H,P-Y)
      IMPLICIT COMPLEX(C,Z)
      PARAMETER(NMAX=100,NPAX=50,NCAX=50)
C     make certain that NMAX=2*NPAX=2*NCAX for later use!
C     here: NPAX = an integer greater than the largest M0-value
      PARAMETER(IFT=200)
      PARAMETER(IFT2=2*IFT)
      PARAMETER(IFT4=4*IFT)
C     make certain that IFT > largest effective M-value for
C     levels Y11 and Y22 (not merely for Y-min of polygon)
      PARAMETER(NTRIM=1)
      DIMENSION ZA(IFT4), ZP(IFT4)
C     ZP will correspond to fundamental polygon pullbacks
      DIMENSION R(2,2), RINV(2,2), REC(IFT4,5)
      DIMENSION ITRIM(NTRIM)
      DIMENSION ZU(NMAX,NPAX)
      DIMENSION ZUTMP2(NCAX,NPAX)
      DIMENSION CX1(NCAX)
      DIMENSION CDH(NTRIM,NCAX)
      DIMENSION CDHX(NTRIM,NCAX)
      DIMENSION RDIF(NTRIM,NCAX)
      DIMENSION RA(NTRIM,NCAX), RX(NTRIM,NCAX)
      COMMON /DENNIS/ PI,TCXX,TSXX,PIXX
C
      PI=(4.0E0)*ATAN(1.0E0)
C
C     INPUT DATA:
C     in addition to PARAMETERS at top
      NGRP=6
      KW=8
      TCXX=COS(PI/FLOAT(NGRP))
      TSXX=SIN(PI/FLOAT(NGRP))
      PIXX=PI/TCXX
      Y11=.085E0
      Y22=.095E0
      DATA ITRIM /30/
C     these are the actual M0-values (in the basic summation)
      DO 4 I=1,IFT2
        J=I-IFT
        XQW=(FLOAT(J)-.5E0)*(TCXX)/FLOAT(IFT)
        ZA(I)=CMPLX(XQW,Y11)
 4    CONTINUE
      DO 6 I=1,IFT2
        J=I-IFT
        XQW=(FLOAT(J)-.5E0)*(TCXX)/FLOAT(IFT)
        ZA(I+IFT2)=CMPLX(XQW,Y22)
 6    CONTINUE
C
C     END-OF-DATA
C
C
```

```
C
        DO 5103 I=1,IFT4
          Z33=ZA(I)
          XA=REAL(Z33)
          YA=AIMAG(Z33)
          CALL IMAGE(NGRP,Z33,ZSTAR,R,RINV)
          ZP(I)=ZSTAR
          REC(I,1)=R(1,1)
          REC(I,2)=R(1,2)
          REC(I,3)=R(2,1)
          REC(I,4)=R(2,2)
          REC(I,5)=ATAN2(R(2,1)*YA, R(2,1)*XA + R(2,2))
 5103   CONTINUE
C
        TIME0=SECOND(1)
C
        CALL FUNC(ZA,ZP,REC,KW,ZU)
C
        YCRIT=(.5E0)*FLOAT(KW-1)/PIXX
        CALL CALCR(KW,YCRIT,YR)
        GMAX=YR
C
        DO 6100 I4=1,NTRIM
C       prepare an adjusted copy of matrix ZU
C       and, then, proceed to do linear algebra
C       note that the throw-away row can be modified!!!!!
        NMAT=ITRIM(I4)-1
        DO 6123 K=1,NMAT
          ZUTMP2(K,NMAT+1)=-ZU(K+1,1)
 6123   CONTINUE
        DO 6124 K0=1,NMAT
        DO 6125 K=1,NMAT
          ZUTMP2(K,K0)=ZU(K+1,K0+1)
 6125   CONTINUE
 6124   CONTINUE
        CALL CSMAT(ZUTMP2,NMAT,CX1)
          CDH(I4,1)=(1.0E0,0.0E0)
        DO 6128 K=2,NMAT+1
          CDH(I4,K)=CX1(K-1)
 6128   CONTINUE
C
        Y11=AIMAG(ZA(1))
        YA=(2.0E0)*Y11
        YB=FLOAT(NMAT+1)*Y11
        CALL CALCR(KW,YA,GA)
        CALL CALCR(KW,YB,GB)
        GT=SQRT(Y11)*AMIN1(GA,GB)
        G11=GMAX/GT
C
        DO 6134 K0=1,NMAT+1
        W99=0.0E0
        DO 6135 K=2,NMAT+1
          W=CABS(ZU(K,K0))
          W99=AMAX1(W99,W)
 6135   CONTINUE
        RA(I4,K0)=W99
 6134   CONTINUE
C
C
C
        DO 6223 K=1,NMAT
          ZUTMP2(K,NMAT+1)=-ZU(NPAX+K+1,1)
 6223   CONTINUE
```

```
        DO 6224 K0=1,NMAT
        DO 6225 K=1,NMAT
          ZUTMP2(K,K0)=ZU(NPAX+K+1,K0+1)
 6225   CONTINUE
 6224   CONTINUE
        CALL CSMAT(ZUTMP2,NMAT,CX1)
          CDHX(I4,1)=(1.0E0,0.0E0)
        DO 6228 K=2,NMAT+1
          CDHX(I4,K)=CX1(K-1)
 6228   CONTINUE
C
        Y22=AIMAG(ZA(1+IFT2))
        YA=(2.0E0)*Y22
        YB=FLOAT(NMAT+1)*Y22
        CALL CALCR(KW,YA,GA)
        CALL CALCR(KW,YB,GB)
        GT=SQRT(Y22)*AMIN1(GA,GB)
        G22=GMAX/GT
C
        DO 6234 K0=1,NMAT+1
        W99=0.0E0
        DO 6235 K=2,NMAT+1
          W=CABS(ZU(NPAX+K,K0))
          W99=AMAX1(W99,W)
 6235   CONTINUE
        RX(I4,K0)=W99
 6234   CONTINUE
C
C
C
        DO 6400 K=2,NMAT+1
          C88=CDH(I4,K)-CDHX(I4,K)
          D88=REAL(C88)
          RDIF(I4,K)=ABS(D88)
 6400   CONTINUE
C
C
 6100   CONTINUE
C
C
        TIME1=SECOND(1)
        TIME=TIME1-TIME0
C
C
C       now do the final print-out
C
        PRINT*,'   '
        PRINT*,'BASIC PARAMETERS:'
        PRINT 90, NGRP, KW
        PRINT 91, IFT
        PRINT 92, AIMAG(ZA(1)), AIMAG(ZA(1+IFT2))
 90     FORMAT('NGRP=',I3,1X,'WEIGHT K=',I3)
 91     FORMAT('FINITE FOURIER Q=',I6)
 92     FORMAT('Y-VALUES:',F9.6,3X,F9.6)
C
C
        DO 8101 I=1,NTRIM
        NMAT=ITRIM(I)-1
        PRINT*,'  '
        PRINT 95, ITRIM(I)
 95     FORMAT('ITRIM=',I4)
        PRINT 951, G11,G22
 951    FORMAT('WITH G-CONTROL RATIOS:',1X,E10.3,4X,E10.3)
```

```
      PRINT*,'RENORMALIZED REAL PARTS:'
      PRINT 96, (K,REAL(CDH(I,K)),REAL(CDHX(I,K)),RDIF(I,K),
     &  K=1,NMAT+1)
 96   FORMAT('re C(',I3,')=',2X,F20.15,2X,F20.15,3X,E9.2)
      PRINT*,'RENORMALIZED IMAGINARY PARTS:'
      PRINT 97, (K, AIMAG(CDH(I,K)), AIMAG(CDHX(I,K)), RA(I,K),
     &  RX(I,K), K=1,NMAT+1)
 97   FORMAT('im C(',I3,')=',2X,F20.15,2X,F20.15,1X,E9.2,1X,E9.2)
 8101 CONTINUE
      END
C
C
C
C
      SUBROUTINE FUNC(ZA,ZP,REC,KW,ZU)
C     CRAY VERSION -- SINGLE PRECISION
C     wherein we calculate the matrix ZU (sub n, sub ell)
      IMPLICIT REAL(A-B,D-H,P-Y)
      IMPLICIT COMPLEX(C,Z)
      INTEGER IFT
      PARAMETER(NMAX=100,NPAX=50,NCAX=50)
      PARAMETER(IFT=200)
      PARAMETER(IFT2=2*IFT)
      PARAMETER(IFT4=4*IFT)
      DIMENSION ZA(IFT4), ZP(IFT4), ZU(NMAX,NPAX), REC(IFT4,5)
      DIMENSION ZREV(IFT4,NPAX), CW(IFT2), U00(NMAX)
      COMMON/ DENNIS/ PI,TCXX,TSXX,PIXX
C
C
      AIFT=1.0E0/FLOAT(IFT2)
      Y11 = AIMAG(ZA(1))
      Y22 = AIMAG(ZA(1+IFT2))
C
      DO 51 N=1,NPAX
        TMP=FLOAT(N)*Y11
        CALL CALCR(KW,TMP,TR)
        U00(N)=SQRT(Y11)*TR
 51   CONTINUE
      DO 52 N=1,NPAX
        TMP=FLOAT(N)*Y22
        CALL CALCR(KW,TMP,TR)
        U00(NPAX+N)=SQRT(Y22)*TR
 52   CONTINUE
C
      DO 200 J=1,IFT4
      YPJ=AIMAG(ZP(J))
      B98=SQRT(YPJ)
      B99= -FLOAT(KW)*REC(J,5)
        DO 205 L=1,NPAX
        Z=FLOAT(L)*ZP(J)
        CALL CALC(KW,Z,ZT)
        ZREV(J,L)=B98*ZT*CEXP(CMPLX(0.0E0,B99))
 205    CONTINUE
 200  CONTINUE
C
      DO 300 N=1,NPAX
      DO 310 L=1,NPAX
        DO 320 J=1,IFT2
        TMP= -PIXX*FLOAT(N)*REAL(ZA(J))
        ZTMP=CMPLX(0.0E0, TMP)
        CW(J)=ZREV(J,L)*CEXP(ZTMP)
 320    CONTINUE
      Z99=CSUM(IFT2,CW,1)*AIFT
```

```
        IF(N.NE.L) THEN
          ZU(N,L)=Z99
        ELSE
          ZU(N,L)=Z99-U00(N)
        ENDIF
 310    CONTINUE
 300    CONTINUE
C
        DO 400 N=1,NPAX
        DO 410 L=1,NPAX
          DO 420 J=1,IFT2
          TMP= -PIXX*FLOAT(N)*REAL(ZA(J+IFT2))
          ZTMP=CMPLX(0.0E0, TMP)
          CW(J)=ZREV(J+IFT2,L)*CEXP(ZTMP)
 420      CONTINUE
        Z99=CSUM(IFT2,CW,1)*AIFT
        IF(N.NE.L) THEN
          ZU(NPAX+N,L)=Z99
        ELSE
          ZU(NPAX+N,L)=Z99-U00(NPAX+N)
        ENDIF
 410    CONTINUE
 400    CONTINUE
        RETURN
        END
C
C
        SUBROUTINE CALC(KW,Z,ZT)
CCC     ZT=Im(Z)^((KW-1)/2) * EXP(i*2PI*Z/2*TCXX)
        IMPLICIT REAL(A-B,D-H,P-Y)
        IMPLICIT COMPLEX(C,Z)
        COMMON/ DENNIS/ PI,TCXX,TSXX,PIXX
        CTMP= CMPLX(0.0E0, 1.0E0)*PIXX*Z
        B91=(.5E0)*FLOAT(KW-1)
        B92=AIMAG(Z)**B91
        ZT=B92*CEXP(CTMP)
        RETURN
        END
C
C
        SUBROUTINE CALCR(KW,Y,YR)
CCC     YR=Y^((KW-1)/2) * EXP(-2PI*Y/2*TCXX)
        IMPLICIT REAL(A-B,D-H,P-Y)
        IMPLICIT COMPLEX(C,Z)
        COMMON/ DENNIS/ PI,TCXX,TSXX,PIXX
        TMP= -Y*PIXX
        B91=(.5E0)*FLOAT(KW-1)
        B92=Y**B91
        YR=B92*EXP(TMP)
        RETURN
        END
C
C
        SUBROUTINE SMAT(CU,N,C)
        IMPLICIT REAL(A-B,D-H,P-Y)
        IMPLICIT COMPLEX(C,Z)
        PARAMETER(NMAX=100,NPAX=50,NCAX=50)
        DIMENSION CU(NCAX,NPAX+1), C(NPAX)
        DIMENSION U(NMAX,NPAX+1)
C
        DO 500 IA=1,N
        DO 501 IB=1,N+1
          U(IA,IB)=REAL(CU(IA,IB))
```

```
 501    CONTINUE
 500    CONTINUE
C
        DO 1000 M=1,N
          TEMP=0.0E0
          MAXI=0
          DO 1100 J=M,N
            IF(TEMP.LT.ABS(U(J,M))) THEN
            TEMP=ABS(U(J,M))
            MAXI=J
            ENDIF
 1100     CONTINUE
*         swap rows if necessary
          IF(MAXI.NE.M) THEN
            DO 1200 J=1,N+1
            T=U(MAXI,J)
            U(MAXI,J)=U(M,J)
            U(M,J)=T
 1200       CONTINUE
          ENDIF
          TEMP=U(M,M)
          U(M,M)=1.0E0
          DO 1300 J=M+1,N+1
            U(M,J)=U(M,J)/TEMP
 1300     CONTINUE
          DO 1400 J=1,N
            IF(J.NE.M) THEN
              T=U(J,M)
              U(J,M)=0.0E0
              DO 1500 K=M+1,N+1
              U(J,K)=U(J,K)-(T*U(M,K))
 1500         CONTINUE
            ENDIF
 1400     CONTINUE
 1000   CONTINUE
        DO 2001 M=1,N
          RCM=U(M,N+1)
          C(M)=CMPLX(RCM,0.0E0)
 2001   CONTINUE
        RETURN
        END
C
C
        SUBROUTINE CSMAT(CU,N,C)
        IMPLICIT REAL(A-B,D-H,P-Y)
        IMPLICIT COMPLEX(C,Z)
        PARAMETER(NMAX=100,NPAX=50,NCAX=50)
        DIMENSION CU(NCAX,NPAX+1), C(NPAX)
        DO 1000 M=1,N
          TEMP=0.0E0
          MAXI=0
          DO 1100 J=M,N
            IF(TEMP.LT.CABS(CU(J,M))) THEN
            TEMP=CABS(CU(J,M))
            MAXI=J
            ENDIF
 1100     CONTINUE
*         swap rows if necessary
          IF(MAXI.NE.M) THEN
            DO 1200 J=1,N+1
            CT=CU(MAXI,J)
            CU(MAXI,J)=CU(M,J)
            CU(M,J)=CT
```

```
 1200       CONTINUE
          ENDIF
          CTEMP=CU(M,M)
          CU(M,M)=(1.0E0,0.0E0)
          DO 1300 J=M+1,N+1
            CU(M,J)=CU(M,J)/CTEMP
 1300     CONTINUE
          DO 1400 J=1,N
            IF(J.NE.M) THEN
              CT=CU(J,M)
              CU(J,M)=(0.0E0,0.0E0)
              DO 1500 K=M+1,N+1
              CU(J,K)=CU(J,K)-(CT*CU(M,K))
 1500         CONTINUE
            ENDIF
 1400     CONTINUE
 1000   CONTINUE
        DO 2001 M=1,N
          C(M)=CU(M,N+1)
 2001   CONTINUE
        RETURN
        END
C
C
      SUBROUTINE IMAGE(N,Z,ZSTAR,R,RINV)
C
C     MAP Z TO ZSTAR
C     N --> GROUP, Z --> INPUT COMPLEX
C     ZSTAR --> OUTPUT COMPLEX, R --> OUTPUT 2 BY 2 MATRIX
C     RINV --> INVERSE OF R
C
      IMPLICIT REAL(A-B,D-H,P-Y)
      IMPLICIT COMPLEX(C,Z)
      REAL  LH
      DIMENSION P(2,2),Q(2,2),R(2,2),RINV(2,2)
      COMMON/ DENNIS/ PI,TCXX,TSXX,PIXX
      NGRP=N
      X=REAL(Z)
      Y=AIMAG(Z)
      LH=(2.0E0)*TCXX
      XD=-ANINT(X/LH)*LH
      X=X+XD
      Q(1,1)=1.0E0
      Q(1,2)=XD
      Q(2,1)=0.0E0
      Q(2,2)=1.0E0

 100  SS=(X**2)+(Y**2)
      IF(SS.LT.(1.0E0)) THEN
         XNEW=-X/SS
         YNEW=Y/SS
         P(1,1)=0.0E0
         P(1,2)=-1.0E0
         P(2,1)=1.0E0
         P(2,2)=0.0E0
         CALL MMULT(2,P,Q,R)
         CALL COPYM(2,R,Q)
C
         XD=-ANINT(XNEW/LH)*LH
         X=XNEW+XD
         Y=YNEW
         P(1,1)=1.0E0
         P(1,2)=XD
```

```
         P(2,1)=0.0E0
         P(2,2)=1.0E0
         CALL MMULT(2,P,Q,R)
         CALL COPYM(2,R,Q)
C
         GOTO 100
      ENDIF
C
      ZSTAR=CMPLX(X,Y)
      CALL COPYM(2,Q,R)
      RINV(1,1)=R(2,2)
      RINV(1,2)=-R(1,2)
      RINV(2,1)=-R(2,1)
      RINV(2,2)=R(1,1)
C
      RETURN
      END
C
C
      SUBROUTINE MMULT(N,P,Q,R)
C
C     MULTIPLY TWO SQUARE MATRICES    P X Q --> R
C     N --> DIMENSION, P & Q --> INPUT MATRICES, R --> OUTPUT MATRIX
C
      IMPLICIT REAL(A-B,D-H,P-Y)
      IMPLICIT COMPLEX(C,Z)
      DIMENSION P(N,N),Q(N,N),R(N,N)
      DO I=1,N
         DO J=1,N
            R(I,J)=0.0E0
            DO K=1,N
               R(I,J)=R(I,J)+P(I,K)*Q(K,J)
            ENDDO
         ENDDO
      ENDDO
      RETURN
      END
C
C
      SUBROUTINE COPYM(N,A,B)
C
C     COPY THE SQUARE MATRIX    A --> B
C     N --> DIMENSION, A --> INPUT MATRIX, B --> OUTPUT MATRIX
C
      IMPLICIT REAL(A-B,D-H,P-Y)
      IMPLICIT COMPLEX(C,Z)
      DIMENSION A(N,N),B(N,N)
      DO I=1,N
         DO J=1,N
            B(I,J)=A(I,J)
         ENDDO
      ENDDO

      RETURN
      END
C
C
```