```
========== beginning of kempo1.f ================================
c*****************************************************************
c
c        1D Electromagnetic Full Particle Code   : KEMPO1
c
c               by Yoshiharu Omura
c               Radio Atmospheric Science Center, Kyoto University
c               Uji, Kyoto, 611, Japan
c                    E-mail: omura@kurasc.kyoto-u.ac.jp
c                    FAX:   +81-774-31-8463
c
c                         Version 5.1       May 26, 1993
c
c*****************************************************************
        program main
        common /timecm/ itime,ntime,iecrct,iwrite,jobnum
        common /diagcm/ iediag, ifdiag, ikdiag, ipdiag, isdiag, ivdiag,
       &                ieplot, ifplot, ikplot, ipplot, isplot, ivplot
c
        call plots
        call factor(0.9)
        call input
        call chkprm
        call pltprm
        call renorm
        if(jobnum.le.1) then
          itime = 0
          call inital
          call positn
          call charge
          call ecrrct
        else
          call reader
        endif
        call fldplt
        call phsplt
        call vdsplt
        ist = itime
c
        do 100 j = ist+1, ist+ntime
          itime = j
          call bfield
```

```
            call velcty
            call positn
c           call currnt
            call curntv
            call bfield
            call efield
            call positn
            if( mod(j,iecrct).eq.0) then
                 call charge
                 call ecrrct
            endif
            if( mod(j,ifdiag).eq.0 ) then
                 if( mod(ifdiag,iecrct).ne.0 ) call charge
                 call fldplt
            endif
            if( mod(j,ikdiag).eq.0 ) call kspplt
            if( mod(j,ipdiag).eq.0 ) call phsplt
            if( mod(j,ivdiag).eq.0 ) call vdsplt
            if( mod(j,isdiag).eq.0 ) call spectr
            if( mod(j,iediag).eq.0 ) call energy
            if( mod(j,iwrite).eq.0 ) call writer
  100    continue
         call writer
         call plot(0.,0.,999)
         stop
         end
c*********************************************************************
         subroutine input
#include "paramt.h"
         common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
         common /ptprmc/ wp(is), qm(is), q(is), vpe(is), vpa(is),
        #                vd(is), pch(is), np(is)
         common /timecm/ itime,ntime,iecrct,iwrite,jobnum
         common /diagcm/ iediag, ifdiag, ikdiag, ipdiag, isdiag, ivdiag,
        #                ieplot, ifplot, ikplot, ipplot, isplot, ivplot
         common /otherc/ vmin,vmax
         common /inputc/ dx, dt, cv, wc, angle
c-------------------------------------------------------
         dx       = 1.0
         dt       = 0.04
         nx       = 32
         ntime    = 4096
```

```
      iediag = 16
      isdiag = 8
      ifdiag = 8192
      ipdiag = 8192
      ivdiag = 9999
      ikdiag = 8192
      ieplot = ntime/iediag
      ifplot = 2
      ikplot = nx/2
      isplot = 512
      ipplot = 1
      ivplot = 1
      vmin     = - 20.0
      vmax     =   20.0
      cv       = 20
      wc       = -1.0
      angle    = 90.
      iecrct = 32
      iwrite = 8192
      jobnum = 0
c
      ns       = 1
      wp(1)    = 2.
      qm(1)    = -1.0
      vpe(1) = 4.0
      vpa(1) = 4.0
      vd(1)    = 0.0
      pch(1) = 0.0
      np(1)    = 512
c
      wp(2)    = 1.0
      qm(2)    = -1.0
      vpe(2) = 0.5
      vpa(2) = 0.5
      vd(2)    = 5.0
      pch(2) = 0.0
      np(2)    = 2048
c
      wp(3)    = 0.5
      qm(3)    = -1.0
      vpe(3) = 0.5
      vpa(3) = 0.5
```

```
      vd(3)   = 10.0
      pch(3) = 0.0
      np(3)   =   2048
c-------------------------------------------------------
      return
      end
c**********************************************************************
      subroutine bfield
#include "paramt.h"
      common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
      common /fieldc/ ex(ix), ey(ix), ez(ix), by(ix), bz(ix),
     &                ajx(ix), ajy(ix), ajz(ix), rho(ix)
c
      do 200 i=2,nxp1
        by(i) = by(i) + ez(i) - ez(i-1)
        bz(i) = bz(i) - ey(i+1) + ey(i)
  200 continue
      by(nxp2) = by(2)
      bz(1)     = bz(nxp1)
      return
      end
c**********************************************************************
      subroutine charge
#include "paramt.h"
      common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
      common /fieldc/ ex(ix), ey(ix), ez(ix), by(ix), bz(ix),
     &                ajx(ix), ajy(ix), ajz(ix), rho(ix)
      common /prtclc/ x(in), vx(in), vy(in),vz(in)
      common /ptprmc/ wp(is), qm(is), q(is), vpe(is), vpa(is),
     &                vd(is), pch(is), np(is)
c
      do 100 i=1,nxp2
        rho(i) = rho0
  100 continue
c
          n2 = 0
      do 210 k=1,ns
        n1 = n2
        n2 = n1 + np(k)
        do 200 m = n1+1, n2
          i   =   x(m)+ 2.0
          s2 = (x(m)+ 2.0 - i)*q(k)
```

```
            s1 = q(k) - s2
            rho(i)   = rho(i)   + s1
            rho(i+1) = rho(i+1) + s2
  200    continue
  210 continue
      rho(2)   = rho(2) + rho(nxp2) - rho0
      rho(1)   = rho(nxp1)
      rho(nxp2) = rho(2)
      return
      end
c**********************************************************************
      subroutine chkprm
#include "paramt.h"
      common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
      common /ptprmc/ wp(is), qm(is), q(is), vpe(is), vpa(is),
     &                vd(is), pch(is), np(is)
      common /timecm/ itime,ntime,iecrct,iwrite,jobnum
      common /diagcm/ iediag, ifdiag, ikdiag, ipdiag, isdiag, ivdiag,
     &                ieplot, ifplot, ikplot, ipplot, isplot, ivplot
      common /otherc/ vmin,vmax
      common /resclc/ rex, ret, rev, ree, reb, rej, rer, res
      common /inputc/ dx, dt, cv, wc, angle
c
c-- size of arrays
      if(nx+2.gt.ix) stop 'number of grids (nx) is too large'
      if(ns.gt.is) stop 'number of species (ns) is too large'
      npa = 0
      do 10 i=1,ns
      npa = npa + np(i)
   10 continue
      if(npa.gt.in) stop 'number of particles is too large'
c-- courant condition
      if(dx/dt.le.cv) then
        print*,'courant condition is not satisfied'
        print*,'   make dt less than ', dx/cv
        stop
      end if
c-- paramters for diagnostics, etc.
      if(iediag.eq.0) iediag = 99999999
      if(isdiag.eq.0) isdiag = 99999999
      if(ifdiag.eq.0) ifdiag = 99999999
      if(ipdiag.eq.0) ipdiag = 99999999
```

```
      if(ikdiag.eq.0) ikdiag = 99999999
      if(iecrct.eq.0) iecrct = 99999999
      if(iwrite.eq.0) iwrite = 99999999
      return
      end
c**********************************************************************
      subroutine curntv
#include "paramt.h"
      parameter(lvec=32)
      common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
      common /fieldc/ ex(ix), ey(ix), ez(ix), by(ix), bz(ix),
     &                ajx(ix), ajy(ix), ajz(ix), rho(ix)
      common /prtclc/ x(in), vx(in), vy(in),vz(in)
      common /ptprmc/ wp(is), qm(is), q(is), vpe(is), vpa(is),
     &                vd(is), pch(is), np(is)
      dimension wrk1(lvec,ix),wrk2(lvec,ix),wrk3(lvec,ix)
c
      do 100 i=1,nxp2
        ajx(i) = 0.0
        ajy(i) = 0.0
        ajz(i) = 0.0
  100 continue
      do 150 i=1,nxp2
      do 150 l=1,lvec
        wrk1(l,i) = 0.0
        wrk2(l,i) = 0.0
        wrk3(l,i) = 0.0
  150 continue
c
      n2 = 0
      do 210 k=1,ns
        n1 = n2
        n2 = n1 + np(k)
        qh = q(k)*0.5
        do 200 ik = n1+1,n2,lvec
c$dir no_recurrence
        do 200 m  = ik,min(ik+lvec-1,n2)
          l = m - ik + 1
          ih = x(m) + 1.5
          s2 = (x(m) + 1.5 - ih)*q(k)
          s1 = q(k) - s2
          ih1 = ih + 1
```

```
              wrk2(l,ih )    = wrk2(l,ih ) + vy(m)*s1
              wrk2(l,ih1)    = wrk2(l,ih1) + vy(m)*s2
              wrk3(l,ih )    = wrk3(l,ih ) + vz(m)*s1
              wrk3(l,ih1)    = wrk3(l,ih1) + vz(m)*s2
c--------- charge conservation method ---------
              qhs = qh * sign(1.0, vx(m))
              avx=abs(vx(m))
              x1 = x(m) + 2.0 - avx
              x2 = x(m) + 2.0 + avx
              i1 = x1
              i2 = x2
              wrk1(l,i1) = wrk1(l,i1) + (i2 - x1)*qhs
              wrk1(l,i2) = wrk1(l,i2) + (x2 - i2)*qhs
c-----------------------------------------------
   200     continue
   210 continue
c
         do 300 i=1,nxp2
         do 300 l=1,lvec
            ajx(i) = ajx(i) + wrk1(l,i)
            ajy(i) = ajy(i) + wrk2(l,i)
            ajz(i) = ajz(i) + wrk3(l,i)
   300 continue
c
         ajx(nxp1) = ajx(1) + ajx(nxp1)
         ajx(2)    = ajx(2) + ajx(nxp2)
         ajy(nxp1) = ajy(1) + ajy(nxp1)
         ajy(2)    = ajy(2) + ajy(nxp2)
         ajy(1)    = ajy(nxp1)
         ajz(nxp1) = ajz(1) + ajz(nxp1)
         ajz(2)    = ajz(2) + ajz(nxp2)
c
         do 350 i = nxp1, 2,-1
            ajy(i) = (ajy(i) + ajy(i-1))*0.5
   350 continue
c-------- cancel the uniform component ----------
         juncan = 1
         if(juncan.eq.1) then
         ajxu = 0.0
         ajyu = 0.0
         ajzu = 0.0
         do 400 i = 2,nxp1
```

```
            ajxu = ajxu + ajx(i)
            ajyu = ajyu + ajy(i)
            ajzu = ajzu + ajz(i)
  400 continue
          ajxu = ajxu/float(nx)
          ajyu = ajyu/float(nx)
          ajzu = ajzu/float(nx)
          do 500 i = 2,nxp1
            ajx(i) = ajx(i) - ajxu
            ajy(i) = ajy(i) - ajyu
            ajz(i) = ajz(i) - ajzu
  500 continue
          endif
          return
          end
c*******************************************************************************
          subroutine currnt
#include "paramt.h"
          common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
          common /fieldc/ ex(ix), ey(ix), ez(ix), by(ix), bz(ix),
         &                ajx(ix), ajy(ix), ajz(ix), rho(ix)
          common /prtclc/ x(in), vx(in), vy(in),vz(in)
          common /ptprmc/ wp(is), qm(is), q(is), vpe(is), vpa(is),
         &                vd(is), pch(is), np(is)
          common /timecm/ itime,ntime,iecrct,iwrite,jobnum
          common /resclc/ rex, ret, rev, ree, reb, rej, rer, res
          common /inputc/ dx, dt, cv, wc, angle
c
          do 100 i=1,nxp2
            ajx(i) = 0.0
            ajy(i) = 0.0
            ajz(i) = 0.0
  100 continue
c
          n2 = 0
          do 210 k=1,ns
            n1 = n2
            n2 = n1 + np(k)
            qh = q(k)*0.5
            do 200 m = n1+1, n2
              ih = x(m) + 1.5
              s2 = (x(m) + 1.5 - ih)*q(k)
```

```
            s1 = q(k) - s2
            ih1 = ih + 1
            ajy(ih)     = ajy(ih)    + vy(m)*s1
            ajy(ih1)    = ajy(ih1) + vy(m)*s2
            ajz(ih)     = ajz(ih)    + vz(m)*s1
            ajz(ih1)    = ajz(ih1) + vz(m)*s2
c--------- charge conservation method ---------
            qhs = qh * sign(1.0, vx(m))
            avx=abs(vx(m))
            x1 = x(m) + 2.0 - avx
            x2 = x(m) + 2.0 + avx
            i1 = x1
            i2 = x2
            ajx(i1) = ajx(i1) + (i2 - x1)*qhs
            ajx(i2) = ajx(i2) + (x2 - i2)*qhs
c-----------------------------------------------
  200     continue
  210 continue
c
        ajx(nxp1) = ajx(1) + ajx(nxp1)
        ajx(2)      = ajx(2) + ajx(nxp2)
        ajy(nxp1) = ajy(1) + ajy(nxp1)
        ajy(2)      = ajy(2) + ajy(nxp2)
        ajy(1)      = ajy(nxp1)
        ajz(nxp1) = ajz(1) + ajz(nxp1)
        ajz(2)      = ajz(2) + ajz(nxp2)
c
        do 300 i = nxp1, 2,-1
          ajy(i) = (ajy(i) + ajy(i-1))*0.5
  300 continue
c-------- cancel the uniform component ---------------
        juncan = 1
        if(juncan.eq.1) then
        ajxu = 0.0
        ajyu = 0.0
        ajzu = 0.0
        do 400 i = 2,nxp1
          ajxu = ajxu + ajx(i)
          ajyu = ajyu + ajy(i)
          ajzu = ajzu + ajz(i)
  400 continue
        ajxu = ajxu/float(nx)
```

```
      ajyu = ajyu/float(nx)
      ajzu = ajzu/float(nx)
      do 500 i = 2,nxp1
        ajx(i) = ajx(i) - ajxu
        ajy(i) = ajy(i) - ajyu
        ajz(i) = ajz(i) - ajzu
  500 continue
      endif
c
      return
      end
c**********************************************************************
      subroutine ecrrct
#include "paramt.h"
      common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
      common /fieldc/ ex(ix), ey(ix), ez(ix), by(ix), bz(ix),
     &                ajx(ix), ajy(ix), ajz(ix), rho(ix)
      common /work1c/ work1(ix),work2(ix)
      common /ecrctc/ rkfact(ix)
c
      nxh=nx/2
      do 100 i=2,nxp1
        work1(i-1) = rho(i) - ex(i) + ex(i-1)
  100 continue
      call realft(work1,nx,1)
      do 200 i=1,nx
        work1(i) = work1(i)*rkfact(i)
  200 continue
      call realft(work1,nx,-1)
      work1(nxp1) = work1(1)
      do 300 i=2,nxp1
        ex(i) = ex(i) + work1(i-1) - work1(i)
  300 continue
        ex(1) = ex(nxp1)
      return
      end
c**********************************************************************
      subroutine efield
#include "paramt.h"
      common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
      common /fieldc/ ex(ix), ey(ix), ez(ix), by(ix), bz(ix),
     &                ajx(ix), ajy(ix), ajz(ix), rho(ix)
```

```
c
      do 100 i=2,nxp1
        ex(i) = ex(i) - 2.*ajx(i)
        ey(i) = ey(i) - tcs*( bz(i) - bz(i-1) ) - 2.*ajy(i)
        ez(i) = ez(i) + tcs*( by(i+1) - by(i) ) - 2.*ajz(i)
  100 continue
      ex(1)      = ex(nxp1)
      ey(nxp2) = ey(2)
      ez(1)      = ez(nxp1)
      return
      end
c**********************************************************************
      subroutine energy
#include "paramt.h"
      parameter(iw=1024)
      common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
      common /fieldc/ ex(ix), ey(ix), ez(ix), by(ix), bz(ix),
     &                ajx(ix), ajy(ix), ajz(ix), rho(ix)
      common /prtclc/ x(in), vx(in), vy(in),vz(in)
      common /ptprmc/ wp(is), qm(is), q(is), vpe(is), vpa(is),
     &                vd(is), pch(is), np(is)
      common /timecm/ itime,ntime,iecrct,iwrite,jobnum
      common /diagcm/ iediag, ifdiag, ikdiag, ipdiag, isdiag, ivdiag,
     &                ieplot, ifplot, ikplot, ipplot, isplot, ivplot
      common /resclc/ rex, ret, rev, ree, reb, rej, rer, res
      common /inputc/ dx, dt, cv, wc, angle
      common /work3c/ wkx(iw,is), wky(iw,is), wkz(iw,is),
     &                wdx(iw,is), wdy(iw,is), wdz(iw,is),
     &                wk1(iw), wk2(iw), wk3(iw), wk4(iw)
      common /rotatc/ sinth, costh
      save ic
      data ic/0/
c
      if(ic.eq.0) t1=dt*itime
      ic=ic+1
      te=0.0
      tb=0.0
      n2=0
      do 60 k=1,ns
      n1=n2
      n2=n1+np(k)
      rm=q(k)/qm(k)
```

```
      tkx=0.0
      tky=0.0
      tkz=0.0
      tdx=0.0
      tdy=0.0
      tdz=0.0
      do 10 i=n1+1,n2
        tkx = tkx + vx(i)*vx(i)
        tky = tky + vy(i)*vy(i)
        tkz = tkz + vz(i)*vz(i)
        tdx = tdx + vx(i)
        tdy = tdy + vy(i)
        tdz = tdz + vz(i)
   10 continue
      wkx(ic,k) = 0.5*rm*tkx/slx/res
      wky(ic,k) = 0.5*rm*tky/slx/res
      wkz(ic,k) = 0.5*rm*tkz/slx/res
      wdx(ic,k) = 0.5*rm*tdx*tdx/float(np(k))/slx/res
      wdy(ic,k) = 0.5*rm*tdy*tdy/float(np(k))/slx/res
      wdz(ic,k) = 0.5*rm*tdz*tdz/float(np(k))/slx/res
   60 continue
      do 20 i=2,nxp1
        te = te + ex(i)*ex(i) + ey(i)*ey(i) + ez(i)*ez(i)
   20 continue
      by0   = wc/qm(1)*sinth
      do 30 i=2,nxp1
        tb = tb + (by(i) - by0)**2 + bz(i)**2
   30 continue
      wk1(ic) = 0.5*te/float(nx) /res
      wk2(ic) = 0.25*tcs*tb/float(nx) /res
      wk3(ic) = 0.
      do 40 k=1,ns
        wk3(ic) = wk3(ic) + wkx(ic,k) + wky(ic,k) + wkz(ic,k)
   40 continue
      wk4(ic) = wk1(ic) + wk2(ic) + wk3(ic)
c
      if(ic.eq.ieplot.or.ic.eq.iw) then
      t2=dt*itime
      nt = 20004
      call newpen(5)
      call symbol(0.5,25.8,0.7,'energy',0.,6)
      call qlook(wk1,ic, 7., 2.,10.,10.,t1,t2,'time',nt,'electric',8)
```

```
      call qlook(wk2,ic,22., 2.,10.,10.,t1,t2,'time',nt,'magnetic',8)
      call qlook(wk3,ic, 7.,15.,10.,10.,t1,t2,'time',nt,'kinetic',7)
      call qlook(wk4,ic,22.,15.,10.,10.,t1,t2,'time',nt,'total',5)
      call chart
      wk11=wk1(1)
      wk21=wk2(1)
      wk31=wk3(1)
      wk41=wk4(1)
      do 50 i=1,ic
        wk1(i) = wk1(i) - wk11
        wk2(i) = wk2(i) - wk21
        wk3(i) = wk3(i) - wk31
        wk4(i) = wk4(i) - wk41
   50 continue
      nt = 4
      call newpen(5)
      call symbol(0.5,25.8,0.7,'energy',0.,6)
      call symbol(1.,12.,0.7,'variation',90.,9)
      call qlook(wk1,ic, 7., 2.,10.,10.,t1,t2,'time',nt,'electric',8)
      call prmplt(12.,12.3,0.45,0.,'E0',2,wk11,3)
      call qlook(wk2,ic,22., 2.,10.,10.,t1,t2,'time',nt,'magnetic',8)
      call prmplt(27.,12.3,0.45,0.,'M0',2,wk21,3)
      call qlook(wk3,ic, 7.,15.,10.,10.,t1,t2,'time',nt,'kinetic',7)
      call prmplt(12.,25.3,0.45,0.,'K0',2,wk31,3)
      call qlook(wk4,ic,22.,15.,10.,10.,t1,t2,'time',nt,'total',5)
      call prmplt(27.,25.3,0.45,0.,'T0',2,wk41,3)
      call chart
      do 70 k=1,ns
      do 80 i=1,ic
        wk1(i) = wkx(i,k) + wky(i,k) + wkz(i,k)
        wk2(i) = wdx(i,k) + wdy(i,k) + wdz(i,k)
        wk3(i) = wk1(i) - wk2(i)
        tpara  = wkx(i,k) - wdx(i,k)
        if(tpara.lt.1.e-7) tpara = 1.e-7
        wk4(i) = 0.5*(wky(i,k)+wkz(i,k)-wdy(i,k)-wdz(i,k))/tpara
   80 continue
      call newpen(5)
      call symbol(0.5,25.8,0.7,'energy',0.,6)
      call newpen(3)
      call symbol(24.,25.7,0.8,'species',0.0,7)
      call number(31.,25.7,0.8,float(k),0.0,-1)
      call qlook(wk3,ic, 7., 2.,10.,10.,t1,t2,'time',nt,'thermal',7)
```

```
        call qlook(wk4,ic,22., 2.,10.,10.,t1,t2,
       &           'time',nt,'anisotropy',10)
        call qlook(wk2,ic, 7.,15.,10.,10.,t1,t2,'time',nt,'drift',5)
        call qlook(wk1,ic,22.,15.,10.,10.,t1,t2,'time',nt,'total',5)
        call chart
   70 continue
        ic=0
        end if
c
        return
        end
c*******************************************************************************
        subroutine fldplt
#include "paramt.h"
        common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
        common /fieldc/ ex(ix), ey(ix), ez(ix), by(ix), bz(ix),
       &                ajx(ix), ajy(ix), ajz(ix), rho(ix)
        common /timecm/ itime,ntime,iecrct,iwrite,jobnum
        common /resclc/ rex, ret, rev, ree, reb, rej, rer, res
        common /inputc/ dx, dt, cv, wc, angle
        common /diagcm/ iediag, ifdiag, ikdiag, ipdiag, isdiag, ivdiag,
       &                ieplot, ifplot, ikplot, ipplot, isplot, ivplot
        common /work1c/ work1(ix),work2(ix)
c
        if(mod(ifplot,2).eq.1)  then
        do 10 i=2,nxp1
          work1(i)=ex(i)/ree
   10 continue
        call qlook(work1(2),nx,6.,19.,10.,5.,0.,slx/rex,'x',1,'ex',2)
c       call qlook2(work2(2),nx,0)
        do 20 i=2,nxp1
          work1(i)=ey(i)/ree
   20 continue
        call qlook(work1(2),nx,6.,11.,10.,5.,0.,slx/rex,'x',1,'ey',2)
        do 30 i=2,nxp1
          work1(i)=ez(i)/ree
   30 continue
        call qlook(work1(2),nx,6.,3.,10.,5.,0.,slx/rex,'x',1,'ez',2)
        do 40 i=2,nxp1
          work1(i)=by(i)/reb
   40 continue
        call qlook(work1(2),nx,23.,11.,10.,5.,0.,slx/rex,'x',1,'by',2)
```

```
      do 50 i=2,nxp1
        work1(i)=bz(i)/reb
   50 continue
      call qlook(work1(2),nx,23.,3.,10.,5.,0.,slx/rex,'x',1,'bz',2)
      do 60 i=2,nxp1
        work1(i)=rho(i)/rer
   60 continue
      call qlook(work1(2),nx,23.,19.,10.,5.,0.,slx/rex,'x',1,'rho',3)
      work2(1)=0.0
      work2(2)=0.0
      call newpen(1)
      call qlook2(work2,2,0)
      call newpen(3)
      call prmplt(25.,25.,0.7,0.,'time',4,itime*dt,2)
      call chart
      endif
c
      if(mod(ifplot,4).ge.2)  then
      call qlkmd2(0.2,0.5)
      do 65 i=2,nxp1
        work1(i)=rho(i)/rer
   65 continue
      call qlook(work1(2),nx,8.,3.,20.,5.,0.,slx/rex,'x',1,'rho',3)
      work2(1)=0.0
      work2(2)=0.0
      call newpen(1)
      call qlook2(work2,2,0)
      do 15 i=2,nxp1
        work1(i)=ex(i)/ree
   15 continue
      call qlook(work1(2),nx,8.,11.,20.,5.,0.,slx/rex,'x',1,'ex',2)
      work2(2) = 0.0
      do 70 i=2,nx
        work2(i+1)=work2(i) - ex(i)
   70 continue
      phi0 = work2(2)
      do 80 i=3,nxp1
        phi0 = phi0 + work2(i)
   80 continue
      phi0 = phi0/float(nx)
      do 90 i=2,nxp1
        work1(i) = (work2(i) - phi0)/(ree*rex)
```

```
   90 continue
      call qlook(work1(2),nx,8.,19.,20.,5.,0.,slx/rex,'x',1,
     &                          'potential',9)
      work2(1)=0.0
      work2(2)=0.0
      call newpen(1)
      call qlook2(work2,2,0)
      call newpen(3)
      call prmplt(25.,25.,0.7,0.,'time',4,itime*dt,2)
      call chart
      call qlkmd2(0.0,0.0)
      endif
      return
      end
c**********************************************************************
      subroutine inital
#include "paramt.h"
      common /inputc/ dx, dt, cv, wc, angle
      common /ptprmc/ wp(is), qm(is), q(is), vpe(is), vpa(is),
     &                vd(is), pch(is), np(is)
      common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
      common /rotatc/ sinth, costh
      common /ecrctc/ rkfact(ix)
      common /prtclc/ x(in), vx(in), vy(in),vz(in)
      common /fieldc/ ex(ix), ey(ix), ez(ix), by(ix), bz(ix),
     &                ajx(ix), ajy(ix), ajz(ix), rho(ix)
      common /resclc/ rex, ret, rev, ree, reb, rej, rer, res
c
      dimension xs(is),xl(is)
c
      twopi = 6.283185308
      theta = twopi/360.0*angle
      sinth = sin(theta)
      costh = cos(theta)
      bx0   = wc/qm(1)*costh
      by0   = wc/qm(1)*sinth
      tcs = 2.0*cv*cv
      slx = nx
      nxp1 = nx + 1
      nxp2 = nx + 2
c
      npt=0
```

```
      rho0 = 0.0
      do 10 k = 1,ns
        xs(k) = 0.0
        xl(k) = slx
        npt     = npt + np(k)
        q(k)    = (slx/float(np(k))) * (wp(k)**2) / qm(k)
        rho0    = rho0 + q(k)*np(k)
   10 continue
      rho0 = -rho0/slx
c
      rkmin = twopi/slx
      nxh = nx/2
      fft = 1.0/float(nxh)
      do 300 i=1,nxh-1
        rk    = sin(rkmin*i*0.5)*2.0
        rkfact(2*i+1) = (1.0/rk**2) * fft
        rkfact(2*i+2) = rkfact(2*i+1)
  300 continue
      rkfact(1) = 0.0
      rk    = sin(rkmin*nxh*0.5)*2.0
      rkfact(2) = (1.0/rk**2) * fft
c ------------- Particle Initialization -----------------
      l = 0
      m = 0
      n2=0
      do 200 k=1,ns
        n1=n2
        n2=n1+np(k)
        phi = twopi/360.0*pch(k)
        vdpa = vd(k)*cos(phi)
        vdpe = vd(k)*sin(phi)
        rkk = rkmin*2
c       vmod = 2.0*rev
c       xmod = vmod/rkk
        do 100 i=n1+1,n2
          x(i)    = xs(k) + xl(k)*(i-n1-1)/float(np(k))
          vxi     = vpa(k)*strndm(l) + vdpa
c         vxi    =    vxi +    vmod*cos(rkk*x(i))
          phase = twopi*unrndm(m)
          vyi     = vpe(k)*strndm(l) + vdpe*cos(phase)
          vz(i) = vpe(k)*strndm(l) + vdpe*sin(phase)
          vx(i) = costh*vxi - sinth*vyi
```

```
              vy(i) = sinth*vxi + costh*vyi
c             x(i) =       x(i) +    xmod*sin(rkk*x(i))
   100     continue
   200 continue
c ------------ Field Initialization -----------------
         do 20 i = 1,nxp2
            ex(i) = 0.0
            ey(i) = 0.0
            ez(i) = 0.0
            by(i) = by0
            bz(i) = 0.0
     20 continue
         return
         end
c*******************************************************************************
         subroutine kspplt
#include "paramt.h"
         common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
         common /fieldc/ ex(ix), ey(ix), ez(ix), by(ix), bz(ix),
        &                ajx(ix), ajy(ix), ajz(ix), rho(ix)
         common /timecm/ itime,ntime,iecrct,iwrite,jobnum
         common /diagcm/ iediag, ifdiag, ikdiag, ipdiag, isdiag, ivdiag,
        &                ieplot, ifplot, ikplot, ipplot, isplot, ivplot
         common /resclc/ rex, ret, rev, ree, reb, rej, rer, res
         common /inputc/ dx, dt, cv, wc, angle
         common /work1c/ work1(ix),work2(ix)
c
         rk=6.283185/slx*ikplot *rex
         do 10 i = 1, nx
              work1(i) = ex(i+1) /ree
     10 continue
         call realft(work1,nx,1)
         fact = 2.0/float(nx)
         j=2
         do 70 i = 3,nx-1,2
            work2(j) = sqrt( work1(i)**2 + work1(i+1)**2 )*fact
            j = j + 1
     70 continue
         work2(1) = abs(work1(1))*fact*0.5
         work2(j) = abs(work1(2))*fact*0.5
         call qlook(work2,ikplot+1,7.,15.,10.,10.,0.,rk,'k',1,'ex',2)
         do 12 i = 1, nx
```

```
            work1(i) = ey(i+1) /ree
   12 continue
      call realft(work1,nx,1)
      fact = 2.0/float(nx)
      j=2
      do 72 i = 3,nx-1,2
        work2(j) = sqrt( work1(i)**2 + work1(i+1)**2 )*fact
        j = j + 1
   72 continue
      work2(1) = abs(work1(1))*fact*0.5
      work2(j) = abs(work1(2))*fact*0.5
      call qlook(work2,ikplot+1,7.,2.,10.,10.,0.,rk,'k',1,'ey',2)
      do 14 i = 1, nx
            work1(i) = ez(i+1) /ree
   14 continue
      call realft(work1,nx,1)
      fact = 2.0/float(nx)
      j=2
      do 74 i = 3,nx-1,2
        work2(j) = sqrt( work1(i)**2 + work1(i+1)**2 )*fact
        j = j + 1
   74 continue
      work2(1) = abs(work1(1))*fact*0.5
      work2(j) = abs(work1(2))*fact*0.5
      call qlook(work2,ikplot+1,22.,15.,10.,10.,0.,rk,'k',1,'ez',2)
c
      do 20 i = 1, nx
            work1(i) = bz(i+1) /reb
   20 continue
      call realft(work1,nx,1)
      fact = 2.0/float(nx)
      j=2
      do 80 i = 3,nx-1,2
        work2(j) = sqrt( work1(i)**2 + work1(i+1)**2 )*fact
        j = j + 1
   80 continue
      work2(1) = abs(work1(1))*fact*0.5
      work2(j) = abs(work1(2))*fact*0.5
      call qlook(work2,ikplot+1,22.,2.,10.,10.,0.,rk,'k',1,'bz',2)
      call prmplt(25.,25.8,0.6,0.,'time',4,dt*itime,2)
      call chart
      return
```

```
      end
c*********************************************************************
      subroutine phsplt
#include "paramt.h"
c
      common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
      common /prtclc/ x(in), vx(in), vy(in),vz(in)
      common /ptprmc/ wp(is), qm(is), q(is), vpe(is), vpa(is),
     &                vd(is), pch(is), np(is)
      common /timecm/ itime,ntime,iecrct,iwrite,jobnum
      common /diagcm/ iediag, ifdiag, ikdiag, ipdiag, isdiag, ivdiag,
     &                ieplot, ifplot, ikplot, ipplot, isplot, ivplot
      common /otherc/ vmin,vmax
      common /resclc/ rex, ret, rev, ree, reb, rej, rer, res
      common /inputc/ dx, dt, cv, wc, angle
      common /rotatc/ sinth, costh
      dimension ipen(7)
      data ipen/3,5,3,2,4,7,1/
c
      ipmax = 16384
c
      v1=vmin
      v2=vmax
      if(vmin.eq.vmax)   call maxmin(vx,npt,v1,v2)
      if(mod(ipplot,2).eq.1) then
        xfact=20.0/slx
        yfact=20.0/(v2-v1)
        call newpen(3)
        call prmplt(16.,25.,0.8,0.,'time',4,itime*dt,2)
        call newpen(4)
        call xaxis1(8.,4.,20.,10,2,0.5,0.8,0.,slx/rex,3,'x',1)
        call xaxis1(8.,24.,20.,10,2,-0.5,0.0,0.,slx/rex,2,'x',1)
        call yaxis1(8.,4.,20.,10,2,0.5,0.8,v1/rev,v2/rev,3,'vx',2)
        call yaxis1(28.,4.,20.,10,2,-0.5,0.0,v1/rev,v2/rev,3,'vx',2)
        n2=0
        do 10 k=1,ns
          n1 = n2
          n2 = n1 + np(k)
          call newpen(ipen(k))
          n3 = np(k)/ipmax + 1
          do 20 i=n1+1,n2,n3
            xx = 8.0 + x(i)*xfact
```

```
            yy = 4.0 + (vx(i)-v1)*yfact
            call plot(xx,yy,3)
            call plot(xx+0.05,yy,2)
20        continue
10     continue
     call chart
     end if
     if(mod(ipplot,4).ge.2) then
       xfact=20.0/(v2-v1)
       yfact=20.0/(v2-v1)
       call newpen(5)
       call prmplt(26.,25.,0.8,0.,'time',4,itime*dt,2)
       call newpen(4)
       call xaxis1(10.,4.,20.,10,2,0.5,0.7,v1/rev,v2/rev,3,
    &                    'v-perp/xy',9)
       call xaxis1(10.,24.,20.,10,2,-0.5,0.0,v1/rev,v2/rev,3,'p',1)
       call yaxis1(10.,4.,20.,10,2,0.5,0.7,v1/rev,v2/rev,3,'vz',2)
       call yaxis1(30.,4.,20.,10,2,-0.5,0.0,v1/rev,v2/rev,3,'vz',2)
       n2=0
       do 30 k=1,ns
         n1 = n2
         n2 = n1 + np(k)
         call newpen(ipen(k))
         do 40 i=n1+1,n2
           vpx = costh*vx(i) + sinth*vy(i)
           vpy =-sinth*vx(i) + costh*vy(i)
           xx = 10.0 + (vpy-v1)*xfact
           yy = 4.0 + (vz(i)-v1)*yfact
           call plot(xx,yy,3)
           call plot(xx+0.05,yy,2)
40        continue
30     continue
     call chart
     end if
     if(mod(ipplot,8).ge.4) then
       xfact=28.0/(v2-v1)
       yfact=14.0/v2
       call newpen(5)
       call prmplt(24.,23.,0.8,0.,'time',4,itime*dt,2)
       call newpen(4)
       call xaxis1(5.,8.,28.,20,4,0.5,0.7,v1/rev,v2/rev,3,'v-para',6)
       call xaxis1(5.,22.,28.,20,4,-0.5,0.0,v1/rev,v2/rev,3,'v-para',6)
```

```
          call yaxis1(5.,8.,14.,10,2,0.5,0.7,0.,v2/rev,3,'v-perp',6)
          call yaxis1(33.,8.,14.,10,2,-0.5,0.0,0.,v2/rev,3,'v-perp',6)
          n2=0
          do 50 k=1,ns
            n1 = n2
            n2 = n1 + np(k)
            call newpen(ipen(k))
            do 60 i=n1+1,n2
              vpx = costh*vx(i) + sinth*vy(i)
              vpy =-sinth*vx(i) + costh*vy(i)
              vperp = sqrt(vpy*vpy+vz(i)*vz(i))
              xx = 5.0 + (vpx-v1)*xfact
              yy = 8.0 + vperp*yfact
              call plot(xx,yy,3)
              call plot(xx+0.05,yy,2)
   60       continue
   50     continue
        call chart
        end if
        return
        end
c**********************************************************************
        subroutine pltprm
#include "paramt.h"
        common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
        common /ptprmc/ wp(is), qm(is), q(is), vpe(is), vpa(is),
       &                vd(is), pch(is), np(is)
        common /timecm/ itime,ntime,iecrct,iwrite,jobnum
        common /diagcm/ iediag, ifdiag, ikdiag, ipdiag, isdiag, ivdiag,
       &                ieplot, ifplot, ikplot, ipplot, isplot, ivplot
        common /otherc/ vmin,vmax
        common /inputc/ dx, dt, cv, wc, angle
c-----------------------------------------------------------
        h=0.6
        call newpen(3)
        call symbol(1.,25.,1.,'kempo1 parameters',0.,17)
        call prmplt(2.,     23.,h,0.,'nx      ',6,float(nx      ),0)
        call prmplt(-999.,23.,h,0.,'ntime ',6,float(ntime ),0)
        call prmplt(-999.,23.,h,0.,'iecrct',6,float(iecrct),0)
        call prmplt(-999.,23.,h,0.,'cv      ',6,cv                ,2)
        call prmplt(-999.,23.,h,0.,'dx      ',6,dx                ,2)
        call prmplt(-999.,23.,h,0.,'dt      ',6,dt                ,2)
```

```
      call prmplt(-999.,23.,h,0.,'wc       ',6,wc                ,2)
      call prmplt(-999.,23.,h,0.,'angle ',6,angle             ,2)
      call prmplt(    13.,23.,h,0.,'iediag',6,float(iediag),0)
      call prmplt(-999.,23.,h,0.,'isdiag',6,float(isdiag),0)
      call prmplt(-999.,23.,h,0.,'ifdiag',6,float(ifdiag),0)
      call prmplt(-999.,23.,h,0.,'ikdiag',6,float(ikdiag),0)
      call prmplt(-999.,23.,h,0.,'ivdiag',6,float(ivdiag),0)
      call prmplt(-999.,23.,h,0.,'ipdiag',6,float(ipdiag),0)
      call prmplt(-999.,23.,h,0.,'iwrite',6,float(iwrite),0)
      call prmplt(-999.,23.,h,0.,'jobnum',6,float(jobnum),0)
      call prmplt(    24.,23.,h,0.,'ieplot',6,float(ieplot),0)
      call prmplt(-999.,23.,h,0.,'isplot',6,float(isplot),0)
      call prmplt(-999.,23.,h,0.,'ifplot',6,float(ifplot),0)
      call prmplt(-999.,23.,h,0.,'ikplot',6,float(ikplot),0)
      call prmplt(-999.,23.,h,0.,'ivplot',6,float(ivplot),0)
      call prmplt(-999.,23.,h,0.,'ipplot',6,float(ipplot),0)
      do 10 i=1,ns
        x=2.+10.*(i-1)
        y=12.
        call symbol(x,y,h,'species',0.,7)
        call number(x+0.8*8,y,h,float(i),0.,-1)
        y=10.
        call prmplt(x       ,y,h,0.,'np ',3,float(np(i)),0)
        call prmplt(-999.,y,h,0.,'qm ',3,qm(i)           ,2)
        call prmplt(-999.,y,h,0.,'wp ',3,wp(i)           ,2)
        call prmplt(-999.,y,h,0.,'vd ',3,vd(i)           ,2)
        call prmplt(-999.,y,h,0.,'pch',3,pch(i)          ,2)
        call prmplt(-999.,y,h,0.,'vpa',3,vpa(i)          ,2)
        call prmplt(-999.,y,h,0.,'vpe',3,vpe(i)          ,2)
   10 continue
      call chart
      return
      end
c**********************************************************************
      subroutine positn
#include "paramt.h"
      common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
      common /prtclc/ x(in), vx(in), vy(in),vz(in)
c
      do 100 i = 1, npt
        x(i) = x(i)+vx(i)
        if(x(i).lt.0.0) x(i) = x(i)+slx
```

```
          if(x(i).ge.slx) x(i) = x(i)-slx
  100 continue
      return
      end
c**********************************************************************
      subroutine reader
#include "paramt.h"
      common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
      common /fieldc/ ex(ix), ey(ix), ez(ix), by(ix), bz(ix),
     &                ajx(ix), ajy(ix), ajz(ix), rho(ix)
      common /prtclc/ x(in), vx(in), vy(in),vz(in)
      common /ptprmc/ wp(is), qm(is), q(is), vpe(is), vpa(is),
     &                vd(is), pch(is), np(is)
      common /ecrctc/ rkfact(ix)
      common /resclc/ rex, ret, rev, ree, reb, rej, rer, res
      common /inputc/ dx, dt, cv, wc, angle
      common /rotatc/ sinth, costh
      common /timecm/ itime,ntime,iecrct,iwrite,jobnum
      common /diagcm/ iediag, ifdiag, ikdiag, ipdiag, isdiag, ivdiag,
     &                ieplot, ifplot, ikplot, ipplot, isplot, ivplot
      common /otherc/ vmin,vmax
c
      ind=90
      open(ind,file='kempo1.cont',status='old',
     &     form='unformatted',access='sequential')
      read(ind) jx,js,jn,itime,jtime,jecrct,jwrite,jobnum
      if(jx.ne.ix)  go to 10
      if(js.ne.is)  go to 10
      if(jn.ne.in)  go to 10
      read(ind) jediag,jfdiag,jkdiag,jpdiag,jsdiag,jvdiag
      read(ind) jeplot,jfplot,jkplot,jpplot,jsplot,jvplot
      read(ind) tcs,bx0,rho0,slx,nx,nxp1,nxp2,npt,ns
      read(ind) dx,dt,cv,wc,angle,sinth,costh,vmin,vmax
      read(ind) rex,ret,rev,ree,reb,rej,rer,res
      read(ind) wp,qm,q,vpe,vpa,vd,pch,np
      read(ind) ex,ey,ez,by,bz,ajx,ajy, ajz,rho,rkfact
      read(ind) x,vx,vy,vz
      jobnum=jobnum+1
      close(ind)
      return
   10 continue
      close(ind)
```

```
      write(0,*) 'inconsistent parameters : ix,is,in'
      write(0,*) 'ix=',jx,'is=',js,'in=',jn
      stop
      end
c**********************************************************************
      subroutine renorm
#include "paramt.h"
      common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
      common /ptprmc/ wp(is), qm(is), q(is), vpe(is), vpa(is),
     &                vd(is), pch(is), np(is)
      common /otherc/ vmin,vmax
      common /resclc/ rex, ret, rev, ree, reb, rej, rer, res
      common /inputc/ dx, dt, cv, wc, angle
c
c-- distance
      rex = 1.0/dx
c-- time
      ret = 2.0/dt
c-- velocity
      rev = rex/ret
c-- electric field, charge, and mass
      ree = rex/(ret*ret)
c-- magnetic field
      reb = 1.0/ret
c-- current density
      rej = rex/(ret*ret*ret)
c-- charge density
      rer = 1.0/(ret*ret)
c-- energy density
      res = (rex*rex)/(ret*ret*ret*ret)
c
      vmin = vmin*rev
      vmax = vmax*rev
      cv   = cv  *rev
      wc   = wc  /ret
c
      do 10 k=1,ns
        wp(k)  = wp(k) /ret
        vpe(k) = vpe(k)*rev
        vpa(k) = vpa(k)*rev
        vd(k)  = vd(k) *rev
   10 continue
```

```
      return
      end
c**********************************************************************
      subroutine spectr
#include "paramt.h"
      parameter(imax=64,jmax=2048,icomp=5)
      common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
      common /fieldc/ ex(ix), ey(ix), ez(ix), by(ix), bz(ix),
     &                ajx(ix), ajy(ix), ajz(ix), rho(ix)
      common /prtclc/ x(in), vx(in), vy(in),vz(in)
      common /ptprmc/ wp(is), qm(is), q(is), vpe(is), vpa(is),
     &                vd(is), pch(is), np(is)
      common /timecm/ itime,ntime,iecrct,iwrite,jobnum
      common /diagcm/ iediag, ifdiag, ikdiag, ipdiag, isdiag, ivdiag,
     &                ieplot, ifplot, ikplot, ipplot, isplot, ivplot
      common /resclc/ rex, ret, rev, ree, reb, rej, rer, res
      common /inputc/ dx, dt, cv, wc, angle
      common /rotatc/ sinth, costh
      common /work1c/ work1(ix),work2(ix)
      common /work2c/ wk2d(imax,jmax,icomp)
      dimension workt(jmax)
      character*2 comp(5)
      dimension xp(4),yp(4)
      save comp,xp,yp,ic
      data comp/'ex','ey','ez','by','bz'/
      data xp / 7.,22., 7.,22./
      data yp /16.,16., 2., 2./
      data ic/0/
c
      minmod = 1
      maxmod = 4
      ikp = 16
      iwp = 512
      ifb = 0
c
      if(maxmod.ge.imax/2) maxmod = imax/2-1
      if(ic.eq.0) t1=dt*itime
      nxrd=imax
      if(nx.lt.nxrd) nxrd=nx
      ic=ic+1
      ncp=1
      do 10 i=1,nx
```

```
          work1(i)=ex(i+1) /ree
   10   continue
        call skfft(work1,nx,wk2d(1,ic,ncp),nxrd)
        if(ncp.eq.icomp) goto 60
        ncp=ncp+1
        do 20 i=1,nx
          work1(i)=ey(i+1) /ree
   20   continue
        call skfft(work1,nx,wk2d(1,ic,ncp),nxrd)
        if(ncp.eq.icomp) goto 60
        ncp=ncp+1
        do 30 i=1,nx
          work1(i)=ez(i+1) /ree
   30   continue
        call skfft(work1,nx,wk2d(1,ic,ncp),nxrd)
        if(ncp.eq.icomp) goto 60
        ncp=ncp+1
        by0    = wc/qm(1)*sinth
        do 40 i=1,nx
          work1(i)=(by(i+1)-by0) /reb
   40   continue
        call skfft(work1,nx,wk2d(1,ic,ncp),nxrd)
        if(ncp.eq.icomp) goto 60
        ncp=ncp+1
        do 50 i=1,nx
          work1(i)=bz(i+1) /reb
   50   continue
        call skfft(work1,nx,wk2d(1,ic,ncp),nxrd)
   60   if(ic.ne.isplot.and.ic.ne.jmax) return
        dtime=dt*isdiag*ic
        t2=dt*itime
c
        do 80 ncp = 1, icomp
          i = (minmod-1)*2 + 1
          do 75 k = minmod, maxmod
            i = i + 2
            m = mod(k-minmod,4) + 1
            do 70 j=1,ic
              workt(j) = sqrt(wk2d(i,j,ncp)**2 + wk2d(i+1,j,ncp)**2)
    70      continue
            call qlook(workt,ic,xp(m),yp(m),10.,10.,
     &                    t1,t2,'time',4,comp(ncp),2)
```

```
            call newpen(5)
            call prmplt(xp(m)+6.,yp(m)+10.5,0.5,0.,
     &                  'mode',4,float(k),0)
            if(m.eq.4) call chart
   75     continue
   80 continue
        if(m.ne.4) call chart
c
        if(ikp.gt.nxrd/2) ikp = nxrd/2
        if(iwp.gt.ic/2) iwp = ic/2
        do 90 ncp=1,icomp
          call newpen(5)
          call symbol(1.,25.5,1.0,comp(ncp),0.,2)
          call newpen(3)
          call wkfft(wk2d(1,1,ncp),imax,jmax,nxrd,ic,work1,workt,1)
          call wkplot(wk2d(1,1,ncp),imax,jmax,nxrd,ic,work1,workt,
     &                  8.,5.,20.,20.,slx/rex,dtime,ikp,iwp,ifb)
          call prmplt(29.,22.,0.8,0.,'t1',2,t1,2)
          call prmplt(29.,20.,0.8,0.,'t2',2,t2,2)
          call chart
   90   continue
        ic = 0
        return
        end
c**********************************************************************
        subroutine vdsplt
#include "paramt.h"
        common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
        common /prtclc/ x(in), vx(in), vy(in),vz(in)
        common /ptprmc/ wp(is), qm(is), q(is), vpe(is), vpa(is),
     &                  vd(is), pch(is), np(is)
        common /diagcm/ iediag, ifdiag, ikdiag, ipdiag, isdiag, ivdiag,
     &                  ieplot, ifplot, ikplot, ipplot, isplot, ivplot
        common /timecm/ itime,ntime,iecrct,iwrite,jobnum
        common /inputc/ dx, dt, cv, wc, angle
        common /otherc/ vmin,vmax
        common /resclc/ rex, ret, rev, ree, reb, rej, rer, res
        common /work1c/ work1(ix),work2(ix)
c
        nv = 101
        if(nv.gt.ix) nv = ix
        n2 = 0
```

```
      do 10 k=1,ns
        n1 = n2
        n2 = n1 + np(k)
        v1 = vmin
        v2 = vmax
        if(vmin.eq.vmax) call maxmin(vx(n1+1),np(k),v1,v2)
        dvi = float(nv-1)/(v2-v1)
      do 20 i = 1,nv
         work1(i) = 0.
20       continue
      do 30 m = n1+1, n2
        if(vx(m).lt.v1.or.vx(m).ge.v2) go to 30
          rv = (vx(m)-v1)*dvi + 1.0
          i  = rv
          s2 = rv - i
          s1 = 1.0 - s2
          work1(i)   = work1(i)   + s1
          work1(i+1) = work1(i+1) + s2
30    continue
      do 40 i = 1,nv
          work1(i) = work1(i)*dvi*rev/float(np(k))
40    continue
      call qlook(work1,nv,7.,15.5,10.,10.,v1/rev,v2/rev,
     &                          'vx',20002,'f(vx)',5)
      if(vmin.eq.vmax) call maxmin(vy(n1+1),np(k),v1,v2)
      dvi = float(nv-1)/(v2-v1)
      do 22 i = 1,nv
         work1(i) = 0.
22       continue
      do 32 m = n1+1, n2
        if(vy(m).lt.v1.or.vy(m).ge.v2) go to 32
          rv = (vy(m)-v1)*dvi + 1.0
          i  = rv
          s2 = rv - i
          s1 = 1.0 - s2
          work1(i)   = work1(i)   + s1
          work1(i+1) = work1(i+1) + s2
32    continue
      do 42 i = 1,nv
          work1(i) = work1(i)*dvi*rev/float(np(k))
42    continue
      call qlook(work1,nv,7.,2.,10.,10.,v1/rev,v2/rev,
```

```
     &                        'vy',20002,'f(vy)',5)
      if(vmin.eq.vmax) call maxmin(vz(n1+1),np(k),v1,v2)
      dvi = float(nv-1)/(v2-v1)
      do 24 i = 1,nv
         work1(i) = 0.
24    continue
      do 34 m = n1+1, n2
        if(vz(m).lt.v1.or.vz(m).ge.v2) go to 34
          rv = (vz(m)-v1)*dvi + 1.0
          i  = rv
          s2 = rv - i
          s1 = 1.0 - s2
          work1(i)   = work1(i)   + s1
          work1(i+1) = work1(i+1) + s2
34  continue
      do 44 i = 1,nv
          work1(i) = work1(i)*dvi*rev/float(np(k))
44  continue
      call qlook(work1,nv,22.,2.,10.,10.,v1/rev,v2/rev,
     &                        'vz',20002,'f(vz)',5)
      call newpen(5)
      call symbol(24.,24.,0.8,'species',0.,7)
      call number(30.4,24.,0.8,float(k),0.,-1)
      call prmplt(24.,22.,0.8,0.,'time',4,itime*dt,2)
      call chart
10  continue
        return
        end
c**********************************************************************
      subroutine velcty
#include "paramt.h"
      common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
      common /fieldc/ ex(ix), ey(ix), ez(ix), by(ix), bz(ix),
     #                ajx(ix), ajy(ix), ajz(ix), rho(ix)
      common /prtclc/ x(in), vx(in), vy(in),vz(in)
      common /ptprmc/ wp(is), qm(is), q(is), vpe(is), vpa(is),
     #                vd(is), pch(is), np(is)
      common /work1c/ work1(ix),work2(ix)
c
      do 100 i = 2, nxp1
        work1(i) = 0.5 * ( ex(i-1) + ex(i) )
100   continue
```

```
c
      work1(nxp2) = work1(2)
c
      do 110 i = 2, nxp1
        work2(i) = 0.5 * ( by(i+1) + by(i) )
  110 continue
c
      work2(1) = work2(nxp1)
c
      n2=0
      do 210 k=1,ns
         n1   = n2
         n2   = n1 + np(k)
         bx1 = bx0*qm(k)
         const = 1.0 + bx1*bx1
c
      do 200 m = n1+1, n2
c
            i    =   x(m) + 2.0
            sf2 = (x(m) + 2.0 - i)*qm(k)
            sf1 = qm(k) - sf2
            ih   =   x(m) + 1.5
            sh2 = (x(m) + 1.5 - ih)*qm(k)
            sh1 = qm(k) - sh2
            i1   = i + 1
            ih1 = ih+ 1
        ex1 = sf1*work1(i)   + sf2*work1(i1)
        ey1 = sf1*ey(i)        + sf2*ey(i1)
        ez1 = sh1*ez(ih)       + sh2*ez(ih1)
        by1 = sh1*work2(ih) + sh2*work2(ih1)
        bz1 = sh1*bz(ih)       + sh2*bz(ih1)
c
                  boris = 2./(const + by1*by1 + bz1*bz1)
c
                  vx(m) = vx(m) + ex1
                  vy(m) = vy(m) + ey1
                  vz(m) = vz(m) + ez1
c
                  vxt      = vx(m) + vy(m)*bz1 - vz(m)*by1
                  vyt      = vy(m) + vz(m)*bx1 - vx(m)*bz1
                  vzt      = vz(m) + vx(m)*by1 - vy(m)*bx1
c
```

```
                vx(m) = vx(m) + boris*(vyt*bz1 - vzt*by1)
                vy(m) = vy(m) + boris*(vzt*bx1 - vxt*bz1)
                vz(m) = vz(m) + boris*(vxt*by1 - vyt*bx1)
c
                vx(m) = vx(m) + ex1
                vy(m) = vy(m) + ey1
                vz(m) = vz(m) + ez1
c
  200 continue
  210 continue
      return
      end
c**********************************************************************
      subroutine writer
#include "paramt.h"
      common /constc/ tcs, bx0, rho0, slx, nx, nxp1, nxp2, npt, ns
      common /fieldc/ ex(ix), ey(ix), ez(ix), by(ix), bz(ix),
     &                ajx(ix), ajy(ix), ajz(ix), rho(ix)
      common /prtclc/ x(in), vx(in), vy(in),vz(in)
      common /ptprmc/ wp(is), qm(is), q(is), vpe(is), vpa(is),
     &                vd(is), pch(is), np(is)
      common /ecrctc/ rkfact(ix)
      common /resclc/ rex, ret, rev, ree, reb, rej, rer, res
      common /inputc/ dx, dt, cv, wc, angle
      common /rotatc/ sinth, costh
      common /timecm/ itime,ntime,iecrct,iwrite,jobnum
      common /diagcm/ iediag, ifdiag, ikdiag, ipdiag, isdiag, ivdiag,
     &                ieplot, ifplot, ikplot, ipplot, isplot, ivplot
      common /otherc/ vmin,vmax
c
      if(jobnum.eq.0) return
      ind=90
      jx = ix
      js = is
      jn = in
      open(ind,file='kempo1.cont',status='unknown',
     &     form='unformatted',access='sequential')
      write(ind) jx,js,jn,itime,ntime,iecrct,iwrite,jobnum
      write(ind) iediag,ifdiag,ikdiag,ipdiag,isdiag,ivdiag
      write(ind) ieplot,ifplot,ikplot,ipplot,isplot,ivplot
      write(ind) tcs,bx0,rho0,slx,nx,nxp1,nxp2,npt,ns
      write(ind) dx,dt,cv,wc,angle,sinth,costh,vmin,vmax
```

```
      write(ind) rex,ret,rev,ree,reb,rej,rer,res
      write(ind) wp,qm,q,vpe,vpa,vd,pch,np
      write(ind) ex,ey,ez,by,bz,ajx,ajy, ajz,rho,rkfact
      write(ind) x,vx,vy,vz
      close(ind)
      return
      end
c---------- "paramt.h" to be included ----------
      parameter(ix=1026, is=3, in=32768)
================== end of kempo1.f ================================
  ============ beginning of libkempo.f ==========================
C*****************************************************************
C       LIBKEMPO1 :      Library for KEMPO1
C                arranged by
C                        Yoshiharu Omura
C                        Radio Atmospheric Science Center
C                        Kyoto University
C                        Uji, Kyoto 611     Japan
C                            E-mail: omura@kurasc.kyoto-u.ac.jp
C                            FAX:    +81-774-31-8463
C
C               Version 1.2                 October 7, 1992
C*****************************************************************
C
C       ************************
      REAL FUNCTION ASCALE(X,RNI)
C       ************************
C       PROGRAMED BY Y. OMURA
C
      IF(ABS(RNI).GE.1000) THEN
        RN=INT(RNI/1000.)*(ABS(RNI)-1000.)
        ARN=ABS(RN)
        ICONT=1
      ELSE
        RN=RNI
        ARN=ABS(RN)
        ICONT=0
      ENDIF
      IF(RN.NE.0.) GO TO 10
      ASCALE=X
      RETURN
   10 CALL ETRANS(X,A,NP)
```

```
      AA=ABS(A)
      IF(ICONT.EQ.1.AND.A.LT.0.) THEN
      IS=1
      IF(X.LT.0.) IS=-1
      OPT=INT(AA/ARN)*ARN
      IF(RN.LT.0.) OPT=OPT+ARN
      ASCALE=(OPT*10.**NP)*FLOAT(IS)
      ELSE
      IS=1
      IF(X.LT.0.) IS=-1
      OPT=INT(AA/ARN)*ARN+ARN
      IF(RN.LT.0.) OPT=OPT-ARN
      ASCALE=(OPT*10.**NP)*FLOAT(IS)
      ENDIF
      RETURN
      END
C     ***************************
      SUBROUTINE DPLOT(XP,YP,IPEN)
      ENTRY DPLOT6(XP,YP,IPEN,DA,BA,PA)
C     ***************************
C     COPIED FROM XYGRAPH BY T. SATO, DEPT. OF E.E., KYOTO UNIV.
C     MODIFIED BY Y. OMURA
C
      DIMENSION A(102),M(102)
      DATA MOVE /-1/
      DATA MODE/-1/
      DATA XPM,YPM,D,B,P /5*0./
C
      IF(IPEN.LE.1) GO TO 100
      IF(MODE.GE.0) GO TO 200
      CALL PLOT(XP,YP,IPEN)
      RETURN
  100 N=6
      IF(IPEN.EQ.-999) THEN
      XP=XPM
      YP=YPM
      IPEN=-MODE
      DA=D
      BA=B
      PA=P
      RETURN
      END IF
```

```
      MODE=-IPEN
      CALL PLOT(XP,YP,3)
      IF(MODE.EQ.-1) RETURN
      IF(MODE.GT.50) MODE=50
      D=0.5
      IF(N.GE.4) D=DA
      B=D*0.5
      IF(MODE.GE.1) B=D*0.2
      IF(N.GE.5) B=BA
      P=B
      IF(N.GE.6) P=PA
      XPM=XP
      YPM=YP
      T=0.
      NA=(MODE+1)*2
      A(1)=D
      A(2)=D+B
      C=A(2)
      M(1)=2
      M(2)=3
      IF(MODE.EQ.0) RETURN
      DO 110 I=3,NA,2
      A(I)=A(I-1)+P
      A(I+1)=A(I)+B
      M(I)=2
      M(I+1)=3
  110 CONTINUE
      C=A(NA)
      RETURN
  200 IF(XP.EQ.XPM.AND.YP.EQ.YPM) RETURN
      S=SQRT((XP-XPM)**2+(YP-YPM)**2)
      IF(IPEN.EQ.2) GO TO 300
      IF(MOVE.EQ.1) T=AMOD(S+T,C)
      IF(MOVE.EQ.-1) T=0.
      XPM=XP
      YPM=YP
      CALL PLOT(XP,YP,3)
      RETURN
  300 DX=(XP-XPM)/S
      DY=(YP-YPM)/S
      DO 10 I=1,NA
      IF(A(I).GE.T) GO TO 20
```

```
 10 CONTINUE
    I=NA
 20 IF(T+S.GT.C) GO TO 400
    DO 210 J=I,NA
    IF(T+S.LE.A(J)) GO TO 220
    CALL PLOT(XPM+DX*(A(J)-T),YPM+DY*(A(J)-T),M(J))
210 CONTINUE
    J=NA
220 CALL PLOT(XP,YP,M(J))
    T=T+S
    XPM=XP
    YPM=YP
    RETURN
400 DO 30 J=I,NA
 30 CALL PLOT(XPM+DX*(A(J)-T),YPM+DY*(A(J)-T),M(J))
    XPM=XPM+DX*(C-T)
    YPM=YPM+DY*(C-T)
    S=S+T-C
    T=0.
    L=S/C
    IF(L.EQ.0) GO TO 500
    S=S-L*C
    DO 40 I=1,L
    DO 50 J=1,NA
 50 CALL PLOT(XPM+DX*A(J),YPM+DY*A(J),M(J))
    XPM=XPM+DX*C
    YPM=YPM+DY*C
 40 CONTINUE
500 DO 60 I=1,NA
    IF(A(I).GE.S) GO TO 70
 60 CONTINUE
    I=NA
 70 IF(I.EQ.1) GO TO 80
    DO 90 J=1,I-1
 90 CALL PLOT(XPM+DX*A(J),YPM+DY*A(J),M(J))
 80 CALL PLOT(XP,YP,M(I))
    T=S
    XPM=XP
    YPM=YP
    RETURN
    END
C       *********************************
```

```
      SUBROUTINE ENUMBR(X,Y,H,R,ANGL,N)
C     *********************************
C     PROGRAMED BY Y. OMURA
C
      R0=ABS(R)
      IS=1
      IF(R.LT.0.) IS=-1
      I=0
      IF(R0.EQ.0.) GO TO 40
   10 IF(R0.GE.1.) GO TO 20
      I=I-1
      R0=R0*10.
      GO TO 10
   20 IF(R0.LT.10.) GO TO 30
      I=I+1
      R0=R0/10.
      GO TO 20
   30 NP=2+N
      NRD=N
      IF(NRD.LT.1) NRD=1
      R0=RNDOFF(R0,NRD)
      IF(R0.LT.10.) GO TO 40
      I=I+1
      R0=R0/10.
   40 CONTINUE
      R0=R0*FLOAT(IS)
      EI=FLOAT(I)
      HF=H*0.8
      IF(N.EQ.-3) GO TO 70
      IF(N.LE.-2) GO TO 50
      CALL NUMBER(X,Y,H,R0,ANGL,N)
   50 IF(I.EQ.0) RETURN
      NP=2+N
      IF(IS.EQ.-1) NP=NP+1
      XP=X+H*FLOAT(NP)
      YP=Y
      CALL XYROT(XP,YP,X,Y,ANGL)
      CALL SYMBOL(XP,YP,HF,'X',ANGL,1)
      XP=X+H*FLOAT(NP)+HF
      YP=Y
      CALL XYROT(XP,YP,X,Y,ANGL)
      CALL SYMBOL(XP,YP,H,'10',ANGL,2)
```

```
      XP=X+H*FLOAT(NP+2)+HF
      YP=Y+0.5*H
      CALL XYROT(XP,YP,X,Y,ANGL)
      CALL NUMBER(XP,YP,HF,EI,ANGL,-1)
      RETURN
   70 CONTINUE
      XP=X
      YP=Y
      CALL SYMBOL(XP,YP,H,'10',ANGL,2)
      XP=X+H*2.
      YP=Y+0.5*H
      CALL XYROT(XP,YP,X,Y,ANGL)
      CALL NUMBER(XP,YP,HF,EI,ANGL,-1)
      RETURN
      END
C     ************************
      SUBROUTINE ETRANS(X,A,NP)
C     ************************
C     PROGRAMED BY Y. OMURA
C
      X0=ABS(X)
      IS=1
      IF(X.LT.0.) IS=-1
      A=0.
      NP=0
      I=0
      IF(X0.EQ.0.) RETURN
   10 IF(X0.GE.1.) GO TO 20
      I=I-1
      X0=X0*10.
      GO TO 10
   20 IF(X0.LT.9.999) GO TO 30
      I=I+1
      X0=X0/10.
      GO TO 20
   30 A=X0*FLOAT(IS)
      NP=I
      RETURN
      END
C*************************************************************
      SUBROUTINE FOUR1(DATA, N, ISIGN)
      REAL*8 WR,WI,WPR,WPI,WTEMP,THETA
```

```
      DIMENSION DATA(N)
      J=1
      DO 11 I=1,N,2
        IF(J.GT.I) THEN
          TEMPR=DATA(J)
          TEMPI=DATA(J+1)
          DATA(J)=DATA(I)
          DATA(J+1)=DATA(I+1)
          DATA(I)=TEMPR
          DATA(I+1)=TEMPI
        ENDIF
        M=N/2
    1   IF((M.GE.2).AND.(J.GT.M)) THEN
          J=J-M
          M=M/2
        GO TO 1
        ENDIF
        J=J+M
   11 CONTINUE
      MMAX=2
    2 IF(N.GT.MMAX) THEN
        ISTEP=2*MMAX
        THETA=6.28318530717959D0/(ISIGN*MMAX)
        WPR=-2.D0*DSIN(0.5D0*THETA)**2
        WPI=DSIN(THETA)
        WR=1.D0
        WI=0.D0
        DO 13 M=1,MMAX,2
          DO 12 I=M,N,ISTEP
            J=I+MMAX
            TEMPR=SNGL(WR)*DATA(J)-SNGL(WI)*DATA(J+1)
            TEMPI=SNGL(WR)*DATA(J+1)+SNGL(WI)*DATA(J)
            DATA(J)=DATA(I)-TEMPR
            DATA(J+1)=DATA(I+1)-TEMPI
            DATA(I)=DATA(I)+TEMPR
            DATA(I+1)=DATA(I+1)+TEMPI
   12     CONTINUE
          WTEMP=WR
          WR=WR*WPR-WI*WPI+WR
          WI=WI*WPR+WTEMP*WPI+WI
   13   CONTINUE
        MMAX=ISTEP
```

```
      GO TO 2
      ENDIF
      RETURN
      END
C     **********************************
      SUBROUTINE GNUMBR(X,Y,H,RNB,ANGL,N)
C     **********************************
C     PROGRAMED BY Y. OMURA
C
      RN=RNB
      AR=ABS(RN)
      IF(AR.NE.0.) GO TO 50
      CALL NUMBER(X,Y,H,RN,ANGL,1)
      RETURN
   50 CONTINUE
      ND=N-1
      IF(ND.LE.0) ND=-1
      NE=N+5
      IF(N.LE.1) NE=NE-1
      CALL ETRANS(AR,A,NP)
      NP1=NP+1
      IF(N.LT.0) GO TO 70
      IF(NP.LE.-1) THEN
        NC=N+1-NP
        IF(NC.LT.NE) THEN
      RN=RNDOFF(RN,NC-2)
        CALL NUMBER(X,Y,H,RN,ANGL,NC-2)
        ELSE
        CALL ENUMBR(X,Y,H,RN,ANGL,ND)
        END IF
      ELSE IF(NP1.GE.N) THEN
        IF(NP1.LT.NE) THEN
        CALL NUMBER(X,Y,H,RN,ANGL,-1)
        ELSE
        CALL ENUMBR(X,Y,H,RN,ANGL,ND)
        END IF
      ELSE IF(NP1.LT.N) THEN
      RN=RNDOFF(RN,N-NP1)
        CALL NUMBER(X,Y,H,RN,ANGL,N-NP1)
      END IF
      RETURN
   70 ND=-N-1
```

```
      IF(ND.EQ.0) ND=-1
      IF(NP1.GT.6) THEN
        RN=RNDOFF(RN,ND)
        CALL ENUMBR(X,Y,H,RN,ANGL,ND)
      ELSE
        CALL NUMBER(X,Y,H,RN,ANGL,-1)
      END IF
      RETURN
      END
C     ************************************************
      SUBROUTINE LXAXIS(XI,YI,XLI,DT,H,VMIN,VMAX,NTEXT,NT)
C     ************************************************
C     PROGRAMED BY Y. OMURA
C
      CHARACTER NTEXT*(*)
C
      AMAX=ABS(VMAX)
      AMIN=ABS(VMIN)
      IF(AMIN.GT.AMAX) THEN
        Y=YI
        X=XI+XLI
        XL=-XLI
        DUMMY=AMAX
        AMAX=AMIN
        AMIN=DUMMY
      ELSE
        X=XI
        Y=YI
        XL=XLI
      END IF
      CALL PLOT(X,Y,3)
      CALL PLOT(X+XL,Y,2)
      X1=ALOG10(AMIN)
      X2=ALOG10(AMAX)
      D=XL/(X2-X1)
      DTY=0.0
      DTH=DT*H
      HA=ABS(H)
      DF=ABS(D*0.3)
      IF(HA.GT.DF) HA=DF
      IF(DTH.LT.0.) DTY=ABS(DT)
      DSY=1.8*HA+DTY
```

```
      IF(H.LT.0.) DSY=-0.5*HA-DTY
      VS=ASCALE(AMIN,1.0)
      CCC=(VS-AMIN)/AMIN
      IF((CCC.GT.0.99).AND.(CCC.LT.1.01)) VS=AMIN
      CALL ETRANS(VS,XT,NP)
      I=INT(RNDOFF(XT,1))
      YY=Y-DSY
      DDX=X2-X1
      MAXNP=1
      IF(DDX.LT.1.5) GO TO 50
   10 IF(I.GE.10) THEN
         I=1
        NP=NP+1
      END IF
      V=FLOAT(I)*(10.**NP)
      IF(V.GT.AMAX) GO TO 20
      DD=DT*0.5
      XX=X+(ALOG10(V)-X1)*D
      IF(I.EQ.1) THEN
      ISFT=1
      CALL ETRANS(V,XT,MP)
      IF(MP.LT.0) ISFT=ISFT+1
      IAMP=IABS(MP)
      IF(IAMP.GE.10) ISFT=ISFT+1
      IF(ISFT.GT.MAXNP) MAXNP=ISFT
      XXP=XX-0.5*HA*(FLOAT(ISFT)*0.8+2.)
          DD=DT
        CALL ENUMBR(XXP,YY,HA,V,0.,-3)
      END IF
      CALL PLOT(XX,Y,3)
      CALL PLOT(XX,Y+DD,2)
      I=I+1
      GO TO 10
   50 CONTINUE
      DDS=0.05*D*0.66666
      HAH=HA*1.2
      IMAX=9
      IF(DDS.LT.HAH) THEN
        IMAX=8
      HAS=DDS
      ELSE
        HAS=HA*0.7
```

```
      END IF
      YYS=Y-0.5*HA-DTY-HAS
      IF(H.LT.0.) YYS=Y-DSY
   30 IF(I.GE.10) THEN
        I=1
       NP=NP+1
      END IF
      V=FLOAT(I)*(10.**NP)
      IF(V.GT.AMAX) GO TO 20
      DD=DT*0.5
      XX=X+(ALOG10(V)-X1)*D
      IF(I.EQ.1) THEN
         DD=DT
      ISFT=1
      CALL ETRANS(V,XT,MP)
      IF(MP.LT.0) ISFT=ISFT+1
      IAMP=IABS(MP)
      IF(IAMP.GE.10) ISFT=ISFT+1
      XXP=XX-0.5*HA*(FLOAT(ISFT)*0.8+2.)
        CALL ENUMBR(XXP,YY,HA,V,0.,-3)
      ELSE IF(I.LE.IMAX) THEN
      CALL NUMBER(XX-0.5*HAS,YYS,HAS,FLOAT(I),0.,-1)
      END IF
      CALL PLOT(XX,Y,3)
      CALL PLOT(XX,Y+DD,2)
      I=I+1
      GO TO 30
   20 CONTINUE
      ANT=FLOAT(NT)
      LINES=1
      YY=Y-3.7*HA-DTY
      IF(H.LT.0.) YY=Y+2.5*HA+DTY+(LINES-1)*1.2*HA*12./7.
      XX=X+XL*0.5-0.6*HA*ANT
      IF(NT.EQ.0) RETURN
      CALL SYMBOL(XX,YY,HA*1.2,NTEXT,0.,NT)
      RETURN
      END
C     ************************************************
      SUBROUTINE LYAXIS(XI,YI,YLI,DT,H,VMIN,VMAX,NTEXT,NT)
C     ************************************************
C
      CHARACTER NTEXT*(*)
```

```
C
      AMAX=ABS(VMAX)
      AMIN=ABS(VMIN)
      IF(AMIN.GT.AMAX) THEN
        X=XI
        Y=YI+YLI
        YL=-YLI
        DUMMY=AMAX
        AMAX=AMIN
        AMIN=DUMMY
      ELSE
        X=XI
        Y=YI
        YL=YLI
      END IF
      CALL PLOT(X,Y,3)
      CALL PLOT(X,Y+YL,2)
      Y1=ALOG10(AMIN)
      Y2=ALOG10(AMAX)
      D=YL/(Y2-Y1)
      DTX=0.0
      DTH=DT*H
      HA=ABS(H)
      IF(HA.GT.D) HA=ABS(DT)
      IF(DTH.LT.0.) DTX=ABS(DT)
      DSX=2.5*HA+DTX
      IF(H.LT.0.) DSX=-0.5*HA-DTX
      VS=ASCALE(AMIN,1.0)
      CCC=(VS-AMIN)/AMIN
      IF((CCC.GT.0.99).AND.(CCC.LT.1.01)) VS=AMIN
      CALL ETRANS(VS,XT,NP)
      I=INT(RNDOFF(XT,1))
      XX=X-DSX
      DDY=Y2-Y1
      MAXNP=1
      SGN=1.0
      IF(H.LT.0.) SGN=0.
      IF(DDY.LT.1.5) GO TO 50
   10 IF(I.GE.10) THEN
        I=1
       NP=NP+1
      END IF
```

```
      V=FLOAT(I)*(10.**NP)
      IF(V.GT.AMAX) GO TO 20
      DD=DT*0.5
      YY=Y+(ALOG10(V)-Y1)*D
      IF(I.EQ.1) THEN
      ISFT=1
      CALL ETRANS(V,XT,MP)
      IF(MP.LT.0) ISFT=ISFT+1
      IAMP=IABS(MP)
      IF(IAMP.GE.10) ISFT=ISFT+1
      IF(ISFT.GT.MAXNP) MAXNP=ISFT
      XXP=XX-0.8*HA*FLOAT(ISFT)*SGN
          DD=DT
        CALL ENUMBR(XXP,YY-0.5*HA,HA,V,0.,-3)
      END IF
      CALL PLOT(X,YY,3)
      CALL PLOT(X+DD,YY,2)
      I=I+1
      GO TO 10
   50 CONTINUE
      DDS=0.05*D*0.66666
      HAH=HA*0.9
      IMAX=9
      IF(DDS.LT.HAH) THEN
        IMAX=8
      HAS=DDS
      ELSE
        HAS=HA*0.7
      END IF
      XXS=X-0.5*HA-DTX-HAS
      IF(H.LT.0.) XXS=X-DSX
   30 IF(I.GE.10) THEN
        I=1
       NP=NP+1
      END IF
      V=FLOAT(I)*(10.**NP)
      IF(V.GT.AMAX) GO TO 20
      DD=DT*0.5
      YY=Y+(ALOG10(V)-Y1)*D
      IF(I.EQ.1) THEN
          DD=DT
      ISFT=1
```

```
      CALL ETRANS(V,XT,MP)
      IF(MP.LT.0) ISFT=ISFT+1
      IAMP=IABS(MP)
      IF(IAMP.GE.10) ISFT=ISFT+1
      IF(ISFT.GT.MAXNP) MAXNP=ISFT
      XXP=XX-0.8*HA*FLOAT(ISFT)*SGN
        CALL ENUMBR(XXP,YY-0.5*HA,HA,V,0.,-3)
      ELSE IF(I.LE.IMAX) THEN
      CALL NUMBER(XXS,YY-0.5*HAS,HAS,FLOAT(I),0.,-1)
      END IF
      CALL PLOT(X,YY,3)
      CALL PLOT(X+DD,YY,2)
      I=I+1
      GO TO 30
   20 CONTINUE
      ANT=FLOAT(NT)
      LINES=1
      XX=X-(FLOAT(MAXNP)*0.8+3.0)*HA-DTX-(LINES-1)*1.2*HA*12./7.
      IF(H.LT.0.) XX=X+(FLOAT(MAXNP)*0.8+4.4)*HA+DTX
      YY=Y+YL*0.5-0.6*HA*ANT
      IF(NT.EQ.0) RETURN
      CALL SYMBOL(XX,YY,HA*1.2,NTEXT,90.,NT)
      RETURN
      END
C     ***********************************
      SUBROUTINE MAXMIN (XXX,N,AMIN,AMAX)
C     ***********************************
      DIMENSION XXX(N)
C
      AMAX=XXX(1)
      AMIN=XXX(1)
      DO 10 I=1,N
      IF(XXX(I).GE.AMAX) AMAX=XXX(I)
      IF(XXX(I).LE.AMIN) AMIN=XXX(I)
   10 CONTINUE
C
      RETURN
      END
C     ************************************************
      SUBROUTINE PRMPLT(XI,YI,HI,ANGLI,NTEXT,NT,RN,NEN)
C     ************************************************
C     PROGRAMED BY Y. OMURA
```

```
C
      CHARACTER NTEXT*(*)
      DATA X,Y,H,ANGL/0.,0.,1.,0./
      IF((XI.EQ.999).OR.(YI.EQ.999.)) THEN
        X=X-H*SIN(ANGL/180.*3.14159)*2.0
        Y=Y+H*COS(ANGL/180.*3.14159)*2.0
      ELSE IF(XI.EQ.-999.) THEN
        X=X+H*SIN(ANGL/180.*3.14159)*2.0
        Y=Y-H*COS(ANGL/180.*3.14159)*2.0
      ELSE
        X=XI
        Y=YI
        H=HI
        ANGL=ANGLI
      ENDIF
      IAB=IABS(NEN)
      IF(IAB.GE.1000) THEN
      ID=INT(FLOAT(IAB)/100.)*100
      FACT=FLOAT(ID)/1000.
      NE=NEN/IAB*(IAB-ID)
      ELSE
      FACT=1.
      NE=NEN
      END IF
      CALL SYMBOL(X,Y,H*FACT,NTEXT,ANGL,NT)
      AT=FLOAT(NT)
      LINES=1
      XX=X+H*FACT*AT+H
      YY=Y
      CALL XYROT(XX,YY,X,Y,ANGL)
      CALL SYMBOL(XX,YY,H,'=',ANGL,1)
      XX=X+H*FACT*AT+H*3.
      YY=Y
      CALL XYROT(XX,YY,X,Y,ANGL)
      IF(NE.GT.0) THEN
      CALL GNUMBR(XX,YY,H,RN,ANGL,NE)
      ELSE IF( NE.EQ.0) THEN
      CALL NUMBER(XX,YY,H,RNDOFF(RN,-1),ANGL,-1)
      ELSE
      CALL NUMBER(XX,YY,H,RNDOFF(RN,-NE),ANGL,-NE)
      ENDIF
      RETURN
```

```
      END
C     ******************************************
      SUBROUTINE QLOOK(AR,N,X,Y,XAL,YAL,XMIN,XMAX,
     &                 NXTEXT,NXI,NYTEXT,NY)
C     ******************************************
C     PROGRAMED BY Y. OMURA, RASC/KYOTO UNIV.
C
      DIMENSION AR(N)
      CHARACTER NXTEXT*(*), NYTEXT*(*)
      COMMON /QLKCM1/IC,VMIN,VMAX,CRMIN,CRMAX,CX,CY,CXAL,CYAL
      DATA NFX,NFY,IPPEN,ITPEN /3,3,3,4/
      DATA DTMOD, HMOD /0.,0./
C
      NX=NXI
      IPLOT=0
      JPLOT=0
      KPLOT=0
      CALL NEWPEN(ITPEN)
      IF(IABS(NX).GE.20000) THEN
        KPLOT=1
        NX=NX/IABS(NX)*(IABS(NX)-20000)
      ENDIF
      IF(IABS(NX).GE.10000) THEN
        JPLOT=1
        NX=NX/IABS(NX)*(IABS(NX)-10000)
      ENDIF
      IF(IABS(NX).GE.1000.AND.IABS(NX).LT.10000) THEN
        IPLOT=1
        NX=NX/IABS(NX)*(IABS(NX)-1000)
      ENDIF
      CALL MAXMIN(AR,N,RMIN,RMAX)
      IF(RMIN.NE.0.) THEN
        A=(RMAX-RMIN)/RMIN
        IF(ABS(A).LT.1.E-5) THEN
      H=MAX(XAL,YAL)/22.
      CALL SYMBOL(X+XAL*0.2,Y+YAL*0.5,H,'NO VARIATION',0.,12)
      CALL PRMPLT(X+XAL*0.2,Y+YAL*0.4,H*0.8,0.,'VALUE',5,RMIN,3)
C       PRINT *,'+QLOOK+   NO VARIATION IN DATA : Y=',RMIN
      IPLOT=1
      ENDIF
      ENDIF
      CRMIN=RMIN
```

```
      VMIN=RMIN
      VMAX=RMAX
      CRMAX=RMAX
      CX=X
      CY=Y
      CXAL=XAL
      CYAL=YAL
      IC=0
      IF(JPLOT.EQ.1) GOTO 30
      IF(RMIN.GT.0..AND.NX.GE.0) THEN
        A=(RMAX-RMIN)/RMIN
        IF(A.GT.10..AND.NY.GE.0) THEN
          A=0.
          DO 10 I=1,N
   10     A=A+AR(I)
          A=A/FLOAT(N)
          DL=ABS(A-(RMIN+RMAX)*0.5)
          A=0.
          DO 20 I=1,N
   20     A=A+ALOG10(AR(I))
          A=A/FLOAT(N)
          DG=ABS(A-(ALOG10(RMIN)+ALOG10(RMAX))*0.5)
          IF(DG.LT.DL) IC=1
        ELSE
          IF(NY.LT.0) IC=1
        ENDIF
      END IF
      NXA=IABS(NX)
      NYA=IABS(NY)
      IF(IC.EQ.1) GO TO 30
      CALL ETRANS(RMIN,A,NP1)
      CALL ETRANS(RMAX,A,NP2)
      IF((RMIN.LT.0.).AND.(RMAX.GT.0.)) THEN
        AMIN=ABS(RMIN)
        AMAX=ABS(RMAX)
        IF(AMAX.GE.AMIN) THEN
        VMAX=ASCALE(RMAX,2.)
        A=2.*10.**(NP2-NP1)
        VMIN=ASCALE(RMIN,A)
        ELSE
        VMIN=ASCALE(RMIN,2.)
        A=2.*10.**(NP1-NP2)
```

```
      VMAX=ASCALE(RMAX,A)
      END IF
    ELSE IF(RMIN.EQ.0.) THEN
    VMIN=RMIN
    VMAX=ASCALE(RMAX,2.)
    ELSE IF(RMAX.EQ.0.) THEN
    VMIN=RMAX
    VMAX=ASCALE(RMIN,2.)
    ELSE IF(RMIN.GT.0.) THEN
      A=(RMAX-RMIN)/RMAX
      A=ASCALE(A,2.)*0.5
      VMAX=ASCALE(RMAX,A)
      A=A*10.**(NP2-NP1)
      VMIN=ASCALE(RMIN,-A)
    ELSE
      A=(RMIN-RMAX)/RMIN
    A=ASCALE(A,2.)*0.5
    VMAX=ASCALE(RMIN,A)
      A=A*10.**(NP1-NP2)
      VMIN=ASCALE(RMAX,-A)
    END IF
    IF(KPLOT.EQ.1) THEN
      VMIN=0.0
      IF(ABS(RMAX).GE.ABS(RMIN)) THEN
        VMAX=ASCALE(RMAX,2.)
       ELSE
        VMAX=ASCALE(RMIN,2.)
       ENDIF
      ENDIF
30 CONTINUE
    NXA=IABS(NX)
    NYA=IABS(NY)
    INX=INT(XAL)
    INX=INT(FLOAT(INX)/2.)*2
    INY=INT(YAL)
    INY=INT(FLOAT(INY)/2.)*2
    IF(DTMOD.EQ.0.) THEN
      DT=MAX(XAL,YAL)/30.
    ELSE
      DT = DTMOD
    ENDIF
    IF(HMOD.EQ.0.) THEN
```

```
   H=MAX(XAL,YAL)/22.
ELSE
   H = HMOD
ENDIF
IF(XMIN.EQ.0..AND.XMAX.GT.0.) THEN
IAXIS=1
CALL ETRANS(XMAX,A,NP)
A=RNDOFF(A,3)
IF(A.GE.8.) THEN
   DAX=1.
   LDX=4
ELSE IF(A.GE.6.) THEN
   DAX=0.5
   LDX=6
ELSE IF(A.GE.5.) THEN
   DAX=0.5
   LDX=5
ELSE IF(A.GE.4.) THEN
   DAX=0.5
   LDX=4
ELSE IF(A.GE.3.) THEN
   DAX=0.5
   LDX=3
ELSE IF(A.GE.2.) THEN
   DAX=0.2
   LDX=5
ELSE
   DAX=0.1
   LDX=5
ENDIF
   DAX=DAX*10.**NP/XMAX*XAL
   CALL XAXIS2(X,Y,XAL,DAX,LDX,DT,H,XMIN,XMAX,NFX,NXTEXT,NXA)
ELSE
CALL XAXIS1(X,Y,XAL,INX,2,DT,H,XMIN,XMAX,NFX,NXTEXT,NXA)
IAXIS=0
ENDIF
IF(IC.EQ.0) THEN
IF(IAXIS.EQ.1) THEN
   CALL XAXIS2(X,Y+YAL,XAL,DAX,LDX,-DT,0.,0.,0.,0,0,0)
ELSE
CALL XAXIS1(X,Y+YAL,XAL,INX,2,-DT,0.,0.,0.,0,0,0)
ENDIF
```

```
IF(VMIN.EQ.0..AND.VMAX.GT.0.) THEN
CALL ETRANS(VMAX,B,NQ)
B=RNDOFF(B,3)
IF(B.GE.8.) THEN
  DBY=1.
  LDY=4
ELSE IF(B.GE.6.) THEN
  DBY=0.5
  LDY=6
ELSE IF(B.GE.5.) THEN
  DBY=0.5
  LDY=5
ELSE IF(B.GE.4.) THEN
  DBY=0.5
  LDY=4
ELSE IF(B.GE.3.) THEN
  DBY=0.5
  LDY=3
ELSE IF(B.GE.2.) THEN
  DBY=0.2
  LDY=5
ELSE
  DBY=0.1
  LDY=5
ENDIF
  DBY=DBY*10.**NQ/(VMAX-VMIN)*YAL
CALL YAXIS2(X,Y,YAL,DBY,LDY,DT,H,VMIN,VMAX,NFY,NYTEXT,NYA)
CALL YAXIS2(X+XAL,Y,YAL,DBY,LDY,-DT,0.,0.,0.,0,0,0)
ELSE IF(ABS(VMIN).EQ.ABS(VMAX)) THEN
CALL ETRANS(VMAX,B,NQ)
B=RNDOFF(B,3)
IF(B.GE.8.) THEN
  DBY=2.
  LDY=4
ELSE IF(B.GE.6.) THEN
  DBY=1.
  LDY=6
ELSE IF(B.GE.5.) THEN
  DBY=1.
  LDY=5
ELSE IF(B.GE.4.) THEN
  DBY=1.0
```

```
  LDY=4
ELSE IF(B.GE.3.) THEN
  DBY=1.0
  LDY=3
ELSE IF(B.GE.2.) THEN
  DBY=0.5
  LDY=4
ELSE
  DBY=0.2
  LDY=5
ENDIF
  DBY=DBY*10.**NQ/(VMAX-VMIN)*YAL
CALL YAXIS2(X,Y,YAL,DBY,LDY,DT,H,VMIN,VMAX,NFY,NYTEXT,NYA)
CALL YAXIS2(X+XAL,Y,YAL,DBY,LDY,-DT,0.,0.,0.,0,0,0)
ELSE
CALL YAXIS1(X,Y,YAL,INY,2,DT,H,VMIN,VMAX,NFY,NYTEXT,NYA)
CALL YAXIS1(X+XAL,Y,YAL,INY,2,-DT,0.,0.,0.,0,0,0)
ENDIF
ELSE
CALL PLOT(X+XAL,Y,3)
CALL PLOT(X+XAL,Y+YAL,2)
CALL PLOT(X,Y+YAL,2)
RMIN0=RMAX*1.E-10
IF(RMIN.LT.RMIN0) THEN
  RMIN=RMIN0
  CRMIN=RMIN
ELSE
  CALL ETRANS(RMAX,RR1,NN1)
  CALL ETRANS(RMIN,RR2,NN2)
  IF(NN1.EQ.NN2) THEN
    RMIN=9.5*(10.**(NN2-1))
    CRMIN=RMIN
    IF(RR1.LT.2.1) THEN
      RMAX=2.1*(10.**NN1)
      CRMAX=RMAX
    END IF
  END IF
END IF
CALL LYAXIS(X,Y,YAL,-DT,H,RMIN,RMAX,NYTEXT,NYA)
END IF
CALL NEWPEN(IPPEN)
IF(IPLOT.NE.1) CALL QLOOK2(AR,N,1)
```

```
      RETURN
C     ********************
      ENTRY QLKMOD(NEX,NEY,JPPEN,JTPEN)
C     ********************
C
      NFX=NEX
      NFY=NEY
      IPPEN=JPPEN
      ITPEN=JTPEN
C
      RETURN
C
C     ********************
      ENTRY QLKMD2(DTIN,HIN)
C     ********************
C
      DTMOD = DTIN
      HMOD = HIN
C
      RETURN
      END
C     *************************
      SUBROUTINE QLOOK2(BR,N,IP)
C     *************************
C        PROGRAMED BY Y. OMURA
C
      DIMENSION BR(N)
      COMMON /QLKCM1/IC,VMIN,VMAX,RMIN,RMAX,X,Y,XAL,YAL
C
      DD=XAL/40.
      IF(IP.LE.-1) DD=XAL/10.
      BD=XAL/80.
      PD=XAL/100.
      IF(IC.EQ.0) THEN
      YFACT=YAL/(VMAX-VMIN)
      DDX=XAL/FLOAT(N-1)
      YMAX=Y+YAL
      XX=X
      YY=Y+(BR(1)-VMIN)*YFACT
      IF(YY.GT.YMAX) YY=YMAX
      IF(YY.LT.Y)    YY=Y
      CALL DPLOT6(XX,YY,IP,DD,BD,PD)
```

```
      IPEN=3
      DO 10 I=2,N
      XX=XX+DDX
      IF(BR(I).EQ.999.) GO TO 10
      YY=Y+(BR(I)-VMIN)*YFACT
      IF(BR(I-1).EQ.999.) THEN
        IP2=3
      ELSE
        IP2=2
      END IF
      IF(YY.GT.YMAX) THEN
        YY=YMAX
        IF(IPEN.EQ.2) THEN
          CALL DPLOT(XX,YY,IP2)
          IPEN=3
        ELSE
          CALL DPLOT(XX,YY,3)
        ENDIF
      ELSE IF(YY.LT.Y) THEN
        YY=Y
        IF(IPEN.EQ.2) THEN
          CALL DPLOT(XX,YY,IP2)
          IPEN=3
        ELSE
          CALL DPLOT(XX,YY,3)
        ENDIF
      ELSE
        CALL DPLOT(XX,YY,IP2)
        IPEN=2
      ENDIF
   10 CONTINUE
      ELSE
      ALR=ALOG10(RMIN)
      YFACT=YAL/(ALOG10(RMAX)-ALR)
      DDX=XAL/FLOAT(N-1)
      YMAX=Y+YAL
      XX=X
      YY=Y+(ALOG10(BR(1))-ALR)*YFACT
      IF(YY.GT.YMAX) YY=YMAX
      IF(YY.LT.Y)    YY=Y
      CALL DPLOT6(XX,YY,IP,DD,BD,PD)
      IPEN=3
```

```
      DO 20 I=2,N
      IF(BR(I).EQ.999.) GO TO 20
      XX=XX+DDX
      YY=Y+(ALOG10(BR(I))-ALR)*YFACT
      IF(BR(I-1).EQ.999.) THEN
        IP2=3
      ELSE
        IP2=2
      END IF
      IF(YY.GT.YMAX) THEN
        YY=YMAX
        IF(IPEN.EQ.2) THEN
          CALL DPLOT(XX,YY,IP2)
          IPEN=3
        ELSE
          CALL DPLOT(XX,YY,3)
        ENDIF
      ELSE IF(YY.LT.Y) THEN
        YY=Y
        IF(IPEN.EQ.2) THEN
          CALL DPLOT(XX,YY,IP2)
          IPEN=3
        ELSE
          CALL DPLOT(XX,YY,3)
        ENDIF
      ELSE
        CALL DPLOT(XX,YY,IP2)
        IPEN=2
      ENDIF
   20 CONTINUE
      ENDIF
      RETURN
      END
C*************************************************************
C FFT OF SINGLE REAL FUNCTION
C    INPUT DATA HAVE 2N ELEMENTS
C       ISIGN = 1 FOR FOURIER TRANSFORM
C             TRANSFORMD DATA SHOULD BE MULTIPLIED BY 1/N
C             WITH SEQUANCE OF
C             C(0),C(N),C(1),S(1),C(2),S(2),.....C(N-1),S(N-1)
C          WHILE X(J) : J=1,2,....,2N IS EXPRESSED AS
C             X(J)= 0.5*C(0) + SUM(C(K)COS(..)+S(K)SIN(..)) + 0.5*C(N)
```

```
C
C       ISIGN = -1 FOR INVERSE FOURIER TRANSFORM
C
C       REFERENCE: NUMERICAL RECIPES BY W.H. PRESS ET AL., CAMBRIDGE 1986
C            MODIFIED BY Y. OMURA, SEPTEMBER, 1989
C
      SUBROUTINE REALFT(DATA,N2,ISIGN)
      REAL*8 WR,WI,WPR,WPI,WTEMP,THETA
      DIMENSION DATA(N2)
      N=N2/2
      THETA=3.141592653589793D0/DBLE(N)
      C1=0.5
      IF(ISIGN.EQ.1) THEN
        C2=-0.5
        CALL FOUR1(DATA,N2,1)
      ELSE
        C2=0.5
        THETA=-THETA
      ENDIF
      WPR=-2.D0*DSIN(0.5D0*THETA)**2
      WPI=DSIN(THETA)
      WR=1.D0+WPR
      WI=WPI
      N2P3=2*N+3
      DO 11 I1=3,N-1,2
        WRS=SNGL(WR)
        WIS=SNGL(WI)
        H1R=C1*(DATA(I1)+DATA(N2P3-I1-1))
        H1I=C1*(DATA(I1+1)-DATA(N2P3-I1))
        H2R=-C2*(DATA(I1+1)+DATA(N2P3-I1))
        H2I=C2*(DATA(I1)-DATA(N2P3-I1-1))
        DATA(I1)=H1R+WRS*H2R-WIS*H2I
        DATA(I1+1)=H1I+WRS*H2I+WIS*H2R
        DATA(N2P3-I1-1)=H1R-WRS*H2R+WIS*H2I
        DATA(N2P3-I1)=-H1I+WRS*H2I+WIS*H2R
        WTEMP=WR
        WR=WR*WPR-WI*WPI+WR
        WI=WI*WPR+WTEMP*WPI+WI
   11 CONTINUE
      IF(ISIGN.EQ.1) THEN
        H1R=DATA(1)
        DATA(1)=H1R+DATA(2)
```

```
        DATA(2)=H1R-DATA(2)
      ELSE
        H1R=DATA(1)
        DATA(1)=C1*(H1R+DATA(2))
        DATA(2)=C1*(H1R-DATA(2))
        CALL FOUR1(DATA,N2,-1)
      ENDIF
      RETURN
      END
C     ******************************
      SUBROUTINE RKFFT(AR,N,BR,M,INC)
C     ******************************
C     PROGRAMED BY Y. OMURA
C
      DIMENSION AR(N),BR(M)
      NM=N/M
      M2=M/2
      CALL REALFT(AR,N,1)
      RNI=2./FLOAT(N)
      DO 10 I=1,N
   10 AR(I)=AR(I)*RNI
      BR(1)=AR(1)
      BR(2)=AR(2)
      NM2=NM*2
      IF((INC.LT.1).OR.(INC.GT.NM)) INC=NM
      I=1+INC*2
      J=3
      DO 20 L=2,M2
        BR(J)=AR(I)
        BR(J+1)=AR(I+1)
        J=J+2
        I=I+NM2
   20 CONTINUE
      RETURN
      END
C     ************************
      REAL FUNCTION RNDOFF(X,N)
C     ************************
C     PROGRAMED BY Y. OMURA
C
      IS=1
      IF(X.LT.0.) IS=-1
```

```
      XA=ABS(X)
      NA=N
      IF(NA.LT.0) NA=N+1
      FACT=10.**NA
      XA=XA*FACT
      IXA=INT(XA)
      DXA=XA-FLOAT(IXA)
      IF(DXA.GE.0.5) IXA=IXA+1
      ADD=0.001
      IF(IXA.EQ.0) ADD=0.
      RNDOFF=(FLOAT(IXA)+ADD)/FACT*FLOAT(IS)
      RETURN
      END
C     ***************************
      SUBROUTINE SKFFT(AR,N,BR,M)
C     ***************************
C     PROGRAMED BY Y. OMURA
C
      DIMENSION AR(N),BR(M)
      CALL REALFT(AR,N,1)
      RNI=2./FLOAT(N)
      DO 10 I=1,N
   10 AR(I)=AR(I)*RNI
      DO 20 I=1,M
        BR(I)=AR(I)
   20 CONTINUE
      IF(N.GT.M) BR(2)=AR(M+1)
      BR(1) = 0.5*BR(1)
      RETURN
      END
C*********************************************************************
      REAL FUNCTION STRNDM(IY)
      X=0.
      DO 10 K=1,12
        X = X + UNRNDM(IY)
   10 CONTINUE
      STRNDM = X - 6.0
      RETURN
      END
C*********************************************************************
      REAL FUNCTION UNRNDM(IY)
      JK=1048576
```

```
      IY=IY*153+7391
      IY=IY-JK*(IY/JK)
      Y=IY
      X=JK
      UNRNDM=Y/X
      RETURN
      END
C     ****************************************
      SUBROUTINE WKFFT(AR,N1,M1,N,M,WK1,WK2,ICNT)
C     ****************************************
C
C-------------BY Y.OMURA     RASC, KYOTO UNIV.   -----
C--- SUBROUTINE TO FOURIER TRANSFORM IN SPACE AND TIME ----
C---   ICNT=0   FFT IN BOTH X(Z) AND Y(T) COMPORNENTS ----
C---   ICNT=1   FFT IN Y(T) COMPORNENT              ----
C---   ICNT=2    FFT IN X(Z) COMPORNENT      ---
C
      DIMENSION AR(N1,M1)
      DIMENSION WK1(N1),WK2(M1)
      DATA PI/3.1415926/
C
      RNI=2./FLOAT(N)
      RMI=2./FLOAT(M)
      IF(ICNT.EQ.1) GO TO 35
      DO 30 J=1,M
        DO 10 I=1,N
   10   WK1(I)=AR(I,J)
        CALL REALFT(WK1,N,1)
        DO 20 I=1,N
   20   AR(I,J)=WK1(I)*RNI
        AR(1,J)=0.5*AR(1,J)
        AR(2,J)=0.5*AR(2,J)
   30 CONTINUE
   35 CONTINUE
      IF(ICNT.EQ.2) GO TO 45
      DO 40 I=1,N
        DO 50 J=1,M
   50   WK2(J)=AR(I,J)
        CALL REALFT(WK2,M,1)
        DO 60 J=1,M
   60   AR(I,J)=WK2(J)*RMI
   40 CONTINUE
```

```
45 CONTINUE
   DO 91 I=1,N
     AR(I,1)=ABS(AR(I,1))*0.5
     AR(I,2)=ABS(AR(I,2))*0.5
91 CONTINUE
   N2=N/2
   M2=M/2
   DO 64 I=1,2
   J=3
   DO 65 L=2,M2
   AR1=AR(I,J)
   AR2=AR(I,J+1)
   SQ=AR1*AR1+AR2*AR2
   ARA=SQRT(SQ)
   IF(ARA.EQ.0.) ARA=0.0001
   T1=ACOS(AR2/ARA)
   IF(AR1.LT.0.) T1=T1+PI
   AR(I,J)=ARA
   AR(I,J+1)=T1
   J=J+2
65 CONTINUE
64 CONTINUE
   DO 66 J=1,2
   I=3
   DO 67 L=2,N2
   AR1=AR(I,J)
   AR2=AR(I+1,J)
   SQ=AR1*AR1+AR2*AR2
   ARA=SQRT(SQ)
   AR(I,J)=ARA
   AR(I+1,J)=ARA
   I=I+2
67 CONTINUE
66 CONTINUE
   J=3
   DO 70 L=2,M2
   I=3
   DO 80 K=2,N2
     CC=AR(I,J)
     CS=AR(I,J+1)
     SC=AR(I+1,J)
     SS=AR(I+1,J+1)
```

```
          SQ=(CS-SC)**2+(CC+SS)**2
          AR(I,J)=0.5*SQRT(SQ)
          SQ=(CS+SC)**2+(CC-SS)**2
          AR(I+1,J)=0.5*SQRT(SQ)
          AR1=AR(I,J)
          IF(AR1.EQ.0.) AR1=0.0001
          AR2=AR(I+1,J)
          IF(AR2.EQ.0.) AR2=0.0001
          TC1=0.5*(CS-SC)/AR1
          TC2=0.5*(CS+SC)/AR2
          T1=ACOS(TC1)
          TSIGN=CC+SS
          IF(TSIGN.LT.0.) T1=T1+PI
          T2=ACOS(TC2)
          TSIGN=CC-SS
          IF(TSIGN.LT.0.) T2=T2+PI
          AR(I,J+1)=T1
          AR(I+1,J+1)=T2
          I=I+2
   80 CONTINUE
        J=J+2
   70 CONTINUE
        RETURN
        END
C       ************************************************************
        SUBROUTINE WKPLOT(AR,N1,M1,N,M,WK1,WK2,X0,Y0,XL,YL,SL,ST,NQ,
       &                  MQ,ID)
C       ************************************************************
C
C            PROGRAMED BY Y. OMURA
C
        CHARACTER XTITLE*43,YTITLE*43
        DIMENSION AR(N1,M1),WK1(N1),WK2(M1)
C
          KAKUDO=0
        PI=3.1415926
        IC=ID
        IF(WK1(1).EQ.999.) THEN
          IPMAX=1
          EMAX=WK1(2)
        ELSE
          IPMAX=0
```

```
      ENDIF
C
      NP=NQ
      MP=MQ
      N2=N/2
      M2=M/2
      IF((NP.LT.1).OR.(NP.GT.N2)) NP=N2
      IF((MP.LT.1).OR.(MP.GT.M2)) MP=M2
      MP1=MP+1
      WMAX=2.*PI*FLOAT(NP)/SL
      YMAX=2.*PI*FLOAT(MP)/ST
      M21=M2+1
      DX=XL/FLOAT(NP)
      DY=YL/FLOAT(MP)
C
C--- PLOT THE AXIS ----
      WMAX=RNDOFF(WMAX,4)
      YMAX=RNDOFF(YMAX,4)
      SIGN=1.
        LOGPLT=0
C
      NWMAX=3
      NYMAX=3
      IF(ABS(IC).GE.10000) THEN
        IC=IC/ABS(IC)*(ABS(IC)-10000)
        DANG=WK1(3)
        KAKUDO=1
      END IF
      IF(ABS(IC).GE.1000) THEN
        IC=IC/ABS(IC)*(ABS(IC)-1000)
        LOGPLT=1
      ENDIF
      IF(IC.LT.0) SIGN=-1.
      DYD=YL/YMAX*ASCALE(YMAX,-0.4)/8.
      IF((YMAX.LT.10.).AND.(YMAX.GE.2.)) THEN
        DYD=YL/YMAX*ASCALE(YMAX,-2.)/8.
      ENDIF
      IF((YMAX.LT.8.).AND.(YMAX.GE.6.)) THEN
        DYD=YL/YMAX*ASCALE(YMAX,-2.)/6.
      ENDIF
      IF((YMAX.LT.2.).AND.(YMAX.GE.1.0)) THEN
        DYD=YL/YMAX*ASCALE(YMAX,-1.)/4.
```

```
      ENDIF
      IF((YMAX.LT.1.).AND.(YMAX.GE.0.5)) THEN
        DYD=YL/YMAX*ASCALE(YMAX,-5.)/2.
      ENDIF
C
C       IF(WMAX.LT.0.1.OR.YMAX.GT.10.0 ) NWMAX=1001
C       IF(YMAX.LT.1.0.OR.YMAX.GT.10.0 ) NYMAX=1001
      IF(DYD.EQ.0.) DYD=YL/(YMAX*10.)*INT(YMAX*10.)/8.
        XTITLE='K'
        NKT=1
      YTITLE='OMEGA'
      NWT=5
      CALL XAXIS1(X0,Y0,XL,4,2,XL*0.02,XL*0.05,
     &            0.,WMAX*SIGN,NWMAX,XTITLE,NKT)
      CALL XAXIS1(X0,Y0+YL,XL,4,2,-XL*0.02,0.,
     &            0.,WMAX*SIGN,NWMAX,XTITLE,NKT)
      CALL YAXIS2(X0,Y0,YL,DYD,2,XL*0.02,XL*0.05,
     &            0.,YMAX,NYMAX,YTITLE,NWT)
      CALL YAXIS2(X0+XL,Y0,YL,DYD,2,-XL*0.02,0.,
     &            0.,YMAX,NYMAX,YTITLE,NWT)
C
      IF(KAKUDO.EQ.1) THEN
        CALL PRMPLT(X0+XL*0.5,Y0+YL*1.15,XL*0.04,0.,'ANGLE',
     $              5,DANG,-2)
      END IF
C--- GET THE MAXIMUM COMPONENT OF AR ----------
      FMAX=1.E-7
      FMIN=1.E+9
      DO 15 L=1,N
   15 WK1(L)=0.
      I0=3
      IF(IC.EQ.-1) I0=4
      J=1
      DO 14 K=1,MP
      WK1(1)=AR(1,J)
      I=I0
      DO 12 L=2,NP
      WK1(L)=AR(I,J)
      IF(IC.EQ.0) WK1(L)=WK1(L)+AR(I+1,J)
      I=I+2
   12 CONTINUE
      IF(NP.EQ.N2) WK1(N2+1)=AR(2,J)
```

```
      WK1(N2+1)=AR(2,J)
      CALL MAXMIN(WK1,NP,VMIN,VMAX)
      IF(VMIN.LT.FMIN) FMIN=VMIN
      IF(VMAX.GT.FMAX) FMAX=VMAX
      J=J+2
   14 CONTINUE
      I=I0
      DO 99 L=2,NP
      WK1(L)=AR(I,2)
      IF(IC.EQ.0) WK1(L)=WK1(L)+AR(I+1,2)
      I=I+2
   99 CONTINUE
      IF(NP.EQ.N2) WK1(N2+1)=AR(2,2)
      WK1(N2+1)=AR(2,2)
      CALL MAXMIN(WK1,NP,VMIN,VMAX)
      IF(VMIN.LT.FMIN) FMIN=VMIN
      IF(VMAX.GT.FMAX) FMAX=VMAX
      IF(IPMAX.EQ.1) FMAX=EMAX
      CALL PRMPLT(X0+XL*0.5,Y0+YL*1.05,XL*0.04,0.,'MAX',3,FMAX,3)
      FACT=DX/FMAX
      IF(LOGPLT.NE.0) THEN
      IF(FMIN.LE.0) FMIN=FMAX*1.E-2
      AFMAX=ALOG10(FMAX)
      AFMIN=ALOG10(FMIN)
      FACT=DX/(AFMAX-AFMIN)
      ENDIF
      J=1
      DO 22 K=1,M2
        WK2(K)=AR(1,J)
        J=J+2
   22 CONTINUE
      WK2(M21)=AR(1,2)
      PX=X0
      PY=Y0
      DD=WK2(1)*FACT
      IF(LOGPLT.NE.0) DD=(ALOG10(WK2(1))-AFMIN)*FACT
      IF(DD.GT.DX) DD=DX
       IF(DD.LT.0.) DD=0.
      CALL PLOT(PX,PY,3)
      CALL PLOT(PX,PY+YL,2)
      CALL PLOT(PX+DD,PY,3)
      DO 32 K=2,MP1
```

```
        PY=PY+DY
        DD=WK2(K)*FACT
      IF(LOGPLT.NE.0) DD=(ALOG10(WK2(K))-AFMIN)*FACT
      IF(DD.GT.DX) DD=DX
       IF(DD.LT.0.) DD=0.
        CALL PLOT(PX+DD,PY,2)
   32 CONTINUE
      PY=Y0
      DO 42 K=1,MP1
        DD=WK2(K)*FACT
      IF(LOGPLT.NE.0) DD=(ALOG10(WK2(K))-AFMIN)*FACT
      IF(DD.GT.DX) DD=DX
       IF(DD.LT.0.) DD=0.
        CALL PLOT(PX,PY,3)
        CALL PLOT(PX+DD,PY,2)
        PY=PY+DY
   42 CONTINUE
      I=I0
      X=X0+DX
      DO 10 L=2,NP
      J=1
      DO 20 K=1,M2
        WK2(K)=AR(I,J)
      IF(IC.EQ.0) WK2(K)=WK2(K)+AR(I+1,J)
        J=J+2
   20 CONTINUE
      WK2(M21)=AR(I,2)
      PX=X
      PY=Y0
      DD=WK2(1)*FACT
      IF(LOGPLT.NE.0) DD=(ALOG10(WK2(1))-AFMIN)*FACT
      IF(DD.GT.DX) DD=DX
       IF(DD.LT.0.) DD=0.
      CALL PLOT(PX,PY,3)
      CALL PLOT(PX,PY+YL,2)
      CALL PLOT(PX+DD,PY,3)
      DO 30 K=2,MP1
        PY=PY+DY
        DD=WK2(K)*FACT
      IF(LOGPLT.NE.0) DD=(ALOG10(WK2(K))-AFMIN)*FACT
      IF(DD.GT.DX) DD=DX
       IF(DD.LT.0.) DD=0.
```

```
      CALL PLOT(PX+DD,PY,2)
30 CONTINUE
   PY=Y0
   DO 40 K=1,MP1
      DD=WK2(K)*FACT
   IF(LOGPLT.NE.0) DD=(ALOG10(WK2(K))-AFMIN)*FACT
   IF(DD.GT.DX) DD=DX
    IF(DD.LT.0.) DD=0.
      CALL PLOT(PX,PY,3)
      CALL PLOT(PX+DD,PY,2)
      PY=PY+DY
40 CONTINUE
   X=X+DX
   I=I+2
10 CONTINUE
   IF(NP.EQ.N2) I=2
   J=1
   DO 21 K=1,M2
      WK2(K)=AR(I,J)
      J=J+2
21 CONTINUE
   WK2(M21)=AR(I,2)
   PX=X
   PY=Y0
   DD=WK2(1)*FACT
   IF(LOGPLT.NE.0) DD=(ALOG10(WK2(1))-AFMIN)*FACT
   IF(DD.GT.DX) DD=DX
    IF(DD.LT.0.) DD=0.
   CALL PLOT(PX,PY,3)
   CALL PLOT(PX,PY+YL,2)
   CALL PLOT(PX+DD,PY,3)
   DO 31 K=2,MP1
      PY=PY+DY
      DD=WK2(K)*FACT
   IF(LOGPLT.NE.0) DD=(ALOG10(WK2(K))-AFMIN)*FACT
   IF(DD.GT.DX) DD=DX
    IF(DD.LT.0.) DD=0.
      CALL PLOT(PX+DD,PY,2)
31 CONTINUE
   PY=Y0
   DO 41 K=1,MP1
      DD=WK2(K)*FACT
```

```
      IF(LOGPLT.NE.0) DD=(ALOG10(WK2(K))-AFMIN)*FACT
      IF(DD.GT.DX) DD=DX
       IF(DD.LT.0.) DD=0.
        CALL PLOT(PX,PY,3)
        CALL PLOT(PX+DD,PY,2)
        PY=PY+DY
   41 CONTINUE
      WK1(2)=FMAX
      IF(IPMAX.EQ.1) WK1(1)=999.
      WK1(3)=FMIN
      WK1(4)=YMAX
      RETURN
      END
C     ***********************************************
      SUBROUTINE XAXIS1(X,Y,DX,MD1,ND2,DT,H,VMIN,VMAX,
     &                  N,NTEXT,NT)
C     ***********************************************
C     PROGRAMED BY Y. OMURA
C
      DIMENSION VM(50)
      CHARACTER NTEXT*(*)
      ND1=MD1
      IF(MD1.GE.1000) ND1=MD1-1000
      CALL PLOT(X,Y,3)
      CALL PLOT(X+DX,Y,2)
      IF(ND1.EQ.0) GO TO 11
      DDX=DX/FLOAT(ND1)
      XX=X
      SDT=0.6*DT
      Y1=Y
      IF(MD1.GE.1000) Y1=Y-SDT
      Y2=Y+SDT
      ND1M=ND1+1
      DO 10 I=1,ND1M
        CALL PLOT(XX,Y1,3)
        CALL PLOT(XX,Y2,2)
        XX=XX+DDX
   10 CONTINUE
   11 IF(ND2.EQ.0) RETURN
      Y1=Y
      IF(MD1.GE.1000) Y1=Y-DT
      Y2=Y+DT
```

```
      DDX=DX/FLOAT(ND2)
      XX=X
      ND3=ND2+1
      DO 30 I=1,ND3
        CALL PLOT(XX,Y1,3)
        CALL PLOT(XX,Y2,2)
        XX=XX+DDX
   30 CONTINUE
      IF(H.EQ.0.0) RETURN
      NPN=N-1
      IF(N.LE.1) NPN=-1
      ZMAX=ABS(VMAX)
      ZMIN=ABS(VMIN)
      IF(ZMIN.GT.ZMAX) ZMAX=ZMIN
      CALL ETRANS(ZMAX,ZMAX,NPW)
      GFACT=10.**NPW
      IF((NPW.LT.1).OR.(NPW.GE.N)) GO TO 80
      GFACT=1.
      NPN=N-NPW-1
   80 CONTINUE
      IF(N.LE.-1) NPN=-N
      IF(N.LE.0) GFACT=1.
      GMAX=VMAX/GFACT
      GMIN=VMIN/GFACT
      CD=0.0
      DTH=DT*H
      IF(MD1.GE.1000) CD=ABS(DT)
      IF(DTH.LT.0.0)   CD=ABS(DT)
      HA=H
      DSY=1.9*H+CD
      IF(H.GE.0.0) GO TO 40
      HA=-H
      DSY=-0.9*HA-CD
   40 CONTINUE
      IF(N.EQ.999) GO TO 555
      IF(N.EQ.0) GO TO 110
  120 DG=(GMAX-GMIN)/10.**(-NPN)
      DG=ABS(DG)
      IF(DG.GT.2.) GO TO 110
      NPN=NPN+1
      GO TO 120
  110 CONTINUE
```

```
      IF(NPN.EQ.0) NPN=-1
      DVM=(GMAX-GMIN)/FLOAT(ND2)
      VMM=GMIN
      DO 70 I=1,ND3
        VM(I)=VMM
        VMM=VMM+DVM
   70 CONTINUE
      DO 50 J=1,ND3
        Z=VM(J)
        Z=RNDOFF(Z,NPN)
        ZABS=ABS(Z)
        I=0
   15   I=I+1
        ZMAX=10.**I
        IF(ZABS.GE.ZMAX) GO TO 15
        IF(Z.LT.0) I=I+1
        NC=I+NPN+1
        DSX=HA*FLOAT(NC)*0.5
        XX=X-DSX+DDX*(FLOAT(J)-1.)
        YY=Y-DSY
        CALL NUMBER(XX,YY,HA,Z,0.,NPN)
   50 CONTINUE
  555 CONTINUE
      ANT=NT
      LINES=1
      DSY=3.9*H+CD
      IF(H.LT.0.0) DSY=-2.8*HA-CD-(LINES-1)*1.2*HA*12./7.
      XX=X+0.5*DX-0.6*HA*ANT
      YY=Y-DSY
      IF(NT.EQ.0) GO TO 77
      CALL SYMBOL(XX,YY,1.2*HA,NTEXT,0.,NT)
      IF(N.EQ.999) RETURN
   77 XX=X+DX-3.6*HA
      XT=X+0.5*DX+0.6*HA*FLOAT(NT)
      IF(XT.GE.XX) XX=XT+HA
      CALL ENUMBR(XX,YY,1.2*HA,GFACT,0.,-2)
      RETURN
      END
C     ***********************************************
      SUBROUTINE XAXIS2(X,Y,DX,DX1,MD1,DT,H,VMIN,VMAX,
     &                  N,NTEXT,NT)
C     ***********************************************
```

```
C       PROGRAMED BY Y. OMURA
C
        DIMENSION VM(50)
        CHARACTER*1 NTEXT(NT)
        MD0=MD1
        IF(MD1.GE.1000) MD0=MD1-1000
        DX2=DX1*FLOAT(MD0)
        CALL PLOT(X,Y,3)
        CALL PLOT(X+DX,Y,2)
        DDX=DX1
        XX=X
        SDT=0.6*DT
        Y1=Y
        IF(MD1.GE.1000) Y1=Y-SDT
        Y2=Y+SDT
        ND1M=RNDOFF(DX/DX1,3)+1.
        DO 10 I=1,ND1M
          CALL PLOT(XX,Y1,3)
          CALL PLOT(XX,Y2,2)
          XX=XX+DDX
   10 CONTINUE
        Y1=Y
        IF(MD1.GE.1000) Y1=Y-DT
        Y2=Y+DT
        DDX=DX2
        XX=X
        ND3=RNDOFF(DX/DX2,3) + 1.
        ND2=ND3-1
        DO 30 I=1,ND3
          CALL PLOT(XX,Y1,3)
          CALL PLOT(XX,Y2,2)
          XX=XX+DDX
   30 CONTINUE
        IF(H.EQ.0.0) RETURN
        NPN=N-1
        IF(N.LE.1) NPN=-1
        ZMAX=ABS(VMAX)
        ZMIN=ABS(VMIN)
        IF(ZMIN.GT.ZMAX) ZMAX=ZMIN
        CALL ETRANS(ZMAX,ZMAX,NPW)
        GFACT=10.**NPW
        IF((NPW.LT.1).OR.(NPW.GE.N)) GO TO 80
```

```
      GFACT=1.
      NPN=N-NPW-1
   80 CONTINUE
      IF(N.LE.-1) NPN=-N
      IF(N.LE.0) GFACT=1.
      GMAX=VMAX/GFACT
      GMIN=VMIN/GFACT
      CD=0.0
      DTH=DT*H
      IF(MD1.GE.1000) CD=ABS(DT)
      IF(DTH.LT.0.0)    CD=ABS(DT)
      HA=H
      DSY=1.9*H+CD
      IF(H.GE.0.0) GO TO 40
      HA=-H
      DSY=-0.9*HA-CD
   40 CONTINUE
      IF(N.EQ.999) GO TO 555
      IF(N.EQ.0) GO TO 110
  120 DG=(GMAX-GMIN)/10.**(-NPN)
      DG=ABS(DG)
      IF(DG.GT.2.) GO TO 110
      NPN=NPN+1
      GO TO 120
  110 CONTINUE
      IF(NPN.EQ.0) NPN=-1
      DVM=(GMAX-GMIN)/(DX/DX2)
      VMM=GMIN
      DO 70 I=1,ND3
        VM(I)=VMM
        VMM=VMM+DVM
   70   CONTINUE
      DO 50 J=1,ND3
        Z=VM(J)
        Z=RNDOFF(Z,NPN)
        ZABS=ABS(Z)
        I=0
   15    I=I+1
        ZMAX=10.**I
        IF(ZABS.GE.ZMAX) GO TO 15
        IF(Z.LT.0) I=I+1
        NC=I+NPN+1
```

```
        DSX=HA*FLOAT(NC)*0.5
        XX=X-DSX+DDX*(FLOAT(J)-1.)
        YY=Y-DSY
        CALL NUMBER(XX,YY,HA,Z,0.,NPN)
   50 CONTINUE
  555 ANT=FLOAT(NT)
      LINES=1
      DSY=3.9*H+CD
      IF(H.LT.0.0) DSY=-2.8*HA-CD-(LINES-1)*1.2*HA*12./7.
      XX=X+0.5*DX-0.6*HA*ANT
      YY=Y-DSY
      IF(NT.EQ.0) GO TO 77
      CALL SYMBOL(XX,YY,1.2*HA,NTEXT,0.,NT)
      IF(N.EQ.999) RETURN
   77 XX=X+DX-3.6*HA
      XT=X+0.5*DX+0.6*HA*FLOAT(NT)
      IF(XT.GE.XX) XX=XT+HA
      CALL ENUMBR(XX,YY,1.2*HA,GFACT,0.,-2)
      RETURN
      END
C     ***********************************************
      SUBROUTINE YAXIS1(X,Y,DY,MD1,ND2,DT,H,VMIN,VMAX,
     &                  N,NTEXT,NT)
C     ***********************************************
C     PROGRAMED BY Y. OMURA
C
      DIMENSION VM(50)
      CHARACTER NTEXT*(*)
      ND1=MD1
      IF(MD1.GE.1000) ND1=MD1-1000
      CALL PLOT(X,Y,3)
      CALL PLOT(X,Y+DY,2)
      IF(ND1.EQ.0) GO TO 11
      DDY=DY/FLOAT(ND1)
      YY=Y
      SDT=0.6*DT
      X1=X
      IF(MD1.GE.1000) X1=X-SDT
      X2=X+SDT
      ND1M=ND1+1
      DO 10 I=1,ND1M
        CALL PLOT(X1,YY,3)
```

```
      CALL PLOT(X2,YY,2)
      YY=YY+DDY
   10 CONTINUE
   11 IF(ND2.EQ.0) RETURN
      X1=X
      IF(MD1.GE.1000) X1=X-DT
      X2=X+DT
      DDY=DY/FLOAT(ND2)
      YY=Y
      ND3=ND2+1
      DO 30 I=1,ND3
        CALL PLOT(X1,YY,3)
        CALL PLOT(X2,YY,2)
        YY=YY+DDY
   30 CONTINUE
      IF(H.EQ.0.0) RETURN
      ZMIN=ABS(VMIN)
      ZMAX=ABS(VMAX)
      IF(ZMIN.GT.ZMAX) ZMAX=ZMIN
      CALL ETRANS(ZMAX,ZMAX,NPW)
      GFACT=10.**NPW
      NPN=N-1
      IF((NPW.LT.1).OR.(NPW.GE.N)) GO TO 80
      GFACT=1.
      NPN=N-NPW-1
   80 CONTINUE
      IF(N.LE.0) GFACT=1.
      IF(N.EQ.1) NPN=-1
      IF(N.LE.-1) NPN=-N
      GMIN=VMIN/GFACT
      GMAX=VMAX/GFACT
      NCMAX=1
      CD=0.0
      HA=H
      IF(H.GE.0.0) GO TO 40
      HA=-H
   40 DSY=0.5*HA
      IF(N.EQ.0) GO TO 110
      IF(N.EQ.999) GO TO 555
  120 DG=(GMAX-GMIN)/10.**(-NPN)
      DG=ABS(DG)
      IF(DG.GT.2.) GO TO 110
```

```
      NPN=NPN+1
      GO TO 120
  110 CONTINUE
      IF(NPN.EQ.0) NPN=-1
      DVM=(GMAX-GMIN)/FLOAT(ND2)
      VMM=GMIN
      DO 70 I=1,ND3
        VM(I)=VMM
        VMM=VMM+DVM
   70 CONTINUE
      DTH=DT*H
      IF(MD1.GE.1000) CD=ABS(DT)
      IF(DTH.LT.0.0)   CD=ABS(DT)
      NCMAX=0
      DO 50 J=1,ND3
        Z=VM(J)
        Z=RNDOFF(Z,NPN)
        ZABS=ABS(Z)
        I=0
   15   I=I+1
        ZMAX=10.**I
        IF(ZABS.GE.ZMAX) GO TO 15
        IF(Z.LT.0) I=I+1
        NC=I+NPN+1
      IF(NC.GT.NCMAX) NCMAX=NC
        DSX=H*(FLOAT(NC)+0.7)+CD
        IF(H.LT.0.0) DSX=-HA*0.7-CD
        XX=X-DSX
        YY=Y-DSY+DDY*(FLOAT(J)-1.)
        CALL NUMBER(XX,YY,HA,Z,0.,NPN)
   50 CONTINUE
  555 CONTINUE
      ANT=FLOAT(NT)
      LINES=1
      IF(H.GE.0.) DSX=H*(FLOAT(NCMAX)+1.5)+CD+(LINES-1)*1.2*HA*12./7.
      IF(H.LT.0.0) DSX=-HA*(FLOAT(NCMAX)+2.7)-CD
      XX=X-DSX
      YY=Y+0.5*DY-0.6*HA*ANT
      IF(NT.EQ.0) GO TO 77
      CALL SYMBOL(XX,YY,1.2*HA,NTEXT,90.,NT)
   77 XX=X-3.*HA
      IF(N.EQ.999) RETURN
```

```
      IF(H.LT.0.0) XX=X
      YY=Y+DY+HA*1.2
      CALL ENUMBR(XX,YY,1.2*HA,GFACT,0.,-2)
      RETURN
      END
C     ***********************************************
      SUBROUTINE YAXIS2(X,Y,DY,DY1,MD1,DT,H,VMIN,VMAX,
     &                  N,NTEXT,NT)
C     ***********************************************
C     PROGRAMED BY Y. OMURA
C
      DIMENSION VM(50)
      CHARACTER NTEXT*(*)
      MD0=MD1
      IF(MD1.GE.1000) MD0=MD1-1000
      DY2=DY1*FLOAT(MD0)
      CALL PLOT(X,Y,3)
      CALL PLOT(X,Y+DY,2)
      DDY=DY1
      YY=Y
      SDT=0.6*DT
      X1=X
      IF(MD1.GE.1000) X1=X-SDT
      X2=X+SDT
      ND1M=RNDOFF(DY/DY1,3)+1.
      DO 10 I=1,ND1M
        CALL PLOT(X1,YY,3)
        CALL PLOT(X2,YY,2)
        YY=YY+DDY
   10 CONTINUE
      X1=X
      IF(MD1.GE.1000) X1=X-DT
      X2=X+DT
      DDY=DY2
      YY=Y
      ND3=RNDOFF(DY/DY2,3)+1.
      ND2=ND3-1
      DO 30 I=1,ND3
        CALL PLOT(X1,YY,3)
        CALL PLOT(X2,YY,2)
        YY=YY+DDY
   30 CONTINUE
```

```
      IF(H.EQ.0.0) RETURN
      ZMIN=ABS(VMIN)
      ZMAX=ABS(VMAX)
      IF(ZMIN.GT.ZMAX) ZMAX=ZMIN
      CALL ETRANS(ZMAX,ZMAX,NPW)
      GFACT=10.**NPW
      NPN=N-1
      IF((NPW.LT.1).OR.(NPW.GE.N)) GO TO 80
      GFACT=1.
      NPN=N-NPW-1
   80 CONTINUE
      IF(N.LE.0) GFACT=1.
      IF(N.EQ.1) NPN=-1
      IF(N.LE.-1) NPN=-N
      GMIN=VMIN/GFACT
      GMAX=VMAX/GFACT
      NCMAX=1
      CD=0.0
      HA=H
      IF(H.GE.0.0) GO TO 40
      HA=-H
   40 DSY=0.5*HA
      IF(N.EQ.0) GO TO 110
      IF(N.EQ.999) GO TO 555
  120 DG=(GMAX-GMIN)/10.**(-NPN)
      DG=ABS(DG)
      IF(DG.GT.2.) GO TO 110
      NPN=NPN+1
      GO TO 120
  110 CONTINUE
      IF(NPN.EQ.0) NPN=-1
      DVM=(GMAX-GMIN)/(DY/DY2)
      VMM=GMIN
      DO 70 I=1,ND3
        VM(I)=VMM
        VMM=VMM+DVM
   70 CONTINUE
      DTH=DT*H
      IF(MD1.GE.1000) CD=ABS(DT)
      IF(DTH.LT.0.0)   CD=ABS(DT)
      NCMAX=0
      DO 50 J=1,ND3
```

```
        Z=VM(J)
        Z=RNDOFF(Z,NPN)
        ZABS=ABS(Z)
        I=0
   15   I=I+1
        ZMAX=10.**I
        IF(ZABS.GE.ZMAX) GO TO 15
        IF(Z.LT.0) I=I+1
        NC=I+NPN+1
      IF(NC.GT.NCMAX) NCMAX=NC
        DSX=H*(FLOAT(NC)+0.7)+CD
        IF(H.LT.0.0) DSX=-HA*0.7-CD
        XX=X-DSX
        YY=Y-DSY+DDY*(FLOAT(J)-1.)
        CALL NUMBER(XX,YY,HA,Z,0.,NPN)
   50 CONTINUE
  555 CONTINUE
      ANT=FLOAT(NT)
      LINES=1
      IF(H.GE.0.) DSX=H*(FLOAT(NCMAX)+1.5)+CD+(LINES-1)*1.2*HA*12./7.
      IF(H.LT.0.0) DSX=-HA*(FLOAT(NCMAX)+2.7)-CD
      XX=X-DSX
      YY=Y+0.5*DY-0.6*HA*ANT
      IF(NT.EQ.0) GO TO 77
      CALL SYMBOL(XX,YY,1.2*HA,NTEXT,90.,NT)
   77 XX=X-3.*HA
      IF(N.EQ.999) RETURN
      IF(H.LT.0.0) XX=X
      YY=Y+DY+HA*1.2
      CALL ENUMBR(XX,YY,1.2*HA,GFACT,0.,-2)
      RETURN
      END
C
C     ***************
      SUBROUTINE XYROT(X,Y,X0,Y0,ANGLE)
C     ***************
C     PROGRAMED BY Y. OMURA
      DATA ANGL0,COSTH,SINTH /0.,1.,0./
      IF(ANGLE.NE.ANGL0) THEN
        THETA=3.14159265*ANGLE/180.
        COSTH=COS(THETA)
        SINTH=SIN(THETA)
```

```
          ANGL0=ANGLE
        END IF
        DX=X-X0
        DY=Y-Y0
        X=X0+DX*COSTH-DY*SINTH
        Y=Y0+DX*SINTH+DY*COSTH
        RETURN
        END
==================== end of libkempo.f ================================
```