

Nov 28, 11 14:12

rllsll.f90

Page 1/20

**MODULE MOD\_RLLSLL****CONTAINS**

```
SUBROUTINE RLLSLL(RPNBOUND,RMESH,VLL,RLL,SLL,TLLP, &
  NCHEB,NPAN,LMSIZE,LMSIZE2,NRMAX, &
  nvec,jlk_index,hlk,jlk,hlk2,jlk2,GMATPREFACTOR, &
  cmoderll,cmodesll,cmotetest,vll2ddr)
```

```
! *****
! radial wave functions by the integral equation method of
! Gonzalez et al, Journal of Computational Physics 134, 134-149 (1997)
! *****
! This routine solves the following two equations:
!
!  $RLL(r) = J(r) - PRE * J(r) * \int_0^r dr' r'^2 H_2(r') * op(V(r')) * RLL(r')$ 
!  $+ PRE * H(r) * \int_0^r dr' r'^2 J_2(r') * op(V(r')) * RLL(r')$ 
!
!  $SLL(r) = H(r) - PRE * H(r) * \int_0^r dr' r'^2 H_2(r') * op(V(r')) * RLL(r')$ 
!  $+ PRE * J(r) * \int_0^r dr' r'^2 H_2(r') * op(V(r')) * SLL(r')$ 
!
! where the integral  $\int_0^r$  runs from 0 to r
! *****
! Potential matrix : VLL(LMSIZE*NVEC,LMSIZE*NVEC)
! *****
! Green function prefactor PRE=GMATPREFACTOR (scalar value)
! *****
! Source terms : J, H (nvec*lmsize,lmsize) or (lmsize,nvec*lmsize)
! J2,H2 (lmsize,nvec*lmsize) or (nvec*lmsize,lmsize)
!
!
! The source term J is for LMSIZE=3 and NVEC=2 given by:
! J =
!      / jlk(jlk_index(1))
!      |      0      jlk(jlk_index(2))
!      |      0      0      jlk(jlk_index(3))
!      | jlk(jlk_index(4))
!      |      0      jlk(jlk_index(5))
!      |      0      0      jlk(jlk_index(6))
!      \
! *****
! Operator op() can be chosen to be a unity or a transpose operation
! The unity operation is used to calculate the right solution
! The transpose operation is used to calculate the left solution
! *****
! RMESH - radial mesh
! RPNBOUND - panel bounds RPNBOUND(0) left panel border of panel 1
! RPNBOUND(1) right panel border of panel 1
! NCHEB - highest chebyshev polynomial
! number of points per panel = NCHEB + 1
! NPAN - number of panels
! LMSIZE - number of columns for the source matrix J etc...
! LMSIZE2 - number of rows for the source matrix J etc...
! NRMAX - total number of radial points (NPAN*(NCHEB+1))
! NVEC - number of LMSIZE*LMSIZE blocks in J (LMSIZE2=NVEC*LMSIZE)
! *****
USE MOD_TIMING
USE MOD_BESHANK
USE MOD_CHEBINT
USE MOD_CONFIG, only: config_testflag
USE MOD_checknan
USE MOD_RLLSLLTOOLS
USE mod_physic_params, only: cvlight
USE SourceTerms
USE MOD_CHEBSHEV
IMPLICIT NONE
  INTEGER NCHEB,NPAN,LMSIZE,LMSIZE2
  INTEGER NRMAX
  INTEGER IVEC, IVEC2, NVEC
  PARAMETER (NRMAX=(N+1)*MMAX)
  DOUBLE COMPLEX CI,CONE,CZERO
  PARAMETER (CI=(0.0D0,1.0D0),CONE=(1.0D0,0.0D0))
  PARAMETER (CZERO=(0.0D0,0.0D0))
!
```

Nov 28, 11 14:12

rllsll.f90

Page 2/20

```
! .. Local Scalars ..
DOUBLE COMPLEX GMATPREFACTOR
! DOUBLE COMPLEX AU,BU,AS,BS,F1U,F2U,F1S,F2S
DOUBLE PRECISION PLLM
INTEGER INFO,ICHEB3,ICHEB2,ICHEB,ICHEB1,IPAN,MN,NM,MN2,NPLM
INTEGER L1,L2,LM1,LM2,LM3
!
! .. Local Arrays ..

DOUBLE COMPLEX :: HLK(:, :), &
  JLK(:, :),
! + NLK(:, :),
DOUBLE COMPLEX :: HLK2(:, :), &
  JLK2(:, :),
! + NLK2(:, :),
! DOUBLE COMPLEX HLK(0:(LMAX+1)*NVEC-1,0:N),
! + JLK(0:(LMAX+1)*NVEC-1,0:N),
! + NLK(0:(LMAX+1)*NVEC-1,0:N),
! DOUBLE COMPLEX HLK2(0:(LMAX+1)*NVEC-1,0:N),
! + JLK2(0:(LMAX+1)*NVEC-1,0:N),
! + NLK2(0:(LMAX+1)*NVEC-1,0:N)

character(len=1) :: cmoderll,cmodesll,cmotetest

DOUBLE COMPLEX &
  SLL(LMSIZE2,LMSIZE,NRMAX), &
  RLL(LMSIZE2,LMSIZE,NRMAX), TLLP(LMSIZE,LMSIZE), &
  SLLP(LMSIZE,LMSIZE), &
  VLL(LMSIZE*nvec,LMSIZE*nvec,NRMAX)

DOUBLE COMPLEX, allocatable :: VLL2DDR(:, :),
DOUBLE COMPLEX, allocatable :: ULL(:, :, :),

DOUBLE COMPLEX, allocatable :: &
! + SLLP(:, :),
  WORK(:, :), &
  WORK2(:, :), &
  ALLP(:, :, :), BLLP(:, :, :), &
  CLLP(:, :, :), DLLP(:, :, :), &
  SLV(:, :, :), SRV(:, :, :), &
  SLV1(:, :, :), SRV1(:, :, :), &
  SLV2(:, :, :), SRV2(:, :, :), &
  SLV3(:, :, :), SRV3(:, :, :), &
  MRNVY(:, :, :), MRNVZ(:, :, :), &
  MRJVY(:, :, :), MRJVZ(:, :, :), &
  MIHVV(:, :, :), MIHVZ(:, :, :), &
  MIJVV(:, :, :), MIJVZ(:, :, :), &
  YILL(:, :, :), ZILL(:, :, :), &
  YRLL(:, :, :), ZRLL(:, :, :), YRLLTMP(:, :, :), &
  YILL1(:, :, :), ZILL1(:, :, :), &
  YRLL1(:, :, :), ZRLL1(:, :, :), &
  YILL2(:, :, :), ZILL2(:, :, :), &
  YRLL2(:, :, :), ZRLL2(:, :, :), &
! + ULL(:, :, :), SLL(:, :, :),
! + RLL(:, :, :), !HLK(LMSIZE,LMSIZE,NRMAX),
! + VNL(:, :, :),
  VHLR(:, :, :), &
  VJLR(:, :, :), &
  VHLL(:, :, :), &
  VJLI(:, :, :),
! + VLL(:, :, :),
DOUBLE COMPLEX, allocatable :: YIF(:, :, :), &
```

Nov 28, 11 14:12

rllsll.f90

Page 3/20

```

        YRF(:,:,:), &
        ZIF(:,:,:), &
        ZRF(:,:,:)
DOUBLE COMPLEX ZSLC1SUM(0:NCHEB)
DOUBLE PRECISION C1(0:NCHEB,0:NCHEB), RPBANBOUND(0:NPAN)
DOUBLE PRECISION CSLC1(0:NCHEB,0:NCHEB), CSRC1(0:NCHEB,0:NCHEB), &
        TAU(0:NCHEB,0:NPAN), CDDRC1(0:NCHEB,0:NCHEB), CDDRC1temp(0:
NCHEB), CDDRC1temp2, &
        SLC1SUM(0:NCHEB), RMESH(NRMAX)
INTEGER jlk_index(:), IPIV(0:NCHEB,LMSIZE2)
INTEGER, allocatable :: IPIV2(:)
LOGICAL TEST
INTEGER :: IERROR, USE_SRATRICK

!       INTEGER ISPINfullgmat

!       DOUBLE PRECISION, parameter :: CVLIGHT = 274.0720442D0
!
! .. External Subroutines ..
EXTERNAL ZGETRF, ZGETRS
!
! .. Intrinsic Functions ..
INTRINSIC ABS, ATAN, COS, DIMAG, EXP, MAX, MIN, SIN, SQRT
!
if ( .not. config_testflag('sph') .or. LMSIZE==1 ) then
    use_sratrick=0
elseif ( config_testflag('sph') ) then
    use_sratrick=1
else
    stop '[rllsll] use_sratrick error'
end if

    call timing_start('rllsll')
    call timing_start('prestuff2')

!       IF (NSRA<=2) THEN
!       ALLOCATE(      HLK(0:(LMAX+1)*NVEC-1,0:N),
!       +            JLK(0:(LMAX+1)*NVEC-1,0:N),
!       ! !      +            NLK(0:(LMAX+1)*NVEC-1,0:N),
!       ! !      +            HLK2(0:(LMAX+1)*NVEC-1,0:N),
!       ! !      +            JLK2(0:(LMAX+1)*NVEC-1,0:N) )
!       ! !      +            NLK2(0:(LMAX+1)*NVEC-1,0:N) )
!
!       ELSEIF (NSRA==3) THEN
!       ALLOCATE(      HLK(2*LMSIZE,0:N),
!       +            JLK(2*LMSIZE,0:N),
!       ! !      +            NLK(2*LMSIZE,0:N),
!       ! !      +            HLK2(2*LMSIZE,0:N),
!       ! !      +            JLK2(2*LMSIZE,0:N) )
!       ! !      +            NLK2(2*LMSIZE,0:N) )
!
!       ELSE
!       STOP '[RLLSLL] NSRA not known'
!       END IF

! if ( config_testflag('writesourceterms')) then
! do lm1=1,ubound(jlk,1)
!   write(3661,'(50000E)') jlk(lm1,:)
!   write(3662,'(50000E)') hlk(lm1,:)
!   write(3663,'(50000E)') jlk2(lm1,:)
!   write(3664,'(50000E)') hlk2(lm1,:)
! end do
! write(*,*) jlk_index
! stop
! end if

```

Nov 28, 11 14:12

rllsll.f90

Page 4/20

```

if ( config_testflag('checknanrllsll') ) then
    call checknan(vll,ierror)
    if (ierror==1) then
        write(*,*) 'vll nan'
        stop
    end if
    call checknan(hlk,ierror)
    if (ierror==1) then
        write(*,*) 'hlk nan'
        stop
    end if
    call checknan(jlk,ierror)
    if (ierror==1) then
        write(*,*) 'jlk nan'
        stop
    end if
    call checknan(hlk2,ierror)
    if (ierror==1) then
        write(*,*) 'hlk2 nan'
        stop
    end if
    call checknan(jlk2,ierror)
    if (ierror==1) then
        write(*,*) 'jlk2 nan'
        stop
    end if
end if
! write(*,*) jlk_index
! stop

    ALLOCATE ( ULL(LMSIZE2,LMSIZE,NRMAX) )

    if ( use_sratrick==0 ) then

        ALLOCATE ( SLV(0:NCHEB,LMSIZE2,0:NCHEB,LMSIZE2), SRV(0:NCHEB,LMSIZE2,0:NCHEB,
LMSIZE2) )

        elseif ( use_sratrick==1 ) then

            ALLOCATE ( WORK2((NCHEB+1)*LMSIZE, (NCHEB+1)*LMSIZE), IPIV2((NCHEB+1)*LMSIZE
))

            ALLOCATE ( SLV1(0:NCHEB,LMSIZE,0:NCHEB,LMSIZE), SRV1(0:NCHEB,LMSIZE,0:NCHEB,
LMSIZE), &
                        SLV2(0:NCHEB,LMSIZE,0:NCHEB,LMSIZE), SRV2(0:NCHEB,LMSIZE,0:NCHEB,
LMSIZE), &
                        SLV3(0:NCHEB,LMSIZE,0:NCHEB,LMSIZE), SRV3(0:NCHEB,LMSIZE,0:NCHEB,
LMSIZE) )

            ALLOCATE (      YILL1(0:NCHEB,LMSIZE,LMSIZE), ZILL1(0:NCHEB,LMSIZE,LMSIZE),
&
                        YRLL1(0:NCHEB,LMSIZE,LMSIZE), ZRLL1(0:NCHEB,LMSIZE,LMSIZE),
&
                        YILL2(0:NCHEB,LMSIZE,LMSIZE), ZILL2(0:NCHEB,LMSIZE,LMSIZE),
&
                        YRLL2(0:NCHEB,LMSIZE,LMSIZE), ZRLL2(0:NCHEB,LMSIZE,LMSIZE), &
                        YRLLTMP(0:NCHEB,LMSIZE,LMSIZE) )

        else
            stop '[rllsll] error with testflag sph'
        end if

        ALLOCATE(      &
            WORK(LMSIZE,LMSIZE), &
            ALLP(LMSIZE,LMSIZE,0:NPAN), BLLP(LMSIZE,LMSIZE,0:NPAN), &
            CLLP(LMSIZE,LMSIZE,0:NPAN), DLLP(LMSIZE,LMSIZE,0:NPAN), &
            MRNVY(LMSIZE,LMSIZE,NPAN), MRNVZ(LMSIZE,LMSIZE,NPAN), &

```

Nov 28, 11 14:12

rllsll.f90

Page 5/20

```

      MRJVY(LMSIZE,LMSIZE,NPAN),MRJVZ(LMSIZE,LMSIZE,NPAN),&
      MIHVV(LMSIZE,LMSIZE,NPAN),MIHVZ(LMSIZE,LMSIZE,NPAN),&
      MIJVV(LMSIZE,LMSIZE,NPAN),MIJVZ(LMSIZE,LMSIZE,NPAN),&
      YILL(0:NCHEB,LMSIZE2,LMSIZE),ZILL(0:NCHEB,LMSIZE2,LMSIZE),&
      YRLI(0:NCHEB,LMSIZE2,LMSIZE),ZRLI(0:NCHEB,LMSIZE2,LMSIZE),&
      +      ULL(LMSIZE,LMSIZE,NRMAX),SLL(LMSIZE,LMSIZE,NRMAX),
      +      RLL(LMSIZE,LMSIZE,NRMAX), !HLL(LMSIZE,LMSIZE,NRMAX),
      VJLRL(LMSIZE,LMSIZE2,0:NCHEB),VHLRL(LMSIZE,LMSIZE2,0:NCHEB),&
      VJLI(LMSIZE,LMSIZE2,0:NCHEB),VHLI(LMSIZE,LMSIZE2,0:NCHEB))
YRLI=(0.0D0,0.0D0)
ZILL=(0.0D0,0.0D0)
YRLI=(0.0D0,0.0D0)
ZILL=(0.0D0,0.0D0)

      ALLOCATE(&
      YIF(LMSIZE2,LMSIZE,0:NCHEB,NPAN),&
      YRF(LMSIZE2,LMSIZE,0:NCHEB,NPAN),&
      ZIF(LMSIZE2,LMSIZE,0:NCHEB,NPAN),&
      ZRF(LMSIZE2,LMSIZE,0:NCHEB,NPAN) )

      ERYD = EIN

      LM1 = 1
      DO L1 = 0,LMAX
      DO M = -L1,L1
      LOFLM(LM1) = L1
      LM1 = LM1 + 1
      END DO
      write(*,*) nvec,LMSIZE2,LMSIZE
      stop
!-----
! Bauer added for Spinorbit Coupling
! LOFLM is an (LMAX+1)**2 array. LOFLM = (/ LOFLM, LOFLM /)
!-----

      VLL=czero

      IF (NSRA<=2) THEN
      LM1 = 1
      DO IVEC=1,NVEC
      DO ISPINfullgmat=0,use_fullgmat
      DO L1 = 0,LMAX
      DO M = -L1,L1
      LOFLM(LM1) = L1+(IVEC-1)*(LMAX+1)
      ! print *, lm1,loflm(lm1)
      LM1 = LM1 + 1
      END DO
      END DO
      END DO !ISPINORBIT=0,use_fullgmat
      END DO !NVEC
      ELSE IF (NSRA==3) THEN
      DO LM1=1,LMSIZE*NVEC
      LOFLM(LM1) = LM1
      END DO !NVEC
      END IF

!-----
      DO IPAN = 1,NPAN
      DO ICHEB = 0,NCHEB
      MN = IPAN*NCHEB + IPAN - ICHEB
      TAU(ICHEB,IPAN) = RMESH(MN)
      END DO
      END DO

      IF (nsra==1) THEN
      ! !      EK = SQRT(ERYD)
      !      GMATPREFACTOR = SQRT(ERYD)
      ! ELSEIF (nsra==2) THEN

```

Nov 28, 11 14:12

rllsll.f90

Page 6/20

```

      !      EK = SQRT(ERYD+(ERYD/CVLIGHT)**2)
      !      GMATPREFACTOR = SQRT(ERYD+(ERYD/CVLIGHT)**2) *(1.0D0+ERYD/CVLIGHT)**2)
      ! ELSEIF (nsra==3) THEN
      !      EK = SQRT(ERYD+(ERYD/CVLIGHT)**2)
      !      GMATPREFACTOR = SQRT(ERYD+(ERYD/CVLIGHT)**2)
      ! ELSE
      !      stop'[rllsll] wrong value for nvec'
      ! END IF

!-----
      call getCLambdaCinv(Ncheb,CDDRC1)

      do icheb = 0, ncheb
      do icheb2 = 0, icheb-1
      write(*,*) 'sub',icheb,icheb2
      CDDRC1temp2 = CDDRC1(icheb2,icheb)
      CDDRC1(icheb2,icheb) = CDDRC1(icheb,icheb2)
      CDDRC1(icheb,icheb2) = CDDRC1temp2
      end do
      end do
      stop

      do icheb = 0, ncheb
      write(11,'(5000E)') CDDRC1(icheb,:)
      end do

      do icheb = 0, (ncheb-1)/2
      write(*,*) 'sub',icheb,ncheb-icheb
      CDDRC1temp = CDDRC1(:,icheb)
      CDDRC1(:,icheb) = CDDRC1(:,ncheb-icheb)
      CDDRC1(:,ncheb-icheb) = CDDRC1temp
      end do

      do icheb = 0, (ncheb-1)/2
      write(*,*) 'sub',icheb,ncheb-icheb
      CDDRC1temp = CDDRC1(icheb,:)
      CDDRC1(icheb,:) = CDDRC1(ncheb-icheb,:)
      CDDRC1(ncheb-icheb,:) = CDDRC1temp
      end do

      stop

      do icheb = 0, ncheb
      write(12,'(5000E)') CDDRC1(icheb,:)
      end do

      CDDRC1=0.0D0
      DO icheb=0,ncheb
      CDDRC1(icheb,icheb)=1.0D0
      END DO

      CALL CHEBINT(CSLC1,CSRC1,SLC1SUM,C1,NCHEB)

      do icheb = 0, ncheb
      write(13,'(5000E)') CSLC1(icheb,:)
      end do

!-----
      call timing_start('local')

! loop over subintervals
      DO IPAN = 1,NPAN
! initialization
      call timing_start('local')
      DO LM2 = 1,LMSIZE
      DO LM1 = 1,LMSIZE2
      DO ICHEB = 0,NCHEB
      YRLI(ICHEB,LM1,LM2) = CZERO

```

Nov 28, 11 14:12 **rllsll.f90** Page 7/20

```

!      ZRLL(ICHEB,LM1,LM2) = CZERO
!      YILL(ICHEB,LM1,LM2) = CZERO
!      ZILL(ICHEB,LM1,LM2) = CZERO
!      END DO
!      END DO
!      DO LM2 = 1,LMSIZE2
!      DO LM1 = 1,LMSIZE2
!      DO ICHEB = 0,NCHEB
!      DO ICHEB1 = 0,NCHEB
!      SLV(ICHEB1,LM1,ICHEB,LM2) = 0.0D0
!      SRV(ICHEB1,LM1,ICHEB,LM2) = 0.0D0
!      END DO
!      END DO
!      END DO
!      DO LM1 = 1,LMSIZE2
!      DO ICHEB = 0,NCHEB
!      SLV(ICHEB,LM1,ICHEB,LM1) = 1.0D0
!      SRV(ICHEB,LM1,ICHEB,LM1) = 1.0D0
!      END DO
!      END DO
!      VNL=czero
!      VHLL=czero
!      VJLR=czero
!      VHLL=czero
!      VJLI=czero
!      VJLI=czero

if (use_sratrick==0) then

    YRLL=czero
    ZRLL=czero
    YILL=czero
    ZILL=czero
else
    YRLL1=czero
    ZRLL1=czero
    YILL1=czero
    ZILL1=czero
    YRLL2=czero
    ZRLL2=czero
    YILL2=czero
    ZILL2=czero
end if

!-----
! 1. prepare VJLR, VNL, VHLL, which appear in the integrands
! TAU(K,IPAN) is used instead of TAU(K,IPAN)**2, which directly gives
! RLL(r) and SLL(r) multiplied with r
!
! 2. prepare the source terms YR, ZR, YI, ZI
! because of the conventions used by
! Gonzalez et al, Journal of Computational Physics 134, 134-149 (1997)
! a factor sqrt(E) is included in the source terms
! this factor is removed by the definition of ZSLC1SUM given below
!

DO ICHEB = 0,NCHEB
    MN = IPAN*NCHEB + IPAN - ICHEB

!      write(76301,'(50000E)') JLK(:,MN) !(:,:,:,M)
!      write(76301,'(50000E)') HLK(:,MN) !(:,:,:,M)

!      IF (NSRA<=2) THEN
!
!      CALL BESHANK(HLK(0,K),JLK(0,K),EK*TAU(K,IPAN),LMAX)
!

```

Nov 28, 11 14:12 **rllsll.f90** Page 8/20

```

!      IF (NSRA==2) THEN
!      CALL BESHANK_SMALLCOMP(HLK(0,K),JLK(0,K),
!      +      EK*TAU(K,IPAN),TAU(K,IPAN),ERYD,LMAX)
!      END IF
!
!      DO L1 = 0,NVEC*(LMAX+1)-1
!      HLK(L1,K) = -CT*HLK(L1,K)
!      END DO
!
!      DO L1 = 0,NVEC*(LMAX+1)-1
!      JLK2(L1,K) = JLK(L1,K)
!      HLK2(L1,K) = HLK(L1,K)
!      END DO
!
!      ELSE IF (NSRA==3) THEN
!
!      !      PASCAL
!      !      Schleifen richtig? zusätzlich noch NLK, NLK2 definieren.
!      call SourceTermSuperVector(LMAX,ERYD,TAU(K,IPAN),JLK(:,K),
!      +      HLK(:,K),JLK2(:,K),HLK2(:,K))
!      END IF
!
!      stop

!      DO IVEC=1,NVEC
!      DO LM2 = 1,LMSIZE2
!      DO LM1 = 1,LMSIZE
!      L1 = jlk_index( LM1+LMSIZE*(IVEC-1) )
!      VJLR(LM1,LM2,ICHEB) = VJLR(LM1,LM2,ICHEB) + &
!      GMATPREFACTOR*TAU(ICHEB,IPAN)*JLK2(L1,MN)*VLL(LM1+LMSIZE*(
!      IVEC-1),LM2,MN)
!      VJLI(LM1,LM2,ICHEB) = VJLI(LM1,LM2,ICHEB) + &
!      GMATPREFACTOR*TAU(ICHEB,IPAN)*JLK2(L1,MN)*VLL(LM2,LM1+LMS
!      IZE*(IVEC-1),MN)
!      VHLL(LM1,LM2,ICHEB) = VHLL(LM1,LM2,ICHEB) + &
!      GMATPREFACTOR*TAU(ICHEB,IPAN)*HLK2(L1,MN)*VLL(LM1+LMSIZE*
!      (IVEC-1),LM2,MN)
!      VHLL(LM1,LM2,ICHEB) = VHLL(LM1,LM2,ICHEB) + &
!      GMATPREFACTOR*TAU(ICHEB,IPAN)*HLK2(L1,MN)*VLL(LM2,LM1+LMS
!      IZE*(IVEC-1),MN)
!      END DO
!      END DO
!      END DO !NVEC

IF (cmoderll=='l') THEN
    DO IVEC2=1,NVEC
    DO LM2 = 1,LMSIZE
    DO IVEC=1,NVEC
    DO LM1 = 1,LMSIZE
    L1 = jlk_index( LM1+LMSIZE*(IVEC-1) )
    VJLR(LM1,LM2+LMSIZE*(IVEC2-1),ICHEB) = VJLR(LM1,LM2+LMSIZE*(
    IVEC2-1),ICHEB) + &
    GMATPREFACTOR*TAU(ICHEB,IPAN)*JLK2(L1,MN)*VLL(LM1+LMSIZE
    *(IVEC-1),LM2+LMSIZE*(IVEC2-1),MN)
    VHLL(LM1,LM2+LMSIZE*(IVEC2-1),ICHEB) = VHLL(LM1,LM2+LMSIZE*(
    IVEC2-1),ICHEB) + &
    GMATPREFACTOR*TAU(ICHEB,IPAN)*HLK2(L1,MN)*VLL(LM1+LMSIZE
    *(IVEC-1),LM2+LMSIZE*(IVEC2-1),MN)
    END DO
    END DO
    END DO !NVEC
ELSEIF (cmoderll=='T') THEN
    DO IVEC2=1,NVEC
    DO LM2 = 1,LMSIZE
    DO IVEC=1,NVEC
    DO LM1 = 1,LMSIZE

```

```

Nov 28, 11 14:12      rllsll.f90      Page 9/20

      L1 = jlk_index( LM1+LMSIZE*(IVEC-1) )
      VJLR(LM1,LM2+LMSIZE*(IVEC2-1),ICHEB) = VJLR(LM1,LM2+LMSIZE*(
IVEC2-1),ICHEB) + &
      GMATPREFACTOR*TAU(ICHEB,IPAN)*JLK2(L1,MN)*VLL(LM2+LMSIZE
*(IVEC-1),LM1+LMSIZE*(IVEC2-1),MN)
      VHLL(LM1,LM2+LMSIZE*(IVEC2-1),ICHEB) = VHLL(LM1,LM2+LMSIZE*(
IVEC2-1),ICHEB) + &
      GMATPREFACTOR*TAU(ICHEB,IPAN)*HLK2(L1,MN)*VLL(LM2+LMSIZE
*(IVEC-1),LM1+LMSIZE*(IVEC2-1),MN)
      END DO
    END DO
  END DO !NVEC
  ELSEIF (cmodesll=='0') THEN
    VJLR(:, :, ICHEB) = CZERO
    VHLL(:, :, ICHEB) = CZERO
  ELSE
    STOP'[RLLSLL] MODE NOT KNOWN'
  END IF

  IF (cmodesll=='1') THEN
    DO IVEC2=1,NVEC
      DO LM2 = 1,LMSIZE
        DO IVEC=1,NVEC
          DO LM1 = 1,LMSIZE
            L1 = jlk_index( LM1+LMSIZE*(IVEC-1) )
            VJLI(LM1,LM2+LMSIZE*(IVEC2-1),ICHEB) = VJLI(LM1,LM2+LMSIZE*(
IVEC2-1),ICHEB) + &
            GMATPREFACTOR*TAU(ICHEB,IPAN)*JLK2(L1,MN)*VLL(LM1+LMSIZE
*(IVEC-1),LM2+LMSIZE*(IVEC2-1),MN)
            VHLL(LM1,LM2+LMSIZE*(IVEC2-1),ICHEB) = VHLL(LM1,LM2+LMSIZE*(
IVEC2-1),ICHEB) + &
            GMATPREFACTOR*TAU(ICHEB,IPAN)*HLK2(L1,MN)*VLL(LM1+LMSIZE
*(IVEC-1),LM2+LMSIZE*(IVEC2-1),MN)
            END DO
          END DO
        END DO !NVEC
      ELSEIF (cmodesll=='T') THEN
        DO IVEC2=1,NVEC
          DO LM2 = 1,LMSIZE
            DO IVEC=1,NVEC
              DO LM1 = 1,LMSIZE
                L1 = jlk_index( LM1+LMSIZE*(IVEC-1) )
                VJLI(LM1,LM2+LMSIZE*(IVEC2-1),ICHEB) = VJLI(LM1,LM2+LMSIZE*(
IVEC2-1),ICHEB) + &
                GMATPREFACTOR*TAU(ICHEB,IPAN)*JLK2(L1,MN)*VLL(LM2+LMSIZE
*(IVEC-1),LM1+LMSIZE*(IVEC2-1),MN)
                VHLL(LM1,LM2+LMSIZE*(IVEC2-1),ICHEB) = VHLL(LM1,LM2+LMSIZE*(
IVEC2-1),ICHEB) + &
                GMATPREFACTOR*TAU(ICHEB,IPAN)*HLK2(L1,MN)*VLL(LM2+LMSIZE
*(IVEC-1),LM1+LMSIZE*(IVEC2-1),MN)
              END DO
            END DO
          END DO !NVEC
        ELSEIF (cmodesll=='0') THEN
          VJLI(:, :, ICHEB) = CZERO
          VHLL(:, :, ICHEB) = CZERO
        ELSE
          STOP'[RLLSLL] MODE NOT KNOWN'
        END IF

        DO IVEC=1,NVEC
          DO LM2 = 1,LMSIZE
            DO LM1 = 1,LMSIZE
              L1 = LOFLM( LM1+LMSIZE*(IVEC-1) )
              VJLR(LM1,LM2+LMSIZE*(IVEC-1),K) =

```

```

Nov 28, 11 14:12      rllsll.f90      Page 10/20

!      +      EK*TAU(K,IPAN)*JLK(L1,K)*VLL(LM1,LM2,MN)
!      VNL(LM1,LM2+LMSIZE*(IVEC-1),K) =
!      +      -EK*TAU(K,IPAN)*NLK(L1,K)*VLL(LM1,LM2,MN)
!      VHLR(LM1,LM2+LMSIZE*(IVEC-1),K) =
!      +      -EK*TAU(K,IPAN)*HLK(L1,K)*VLL(LM1,LM2,MN)
!      END DO
!      END DO
!      END DO !IVEC=1,NVEC

  IF ( use_sratrick==0 ) THEN
    DO IVEC=1,NVEC
      DO LM1 = 1,LMSIZE
        L1 = jlk_index( LM1+LMSIZE*(IVEC-1) )
        YRLL(ICHEB,LM1+LMSIZE*(IVEC-1),LM1) = TAU(ICHEB,IPAN)*JLK(L1,MN)
      )
      ZRLL(ICHEB,LM1+LMSIZE*(IVEC-1),LM1) = TAU(ICHEB,IPAN)*HLK(L1,MN)
    )
      YILL(ICHEB,LM1+LMSIZE*(IVEC-1),LM1) = TAU(ICHEB,IPAN)*HLK(L1,MN)
    )
      ZILL(ICHEB,LM1+LMSIZE*(IVEC-1),LM1) = TAU(ICHEB,IPAN)*JLK(L1,MN)
    )
      !      write(76401,'(50000E)') YRLL(ICHEB,LM1+LMSIZE*(IVEC-1),LM1), &
      !      ZRLL(ICHEB,LM1+LMSIZE*(IVEC-1),LM1), &
      !      YILL(ICHEB,LM1+LMSIZE*(IVEC-1),LM1), &
      !      ZILL(ICHEB,LM1+LMSIZE*(IVEC-1),LM1)

      !      if (cmodesll=='c') THEN
      !      ZRLL(ICHEB,LM1+LMSIZE*(IVEC-1),LM1) = conjg( ZRLL(ICHEB,LM1+L
MSIZE*(IVEC-1),LM1) )
      !      YILL(ICHEB,LM1+LMSIZE*(IVEC-1),LM1) = conjg( ZRLL(ICHEB,LM1+L
MSIZE*(IVEC-1),LM1) )
      !      end if

      END DO
    END DO !IVEC=1,NVEC

  elseif ( use_sratrick==1 ) THEN

    DO LM1 = 1,LMSIZE
      L1 = jlk_index( LM1+LMSIZE*(1-1) )
      L2 = jlk_index( LM1+LMSIZE*(2-1) )
      YRLL1(ICHEB,LM1+LMSIZE*(1-1),LM1) = TAU(ICHEB,IPAN)*JLK(L1,MN)
      ZRLL1(ICHEB,LM1+LMSIZE*(1-1),LM1) = TAU(ICHEB,IPAN)*HLK(L1,MN)
      YILL1(ICHEB,LM1+LMSIZE*(1-1),LM1) = TAU(ICHEB,IPAN)*HLK(L1,MN)
      ZILL1(ICHEB,LM1+LMSIZE*(1-1),LM1) = TAU(ICHEB,IPAN)*JLK(L1,MN)
      YRLL2(ICHEB,LM1+LMSIZE*(1-1),LM1) = TAU(ICHEB,IPAN)*JLK(L2,MN)
      ZRLL2(ICHEB,LM1+LMSIZE*(1-1),LM1) = TAU(ICHEB,IPAN)*HLK(L2,MN)
      YILL2(ICHEB,LM1+LMSIZE*(1-1),LM1) = TAU(ICHEB,IPAN)*HLK(L2,MN)
      ZILL2(ICHEB,LM1+LMSIZE*(1-1),LM1) = TAU(ICHEB,IPAN)*JLK(L2,MN)
    END DO
  end if

  END DO

!      write(76200,'(50000E)') TAU!(ICHEB,IPAN)
!      write(76201,'(50000E)') YRLL !(:,:,:,M)
!      write(76202,'(50000E)') ZRLL !(:,:,:,M)
!      write(76203,'(50000E)') YILL !(:,:,:,M)
!      write(76204,'(50000E)') ZILL !(:,:,:,M)
!      stop
!-----
! determine the matrices in equations (4.5a) and (4.5b)
  IF ( use_sratrick==0 ) THEN

```

Wednesday December 21, 2011

---

90
6/10

Nov 28, 11 14:12

rllsll.f90

Page 13/20

```

ICHEB2) &
      +TAU(ICHEB,IPAN)*HLK(L1,MN)*CSRCL(ICHEB,ICHEB2)*VJLI(LM3,LM2,
ICHEB2)) &
      *(RSPANBOUND(IPAN)-RSPANBOUND(IPAN-1))/ 2.DO

      END DO
    END DO
  END DO
END DO

DO LM1 = 1,LMSIZE
  DO ICHEB = 0,NCHEB
    SLV1(ICHEB,LM1,ICHEB,LM1) = SLV1(ICHEB,LM1,ICHEB,LM1) + 1.DO
    SRV1(ICHEB,LM1,ICHEB,LM1) = SRV1(ICHEB,LM1,ICHEB,LM1) + 1.DO
!    SLV2(ICHEB,LM1,ICHEB,LM1) = SLV2(ICHEB,LM1,ICHEB,LM1) + 1.DO
!    SRV2(ICHEB,LM1,ICHEB,LM1) = SRV2(ICHEB,LM1,ICHEB,LM1) + 1.DO
  END DO
END DO

ELSE
  stop 'rllsll error in inversion'
END IF
call timing_pause('local1')
call timing_start('local2')

!-----
! determine the local solutions
! solve the equations SLV*YRLL=S and SLV*ZRLL=C
! and SRV*YILL=C and SRV*ZILL=S
!
! DO LM1 = 1,LMSIZE
! DO J = 0,N
!   write(452,'(5000F)') SLV(:,J,LM1)
! END DO
! END DO

! DO LM1 = 1,LMSIZE
! DO ICHEB = 0,NCHEB
!   write(452,'(5000F)') YRLL(ICHEB,,:)
! END DO
! END DO

! if (lmsize/=1) then
! DO LM1 = 1,LMSIZE2
! DO ICHEB = 0,NCHEB
!   write(3883,'(5000E)') slv(:,ICHEB,lm1)
! end do
! end do
! stop
! end if

! if (lmsize/=1) then
! DO LM1 = 1,LMSIZE
! DO ICHEB = 0,NCHEB
!   write(3884,'(5000E)') slv1(:,ICHEB,lm1)
! end do
! end do
! DO LM1 = 1,LMSIZE
! DO ICHEB = 0,NCHEB
!   write(3885,'(5000E)') slv2(:,ICHEB,lm1)
! end do
! end do
! stop
! end if

```

Nov 28, 11 14:12

rllsll.f90

Page 14/20

```

!   if ( .true. ) then
!   if ( use_sratick==0 ) then
!     NPLM = (NCHEB+1)*LMSIZE2

!     IF (cmorderll/=0') THEN
!       CALL ZGETRF(NPLM,NPLM,SLV,NPLM,IPIV,INFO)
!       IF (INFO/=0) STOP'RLLSLL: ZGETRF'
!       CALL ZGETRS('N',NPLM,LMSIZE,SLV,NPLM,IPIV,YRLL,NPLM,INFO)
!       CALL ZGETRS('N',NPLM,LMSIZE,SLV,NPLM,IPIV,ZRLL,NPLM,INFO)
!     ELSE
!       YRLL=CZERO
!       ZRLL=CZERO
!     END IF
!     IF (cmodesll/=0') THEN
!       CALL ZGETRF(NPLM,NPLM,SRV,NPLM,IPIV,INFO)
!       IF (INFO/=0) STOP'RLLSLL: ZGETRF'
!       CALL ZGETRS('N',NPLM,LMSIZE,SRV,NPLM,IPIV,YILL,NPLM,INFO)
!       CALL ZGETRS('N',NPLM,LMSIZE,SRV,NPLM,IPIV,ZILL,NPLM,INFO)
!     ELSE
!       YILL=CZERO
!       ZILL=CZERO
!     END IF
!   elseif ( .false. ) then
!   elseif ( use_sratick==1 ) then
!     NPLM = (NCHEB+1)*LMSIZE

!     call rllslltools_solve_sra(NPLM,LMSIZE,SLV,YRLL,ZRLL)

!     call timing_start('local2.1')

!     call inverse(NPLM,slv1,work2,ipiv2)

! call ZGETRF( NPLM, NPLM, slv1, NPLM, IPIV2, INFO )
! ! if (info/=0) stop '[inverse] error INFO'
! call ZGETRI( NPLM, slv1, NPLM, IPIV2, WORK2, NPLM*NPLM, INFO )

!     call inverse(NPLM,slv1,work2,ipiv2)

! call ZGETRF( NPLM, NPLM, slv1, NPLM, IPIV2, INFO )
! ! if (info/=0) stop '[inverse] error INFO'
! call ZGETRI( NPLM, slv1, NPLM, IPIV2, WORK2, NPLM*NPLM, INFO )
!   call timing_pause('local2.1')

!   call timing_start('local2.2')

!   CALL ZGEMM('N','N',NPLM,NPLM,NPLM,-CONE,slv2, &
!     NPLM,slv1,NPLM,CZERO,slv3,NPLM)
!   CALL ZGEMM('N','N',NPLM,NPLM,NPLM,-CONE,slv2, &
!     NPLM,slv1,NPLM,CZERO,slv3,NPLM)

!   CALL ZGEMM('N','N',NPLM,LMSIZE,NPLM,CONE,slv1, &
!     NPLM,YRLL1,NPLM,CZERO,YRLLTMP,NPLM)
!   YRLL1=YRLLTMP
!   CALL ZGEMM('N','N',NPLM,LMSIZE,NPLM,-CONE,slv2, &
!     NPLM,YRLL1,NPLM,CONE,YRLL2,NPLM)

!   CALL ZGEMM('N','N',NPLM,LMSIZE,NPLM,CONE,slv1, &
!     NPLM,ZRLL1,NPLM,CZERO,YRLLTMP,NPLM)
!   ZRLL1=YRLLTMP
!   CALL ZGEMM('N','N',NPLM,LMSIZE,NPLM,-CONE,slv2, &
!     NPLM,ZRLL1,NPLM,CONE,ZRLL2,NPLM)

```

Nov 28, 11 14:12

rllsll.f90

Page 15/20

```

CALL ZGEMM('N','N',NPLM,LMSIZE,NPLM,CONE,svr1, &
  NPLM,YILL1,NPLM,CZERO,YRLLTMP,NPLM)
YILL1=YRLLTMP
CALL ZGEMM('N','N',NPLM,LMSIZE,NPLM,-CONE,svr2, &
  NPLM,YILL1,NPLM,CONE,YILL2,NPLM)

CALL ZGEMM('N','N',NPLM,LMSIZE,NPLM,CONE,svr1, &
  NPLM,ZILL1,NPLM,CZERO,YRLLTMP,NPLM)
ZILL1=YRLLTMP
CALL ZGEMM('N','N',NPLM,LMSIZE,NPLM,-CONE,svr2, &
  NPLM,ZILL1,NPLM,CONE,ZILL2,NPLM)

!      call timing_pause('local2.2')

!      call rllslltools(NPLM,NCHEB+1,LMSIZE,SLV,YRLL,ZRLL)
!      call rllslltools(NPLM,NCHEB+1,LMSIZE,SRV,YILL,ZILL)
ELSE
  stop 'rllsll error in inversion'
END IF

if ( config_testflag('checknanrllsll') ) then
  call checknan(YRLL,ierror)
  if (ierror==1) then
    write(*,*) 'YRLL nan'
    stop
  end if
end if

!      if (lmsize/=1) then
!      DO ICHEB = 0,NCHEB
!        write(453,'(5000F)') YRLL1(ICHEB,,:)
!      END DO
!      stop
!    end if

!-----

if ( use_sratrack==0 ) then
  DO ICHEB = 0,NCHEB
    DO LM2 = 1,LMSIZE
      DO LM1 = 1,LMSIZE2
        YRF(LM1,LM2,ICHEB,IPAN) = YRLL(ICHEB,LM1,LM2)
        ZRF(LM1,LM2,ICHEB,IPAN) = ZRLL(ICHEB,LM1,LM2)
        YIF(LM1,LM2,ICHEB,IPAN) = YILL(ICHEB,LM1,LM2)
        ZIF(LM1,LM2,ICHEB,IPAN) = ZILL(ICHEB,LM1,LM2)
      END DO
    END DO
  END DO

elseif ( use_sratrack==1 ) then
  DO ICHEB = 0,NCHEB
    DO LM2 = 1,LMSIZE
      DO LM1 = 1,LMSIZE
        YRF(LM1,LM2,ICHEB,IPAN) = YRLL1(ICHEB,LM1,LM2)
        ZRF(LM1,LM2,ICHEB,IPAN) = ZRLL1(ICHEB,LM1,LM2)
        YIF(LM1,LM2,ICHEB,IPAN) = YILL1(ICHEB,LM1,LM2)
        ZIF(LM1,LM2,ICHEB,IPAN) = ZILL1(ICHEB,LM1,LM2)
      END DO
    END DO
  END DO

  DO ICHEB = 0,NCHEB
    DO LM2 = 1,LMSIZE

```

Nov 28, 11 14:12

rllsll.f90

Page 16/20

```

DO LM1 = 1,LMSIZE
  YRF(LM1+LMSIZE,LM2,ICHEB,IPAN) = YRLL2(ICHEB,LM1,LM2)
  ZRF(LM1+LMSIZE,LM2,ICHEB,IPAN) = ZRLL2(ICHEB,LM1,LM2)
  YIF(LM1+LMSIZE,LM2,ICHEB,IPAN) = YILL2(ICHEB,LM1,LM2)
  ZIF(LM1+LMSIZE,LM2,ICHEB,IPAN) = ZILL2(ICHEB,LM1,LM2)
END DO
END DO
END DO

end if

! if (lmsize/=1) then
!   write(76101,'(99000E)') YRF(:,1,IPAN)
!   write(76102,'(99000E)') ZRF(:,1,IPAN)
!   write(76103,'(99000E)') YIF(:,1,IPAN)
!   write(76104,'(99000E)') ZIF(:,1,IPAN)
! stop
! end if
!-----
! determine the left hand sides of equations (3.5), (3.6)
! (3.8), and (3.9) with the equations on page 143.
  call timing_pause('local2')
  call timing_start('local3')

DO ICHEB = 0,NCHEB
  ZSLC1SUM(ICHEB) = SLC1SUM(ICHEB) * (RPNBOUND(IPAN)-RPNBOUND(IPAN-1))/
(2.DO)
END DO
CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(0),VHLR(1,1,0), &
  LMSIZE,YRF(1,1,0,IPAN),LMSIZE2,CZERO,MRNVY(1,1,IPAN),LMSIZE)
CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(0),VJLR(1,1,0), &
  LMSIZE,YRF(1,1,0,IPAN),LMSIZE2,CZERO,MRJVY(1,1,IPAN),LMSIZE)
CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(0),VHLR(1,1,0), &
  LMSIZE,ZRF(1,1,0,IPAN),LMSIZE2,CZERO,MRNVZ(1,1,IPAN),LMSIZE)
CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(0),VJLR(1,1,0), &
  LMSIZE,ZRF(1,1,0,IPAN),LMSIZE2,CZERO,MRJVZ(1,1,IPAN),LMSIZE)
CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(0),VHLI(1,1,0), &
  LMSIZE,YIF(1,1,0,IPAN),LMSIZE2,CZERO,MIHVY(1,1,IPAN),LMSIZE)
CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(0),VJLI(1,1,0), &
  LMSIZE,YIF(1,1,0,IPAN),LMSIZE2,CZERO,MIJVY(1,1,IPAN),LMSIZE)
CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(0),VHLI(1,1,0), &
  LMSIZE,ZIF(1,1,0,IPAN),LMSIZE2,CZERO,MIHVZ(1,1,IPAN),LMSIZE)
CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(0),VJLI(1,1,0), &
  LMSIZE,ZIF(1,1,0,IPAN),LMSIZE2,CZERO,MIJVZ(1,1,IPAN),LMSIZE)
DO ICHEB = 1,NCHEB
  CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(ICHEB),VHLR(1,1,ICHE
B), &
    LMSIZE,YRF(1,1,ICHEB,IPAN),LMSIZE2,CONE,MRNVY(1,1,IPAN),LMSIZE)
  CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(ICHEB),VJLR(1,1,ICHE
B), &
    LMSIZE,YRF(1,1,ICHEB,IPAN),LMSIZE2,CONE,MRJVY(1,1,IPAN),LMSIZE)
  CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(ICHEB),VHLR(1,1,ICHE
B), &
    LMSIZE,ZRF(1,1,ICHEB,IPAN),LMSIZE2,CONE,MRNVZ(1,1,IPAN),LMSIZE)
  CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(ICHEB),VJLR(1,1,ICHE
B), &
    LMSIZE,ZRF(1,1,ICHEB,IPAN),LMSIZE2,CONE,MRJVZ(1,1,IPAN),LMSIZE)
  CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(ICHEB),VHLI(1,1,ICHE
B), &
    LMSIZE,YIF(1,1,ICHEB,IPAN),LMSIZE2,CONE,MIHVY(1,1,IPAN),LMSIZE)
  CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(ICHEB),VJLI(1,1,ICHE
B), &
    LMSIZE,YIF(1,1,ICHEB,IPAN),LMSIZE2,CONE,MIJVY(1,1,IPAN),LMSIZE)
  CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(ICHEB),VHLI(1,1,ICHE
B), &
    LMSIZE,ZIF(1,1,ICHEB,IPAN),LMSIZE2,CONE,MIHVZ(1,1,IPAN),LMSIZE)
  CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE2,ZSLC1SUM(ICHEB),VJLI(1,1,ICHE
B), &
    LMSIZE,ZIF(1,1,ICHEB,IPAN),LMSIZE2,CONE,MIJVZ(1,1,IPAN),LMSIZE)

```



Nov 28, 11 14:12

rllsl.f90

Page 17/20

```

      END DO
      call timing_pause('local3')

if ( config_testflag('checknanrllsl') ) then
  call checknan(MRNVY,ierror)
  if (ierror==1) then
    write(*,*) 'MRNVY nan'
    stop
  end if
end if
if ( config_testflag('checknanrllsl') ) then
  call checknan(MRJVY,ierror)
  if (ierror==1) then
    write(*,*) 'MRJVY nan'
    stop
  end if
end if

!      write(76001,'(50000E)') MRNVY(:, :, IPAN)
!      write(76002,'(50000E)') MRNVZ(:, :, IPAN)
!      write(76003,'(50000E)') MIJVY(:, :, IPAN)
!      write(76004,'(50000E)') MIJVZ(:, :, IPAN)
      END DO
! end of loop over the subintervals

      call timing_stop('local')
      call timing_start('afterlocal')

!*****
! calculate A(M), B(M), C(M), D(M) for m from 1 to MMAX
! starting from A(0) = 1, B(0) = 0, C(MMAX) = 0 and D(MMAX) = 1
      DO LM2 = 1,LMSIZE
        DO LM1 = 1,LMSIZE
          BLLP(LM1,LM2,0) = 0.D0
          ALLP(LM1,LM2,0) = 0.D0
        END DO
      END DO
      DO LM1 = 1,LMSIZE
        ALLP(LM1,LM1,0) = 1.D0
      END DO
      DO IPAN = 1,NPAN
        CALL ZCOPY(LMSIZE*LMSIZE,ALLP(1,1,IPAN-1),1,ALLP(1,1,IPAN),1)
        CALL ZCOPY(LMSIZE*LMSIZE,BLLP(1,1,IPAN-1),1,BLLP(1,1,IPAN),1)
        CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE,-CONE,MRNVY(1,1,IPAN), &
          LMSIZE,ALLP(1,1,IPAN-1),LMSIZE,CONE,ALLP(1,1,IPAN),LMSIZE)
        CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE,-CONE,MRNVZ(1,1,IPAN), &
          LMSIZE,BLLP(1,1,IPAN-1),LMSIZE,CONE,ALLP(1,1,IPAN),LMSIZE)
        CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE,CONE,MRJVY(1,1,IPAN), &
          LMSIZE,ALLP(1,1,IPAN-1),LMSIZE,CONE,BLLP(1,1,IPAN),LMSIZE)
        CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE,CONE,MRJVZ(1,1,IPAN), &
          LMSIZE,BLLP(1,1,IPAN-1),LMSIZE,CONE,BLLP(1,1,IPAN),LMSIZE)

if ( config_testflag('checknanrllsl') ) then
  call checknan(MRNVY,ierror)
  if (ierror==1) then
    write(*,*) 'MRNVY nan'
    stop
  end if
end if

if ( config_testflag('checknanrllsl') ) then
  call checknan(MRNVZ,ierror)
  if (ierror==1) then
    write(*,*) 'MRNVY nan'
    stop
  end if
end if

```

Nov 28, 11 14:12

rllsl.f90

Page 18/20

```

      end if
    end if

if ( config_testflag('checknanrllsl') ) then
  call checknan(ALLP,ierror)
  if (ierror==1) then
    write(*,*) 'ALLP nan'
    write(*,*) 'ALLP nan', ipan

    write(7878,'(50000E)') ALLP(:, :, IPAN)

    stop
  end if
end if

      END DO
!
      DO LM2 = 1,LMSIZE
        DO LM1 = 1,LMSIZE
          DLLP(LM1,LM2,NPAN) = 0.D0
          CLLP(LM1,LM2,NPAN) = 0.D0
        END DO
      END DO
      DO LM1 = 1,LMSIZE
        DLLP(LM1,LM1,NPAN) = 1.D0
      END DO
      DO IPAN = 1,NPAN
        CALL ZCOPY(LMSIZE*LMSIZE,CLLP(1,1,IPAN),1,CLLP(1,1,IPAN-1),1)
        CALL ZCOPY(LMSIZE*LMSIZE,DLLP(1,1,IPAN),1,DLLP(1,1,IPAN-1),1)
        CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE,CONE,MIHVZ(1,1,IPAN), &
          LMSIZE,CLLP(1,1,IPAN),LMSIZE,CONE,CLLP(1,1,IPAN-1),LMSIZE)
        CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE,CONE,MIHVV(1,1,IPAN), &
          LMSIZE,DLLP(1,1,IPAN),LMSIZE,CONE,CLLP(1,1,IPAN-1),LMSIZE)
        CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE,-CONE,MIJVZ(1,1,IPAN), &
          LMSIZE,CLLP(1,1,IPAN),LMSIZE,CONE,DLLP(1,1,IPAN-1),LMSIZE)
        CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE,-CONE,MIJVV(1,1,IPAN), &
          LMSIZE,DLLP(1,1,IPAN),LMSIZE,CONE,DLLP(1,1,IPAN-1),LMSIZE)

      END DO
!-----
! determine the regular solution ULL and the irregular solution SLL
      DO IPAN = 1,NPAN
        DO ICHEB = 0,NCHEB
          MN = IPAN*NCHEB + IPAN - ICHEB
          CALL ZGEMM('N','N',LMSIZE2,LMSIZE,LMSIZE,CONE,YRF(1,1,ICHEB,IPAN), &
            LMSIZE2,ALLP(1,1,IPAN-1),LMSIZE,CZERO,ULL(1,1,MN),LMSIZE2)
          CALL ZGEMM('N','N',LMSIZE2,LMSIZE,LMSIZE,CONE,ZRF(1,1,ICHEB,IPAN), &
            LMSIZE2,BLLP(1,1,IPAN-1),LMSIZE,CONE,ULL(1,1,MN),LMSIZE2)
          CALL ZGEMM('N','N',LMSIZE2,LMSIZE,LMSIZE,CONE,ZIF(1,1,ICHEB,IPAN), &
            LMSIZE2,CLLP(1,1,IPAN),LMSIZE,CZERO,SLL(1,1,MN),LMSIZE2)
          CALL ZGEMM('N','N',LMSIZE2,LMSIZE,LMSIZE,CONE,YIF(1,1,ICHEB,IPAN), &
            LMSIZE2,DLLP(1,1,IPAN),LMSIZE,CONE,SLL(1,1,MN),LMSIZE2)

!      write(6670,'(50000E)') CLLP(:, :, IPAN)
!      write(6671,'(50000E)') DLLP(:, :, IPAN)

      END DO
    END DO

      call timing_stop('afterlocal')
      call timing_start('endstuff')

!-----
! replace regular wave function in the first subinterval by a
! linear function times r**(l+1)
! replace irregular wave function in the first subinterval by a
! linear function divided by r**1
!      IF(config_testflag('wforigin')) THEN
!      DO LM2 =1,LMSIZE

```

Nov 28, 11 14:12

rllsll.f90

Page 19/20

```

!      DO LM1 =1,LMSIZE
!      PLLM = 0.5D0*(LOFLM(LM1)+LOFLM(LM2))
!      PLLM = LOFLM(LM1)
!      F1U = ULL(LM1,LM2,2*N+2-N)/TAU(N,2)**(PLLM+1.D0)
!      F2U = ULL(LM1,LM2,2*N+2-N/2)/TAU(N/2,2)**(PLLM+1.D0)
!      AU = (F1U*TAU(N/2,2)-F2U*TAU(N,2))/(TAU(N/2,2)-TAU(N,2))
!      BU = (F1U-F2U)/(TAU(N,2)-TAU(N/2,2))
!      F1S = SLL(LM1,LM2,2*N+2-N)*TAU(N,2)**PLLM
!      F2S = SLL(LM1,LM2,2*N+2-N/2)*TAU(N/2,2)**PLLM
!      AS = (F1S*TAU(N/2,2)-F2S*TAU(N,2))/(TAU(N/2,2)-TAU(N,2))
!      BS = (F1S-F2S)/(TAU(N,2)-TAU(N/2,2))
!      DO K = 0,N
!      MN = N + 1 - K
!      ULL(LM1,LM2,MN) = (AU+BU*TAU(K,1))*TAU(K,1)**(PLLM+1.D0)
!      SLL(LM1,LM2,MN) = (AS+BS*TAU(K,1))/TAU(K,1)**PLLM
!      END DO
!      END DO
!      END DO
!      END IF
! C-----
! transform from Volterra solution to Fredholm solution
! calculate alpha and t matrices
!      CALL ZAXPY(LMSIZE*LMSIZE,CI,BLLP(1,1,MMAX),1,ALLP(1,1,MMAX),1)      ! c
! calculate the transformation matrix alpha
!
! ming A is calculated with a neuman function
!
! ij (?)
! David
!      DO NM = 1,NRMAX
!      CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE,CONE,SLL(1,1,NM),
!      +      LMSIZE,ALLP(1,1,MMAX),LMSIZE,CZERO,HLL(1,1,NM),LMSIZE)
!      END DO
! end David
!
!      CALL ZGETRF(LMSIZE,LMSIZE,ALLP(1,1,NPAN),LMSIZE,IPIV,INFO)      !
! invert alpha
!      CALL ZGETRI(LMSIZE,ALLP(1,1,NPAN),LMSIZE,IPIV,WORK,LMSIZE*LMSIZE,INFO)
! invert alpha -> transformation matrix RLL=alpha^-1*RLL
!      CALL ZGEMM('N','N',LMSIZE,LMSIZE,LMSIZE,CONE/GMATPREFACTOR,BLLP(1,1,NPAN
! ), &      ! calc t-matrix TLL = BLL*alpha^-1
!      LMSIZE,ALLP(1,1,NPAN),LMSIZE,CZERO,TLLP,LMSIZE)
!      DO LM2 = 1,LMSIZE
!      DO LM1 = 1,LMSIZE
!      SRLLP(LM1,LM2) = 2.D0*EK*TLLP(LM1,LM2)
!      END DO
!      SRLLP(LM2,LM2) = SRLLP(LM2,LM2) + CI
!      END DO
!      CALL ZGETRF(LMSIZE,LMSIZE,SRLLP,LMSIZE,IPIV,INFO)
!      CALL ZGETRI(LMSIZE,SRLLP,LMSIZE,IPIV,WORK,LMSIZE*LMSIZE,INFO)
!
!
!      DO NM = 1,NRMAX
!      CALL ZGEMM('N','N',LMSIZE2,LMSIZE,LMSIZE,CONE,ULL(1,1,NM), &
!      LMSIZE2,ALLP(1,1,NPAN),LMSIZE,CZERO,RLL(1,1,NM),LMSIZE2)
!      END DO
!      call timing_stop('endstuff')
!
!      call timing_start('checknan')
!
!      DO LM2 = 1,LMSIZE
!      DO LM1 = 1,LMSIZE
!      write(7381,'(50000E)') ULL(lm1,lm2,:)
!      END DO
!      END DO
!
!      write(7382,'(50000E)') ALLP(:, :,NPAN)

```

Nov 28, 11 14:12

rllsll.f90

Page 20/20

```

if ( config_testflag('checknanrllsll') ) then
  call checknan(ULL,ierror)
  if (ierror==1) then
    write(*,*) 'ULL nan'
    stop
  end if
end if

if ( config_testflag('checknanrllsll') ) then
  call checknan(RLL,ierror)
  if (ierror==1) then
    write(*,*) 'RLL nan'
    stop
  end if
end if

  call timing_stop('checknan')

  call timing_stop('local1')
  call timing_stop('local2')
!   if (use_sratrik==1) call timing_stop('local2.1')
!   if (use_sratrik==1) call timing_stop('local2.2')
  call timing_stop('local3')

  call timing_stop('rllsll')
! stop

! write(554321,'(50000E)') ull(1,1,:)
  RETURN
END SUBROUTINE

subroutine inverse(nmat,mat)
!interface
integer      :: nmat
double complex :: mat(nmat,nmat)
double complex :: work(nmat,nmat)
!local
integer      :: IPIV(nmat)
integer      :: info

call ZGETRF( nmat, nmat, mat, nmat, IPIV, INFO )
if (info/=0) stop '[inverse]error INFO'
call ZGETRI( nmat, mat, nmat, IPIV, WORK, nmat*nmat, INFO )
if (info/=0) stop '[inverse]error INFO'
end subroutine inverse

```

END MODULE MOD\_RLLSLL