```
!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12
!
! Copyright D.J. Jeffery, 2006jan01.

!  plot.f is for making plots of supernova models.
!
!   References:
!
!     \bibitem[Metcalf et al.(2004)]{metcalf2004} Metcalf, M., Reid, J.,
!         \&~Cohen, M. 2004,
!          Fortran 95/2003 Explained
!          (Oxford:  Oxford University Press) (MRC)
!          % Pretty good reference book.
!
!     \bibitem[Nomoto et al.(1984)]{nomoto1984}
!           Nomoto, K., Thielemann, F.-K., \&~Yokoi, \apj, 286, 644
!           % The original W7 paper.
!          % Here the iron peak yield is .86 solar masses
!
!     \bibitem[Press et al.(1992)]{press1992} Press, W. H., Teukolsky, S. A.,
!          Vetterling, W. T., \& Flannery, B. P. 1992,
!            Numerical Recipes in Fortran
!            (Cambridge:  Cambridge University Press)
!            % Bill should have given me a complimentary copy when
!            % I co-taught with him in 1993!
!
!      \bibitem[Thielemann et al.(1986)]{thielemann1986} Thielemann, F.-K.,
!            Nomoto, K.,
!            \&~Yokoi, A\&A, 158, 17
!             % The final W7 composition paper.
!             % I sum the iron-peak to be .79 solar masses.
!             % This is less than the old W7 which had .86 solar masses
!             % of iron-peak
!
!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12
!
      program plot
      use numerical
      use physical
      use astronomical
      use atomic_mod       ! The module is in /aalib/constant.f.
      use model_mod        ! The module is in /model/readin.f.
      implicit none
!
      integer, parameter :: nfile=10
      integer, parameter :: nvel_label=50
!
! The ``s'' endings indicate single precision.
!
      real (kind=nsngl) :: abundmax
      real (kind=nsngl) :: abundmin
      real (kind=nsngl) :: abunds(ncell,natom)
      character :: afile(nfile)*180
      character :: aplot*180
      character :: atmp*80
      character :: caption(nfile)*80
      character :: file_abund*180='abund.ps'
      character :: file_rho*180='rho.ps'
      integer :: i,j,k,l,m,n
      integer :: iabund=1
      integer :: iatoms(nfile)
      integer :: icells(nfile)
      integer :: idash(nfile)
      integer :: ifile
      integer :: ifile_expon(nfile)
      integer :: ifile_type(nfile)
      integer :: ipost(2)=1
      integer :: iradios(nfile)
      integer :: iradio_isobars=2
```



```
      integer :: itmp
      integer :: ivel_label=0
      integer :: ivel_label_radio=0
      integer :: ivel_labelb(nvel_label)=0
      real (kind=nsngl) :: radios(ncell,nradio_isobar,nradio)
      real (kind=nsngl) :: rhocs(nfile)
      real (kind=nsngl) :: rhomax
      real (kind=nsngl) :: rhomin
      real (kind=nsngl) :: rhos(ncell,nfile)
      real (kind=nsngl) :: sum1,sum2
      character :: title*80
      real (kind=nsngl) :: tmp
      real (kind=nprecision) :: tmpdble
      real (kind=nsngl) :: tmp1,tmp2,tmp3
      real (kind=nprecision) :: vefolds(nfile)
      real (kind=nsngl) :: vel_label(nvel_label)
      real (kind=nsngl) :: vel_label_radio(nvel_label)
      real (kind=nsngl) :: velmax
      real (kind=nsngl) :: velmax_abund
      real (kind=nsngl) :: velmin
      real (kind=nsngl) :: velmin_abund
      real (kind=nsngl) :: vels(ncell,nfile)
      real (kind=nsngl) :: xcorr=1.
      real (kind=nsngl) :: xytable(2)=(/1.e+4,9.e-9/)
      real (kind=nsngl) :: ycorr=1.5
!
      namelist/plotpar/aplot
!
      namelist/figpar/                                                   &
     &  abundmax                                                         &
     & ,abundmin                                                         &
     & ,afile                                                            &
     & ,caption                                                          &
     & ,file_abund                                                       &
     & ,file_rho                                                         &
     & ,iabund                                                           &
     & ,idash                                                            &
     & ,ifile                                                            &
     & ,ifile_expon                                                      &
     & ,ifile_type                                                       &
     & ,ipost                                                            &
     & ,iradio                                                           &
     & ,ivel_center                                                      &
     & ,ivel_kms                                                         &
     & ,ivel_label                                                       &
     & ,ivel_labelb                                                      &
     & ,ivel_label_radio                                                 &
     & ,rhomax                                                           &
     & ,rhomin                                                           &
     & ,title                                                            &
     & ,vel_label                                                        &
     & ,vel_label_radio                                                  &
     & ,velmax                                                           &
     & ,velmax_abund                                                     &
     & ,velmin                                                           &
     & ,velmin_abund                                                     &
     & ,xytable
!
!----------------------------------------------------------------------
      write(*,*)
      write(*,*) 'Before reading the atomic data from /aalib/atomic.'
!----------------------------------------------------------------------
!
      open(unit=1,file='../../../../aalib/atomic.dat',                   &
     &     status='old',action='read')
        read(1,atomic_data)
      close(unit=1)
!
!----------------------------------------------------------------------
```

```
      write(*,*)
      write(*,*) 'Before reading the plot parameters.'
!-----------------------------------------------------------------------
!
      open(unit=1,file='plot.par',status='old',action='read')
         read(1,plotpar)
         write(*,plotpar)
      close(unit=1)
      open(unit=1,file=trim(aplot),status='old',action='read')
         read(1,figpar)
!         write(*,figpar)
      close(unit=1)
!
!-----------------------------------------------------------------------
      write(*,*)
      write(*,*) 'Before reading the data files.'
      write(*,*) 'Note we can plot many density profiles, but only'
      write(*,*) 'one abundance file as specified by iabund.'
!-----------------------------------------------------------------------
!
      do402 : do i=1,ifile
        if(ifile_type(i) .eq. 1) then
            call w7(trim(afile(i)))
          else if(ifile_type(i) .eq. 2) then
            call bravo(trim(afile(i)))
          else if(ifile_type(i) .eq. 3) then
            write(*,*) 'Before subroutine hoeflich'
            call hoeflich(trim(afile(i)))
        end if
        write(*,*) 'After the model read-in.'
        write(*,*) 'i,ifile,iatom,icell,iradio'
        write(*,*) i,ifile,iatom,icell,iradio
        write(*,*) 'rhoc,vefold'
        write(*,*) rhoc,vefold
!
        iatoms(i)=iatom
        icells(i)=icell
        iradios(i)=iradio
        rhos(:icell,i)=real(rho(:icell),nsngl)
        rhocs(i)=real(rhoc,nsngl)
        vels(:icell,i)=real(vel(:icell),nsngl)
        vefolds(i)=real(vefold,nsngl)
!
        rhos(:icell,i)=log10(rhos(:icell,i))
        rhocs(i)=log10(rhocs(i))
!
        write(*,*) 'After logging density.'
        if(i .eq. iabund) then
            abunds(:icells(i),:)=                                       &
     &       log10( real(max(abund(:icell,:),1.d-30),nsngl) )
            radios(:icells(i),:,:)=                                     &
     &       log10( real(max(radio(:icell,:,:),1.d-30),nsngl) )
        end if
      end do do402
!
      rhomax=log10(rhomax)
      rhomin=log10(rhomin)
      xytable(2)=log10(xytable(2))
!
!-----------------------------------------------------------------------
      write(*,*)
      write(*,*) 'Before density plot.'
!-----------------------------------------------------------------------
!
      if(ipost(1) .eq. 1) then
          call sm_device(                                               &
     &         'postencap '//file_rho)        ! Square plot
        else if(ipost(1) .eq. 2) then
          call sm_device(                                               &
```



```
     &         'postportencap '//file_rho) ! Portrait plot
         else
           call sm_device(                                              &
     &         'postlandencap '//file_rho)    ! Landscape plot
       end if
       call sm_graphics
       call sm_defvar('TeX_strings','1')
!
       call sm_ticksize(1.,0.,-1.,10.)
!
       print*,'Before sm_limits.'
       write(*,*) 'velmin,velmax,rhomin,rhomax'
       write(*,*) velmin,velmax,rhomin,rhomax
       call sm_limits(velmin,velmax,rhomin,rhomax)
!       print*,'Before sm_box.'
       call sm_box(1,2,0,0)
!       print*,'Before sm_expand.'
       call sm_expand(1.)
       call sm_xlabel('\rm Velocity (km s^{-1})')
!       print*,'Before sm_label.'
       call sm_ylabel('\rm Density at 1 day past explosion  (g cm^{-3})')
!       call sm_expand(1.0)
!
       print*,'Before sm_grelocate.'
       call sm_grelocate(2000,0)   ! Location in screen coordinates:
!                                  !  they run 0--32787
       call sm_putlabel(9,trim(title)//'\rm  Density Profile')
!
       write(*,*) 'Before connecting the curves.'
       do410 : do i=1,ifile
!           write(*,*) vels(:,i)
!           write(*,*) rhos(:,i)
          call sm_ltype(idash(i))
          call sm_conn(vels(:,i),rhos(:,i),icells(i))
          if(ifile_expon(i) .eq. 1) then    ! Must update for multiple
!                                           !   density profiles.
              itmp=ifile+1
              call sm_ltype(1)
              idash(itmp)=1
              tmp1=rhocs(i)+log10(exp(-velmin/vefolds(i)))
              tmp2=rhocs(i)+log10(exp(-velmax/vefolds(i)))
              write(*,*) 'tmp1,tmp2'
              write(*,*) tmp1,tmp2
              call sm_relocate(velmin,tmp1)
              call sm_draw(velmax,tmp2)
              write(atmp,'(f5.0)') vefolds(i)
!              caption(itmp)=trim(caption(2))//trim(atmp)//' km s^{-1}'
              caption(itmp)='\rm Exponential fit, v_{e}='
  &
     &                     //trim(atmp)//' km s^{-1}'
          end if
       end do do410
       write(*,*) 'After connecting the curves.'
!
       if(maxval(ifile_expon) .eq. 0) then
           itmp=ifile
         else
           itmp=ifile+1   ! At the moment I allow only one exponential fit.
       end if
       call sim_scale(velmin,rhomin,velmax,rhomax)
       call sim_table(xytable,itmp,idash,caption,xcorr,ycorr)
!
       call sm_alpha
       call sm_hardcopy
!
       write(*,*) 'The density plot is completed.'
!
!
       if(iabund .ne. 0) then   ! start iabund if
```

```
!
      abundmax=log10(abundmax)
      abundmin=log10(abundmin)

!
!-----------------------------------------------------------------------
      write(*,*)
      write(*,*) 'Before abundance plot for file ',iabund,afile(iabund)
      write(*,*) 'At the moment I allow only one model abundances to'
      write(*,*) 'plotted at at time.'
!-----------------------------------------------------------------------
!
      if(ipost(2) .eq. 1) then
          call sm_device(                                                &
     &         'postencap '//file_abund)        ! Square plot
        else if(ipost(2) .eq. 2) then
          call sm_device(                                                &
     &         'postportencap '//file_abund)    ! Portrait plot
        else
          call sm_device(                                                &
     &         'postlandencap '//file_abund)    ! Landscape plot
      end if
      call sm_graphics
      call sm_defvar('TeX_strings','1')
!
      call sm_ticksize(1.,0.,-1.,10.)
!
      print*,'Before sm_limits.'
      write(*,*) 'velmin_abund,velmax_abund,abundmin,abundmax'
      write(*,*) velmin_abund,velmax_abund,abundmin,abundmax
      call sm_limits(velmin_abund,velmax_abund,abundmin,abundmax)
!      print*,'Before sm_box.'
      call sm_box(1,2,0,0)
!      print*,'Before sm_expand.'
      call sm_expand(1.)
      call sm_xlabel('\rm Velocity (km s^{-1})')
!      print*,'Before sm_label.'
      call sm_ylabel('\rm Mass Fraction')
!      call sm_expand(1.0)
!
      print*,'Before sm_grelocate.'
      call sm_grelocate(2000,0)   ! Location in screen coordinates:
!                                 !   they run 0--32787
      call sm_putlabel(9,trim(title)//'\rm  Composition Structure')
!
      write(*,*) 'Before connecting the curves.'
!      write(*,*) iatoms(iabund),abunds(:,26)
      call sm_ltype(0)
      do420 : do i=1,iatoms(iabund)
         call sm_conn(vels(:,iabund),abunds(:,i),icells(iabund))
          do422 : do j=1,ivel_label
             if(ivel_labelb(j) .lt. 0) then
                if(ivel_labelb(j) .eq. -i) cycle do422   ! Omit one label.
               else if(ivel_labelb(j) .gt. 0) then
                if(ivel_labelb(j) .ne. i) cycle do422  ! Omit all but one.
             end if
             call interpolation(icells(iabund),                            &
     &         real(vels(:,iabund),nprecision),                            &
     &         real(abunds(:,i),nprecision),                               &
     &         1,1,real(vel_label(j),nprecision),k,tmpdble)
             tmp=real(tmpdble,nsngl)
!              if(i .eq. 8) print*,vel_label(j),vel(k,iabund),abund(k,i)
             if(tmp .gt. abundmax  .or.  tmp .lt. abundmin) cycle do422
             if(vel_label(j) .gt. velmax_abund  .or.                      &
     &          vel_label(j) .lt. velmin_abund       ) cycle do422
             call sm_relocate(vel_label(j),tmp)
             call sm_putlabel(5,trim(aname(i)))
          end do do422
      end do do420
```



```
      call sm_ltype(1)
      do430 : do i=1,iradios(iabund)
        do440 : do j=1,iradio_isobars     ! Only the first two isotopes in
!                                         !   the chain so far.
!                                         !   The 3rd one is stable in the
!                                         !   chains I've got now.
          call sm_conn(vels(:,iabund),radios(:,j,i),icells(iabund))
!
          do442 : do k=1,ivel_label_radio
             call interpolation(icells(iabund),
     &         real(vels(:,iabund),nprecision),                            &
     &         real(radios(:,j,i),nprecision),                             &
     &         1,1,real(vel_label_radio(k),nprecision),m,tmpdble)
             tmp=real(tmpdble,nsngl)
!             if(i .eq. 8) print*,vel_label(k),vel(m,iabund),radio(m,j,i)
             if(tmp .gt. abundmax  .or.  tmp .lt. abundmin) cycle do442
             if(vel_label_radio(k) .gt. velmax_abund  .or.                &
     &          vel_label_radio(k) .lt. velmin_abund      ) cycle do442
             call sm_relocate(vel_label_radio(k),tmp)
             call sm_putlabel(5,trim(aradio(j,i)))
          end do do442
!
        end do do440
      end do do430
      write(*,*) 'After connecting the curves.'
!
!      call sim_scale(velmin_abund,rhomin,velmax_abund,rhomax)
!      call sim_table(xytable,itmp,idash,caption,xcorr,ycorr)
!
      call sm_alpha
      call sm_hardcopy
!
      write(*,*) 'The abundance plot is completed.'
!
      end if                                         ! end iabund if
!
      end program plot
!
!
!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12
!
!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12
!

&param
afile='./bravo/prd5.5_mod.dat'
ifile_type=2    ! 1 for W7;  2 for Bravo models.
ivel_kms=0      ! A PHOENIX model has velocities in cm/s.
ivel_center=0   ! A PHOENIX model has velocities at the edge.
/

!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12

!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12
!
! Copyright D.J. Jeffery, 2006jan01.
!
!  plot.f is for making plots of supernova models.
!
!   References:
!
!     \bibitem[Metcalf et al.(2004)]{metcalf2004} Metcalf, M., Reid, J.,
!         \&~Cohen, M. 2004,
!          Fortran 95/2003 Explained
!          (Oxford:  Oxford University Press) (MRC)
!          % Pretty good reference book.
!
!     \bibitem[Nomoto et al.(1984)]{nomoto1984}
!           Nomoto, K., Thielemann, F.-K., \&~Yokoi, \apj, 286, 644
```

```
!           % The original W7 paper.
!          % Here the iron peak yield is .86 solar masses
!
!     \bibitem[Press et al.(1992)]{press1992} Press, W. H., Teukolsky, S. A.,
!          Vetterling, W. T., \& Flannery, B. P. 1992,
!            Numerical Recipes in Fortran
!            (Cambridge:  Cambridge University Press)
!            % Bill should have given me a complimentary copy when
!            % I co-taught with him in 1993!
!
!      \bibitem[Thielemann et al.(1986)]{thielemann1986} Thielemann, F.-K.,
!            Nomoto, K.,
!            \&~Yokoi, A\&A, 158, 17
!             % The final W7 composition paper.
!             % I sum the iron-peak to be .79 solar masses.
!             % This is less than the old W7 which had .86 solar masses
!             % of iron-peak
!
!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12
!
!
      module phow_mod
      use numerical
      use physical
      use astronomical
      implicit none
!
      integer, parameter :: nel=45
      integer, parameter :: ncolumn=6  ! 6 columns in the composition
!                                      !   output.  Why not 7 like in the
!                                      !   indexing array.  It would make
!                                      !   more sense as 7, but who knows.
!
      integer :: index_pho(nel)=(/                                       &
!
!        H,   He,  Li,  Be,   B,   C,   N,     7
     &   1,    2,   3,   4,   5,   6,   7,                               &
!
!        o,    f,  ne,  na,  mg,  al,  si,    14
     &   8,    9,  10,  11,  12,  13,  14,                               &
!
!        p,    s,  cl,  ar,   k,  ca,  sc,    21
     &  15,   16,  17,  18,  19,  20,  21,                               &
!
!     ti44, ti48,   v,  cr,  mn,fe52,fe54,    28    Fe52 unstable, Fe54 stable.
     &   0,   22,  23,  24,  25,   0,   0,                               &
!
!     fe56, co56,co57,ni56,ni60,  cu,  zn,    35
     &  26,    0,   0,   0,  28,  29,  30,                               &
!
!       ga,   kr,  rb,  sr,   y,  zr,  nb,    42
     &  31,   36,  37,  38,  39,  40,  41,                               &
!
!        ba,  la , cs,                        45
     &   56,  57,  55                                                    &
!
     & /)

      real (kind=nprecision) :: small_pho=1.d-70
!
      end module phow_mod

!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12
!
      program phow
      use numerical
      use physical
      use astronomical
      use atomic_mod       ! The module is in /aalib/constant.f.
```



```
      use model_mod        ! The module is in /model.
      use phow_mod
      implicit none
!
      character :: afile*180
      real (kind=nprecision) :: anorm
      real (kind=nprecision) :: fm(0:natom)=0.d0
      real (kind=nprecision) :: fm_pho(nel)=0.d0
      real (kind=nprecision) :: fm_pho2(nel)
      integer :: i,j,k,l,m,n
      integer :: ifile_type
      integer :: irow
      real (kind=nprecision) :: tmp,tmp2,tmp3,tmp4
!
      namelist/param/afile,ifile_type,ivel_kms,ivel_center
!
      irow=nel/ncolumn
      if(irow*ncolumn .lt. nel) irow=irow+1
!
      open(unit=1,file='/home/jeffery/public_html/aalib/atomic.dat',       &
     &     status='old',action='read')
        read(1,atomic_data)
      close(unit=1)
!
      open(unit=1,file='phow.par',status='old',action='read')
        read(1,param)
      close(unit=1)
!
      open(unit=1,file=afile,status='old',action='read')
        if(ifile_type .eq. 1) then
            call w7(trim(afile))
          else if(ifile_type .eq. 2) then
            call bravo(trim(afile))
            i=1
            write(*,*) 'i,iatom,abund(i,8),awt(8)'
            write(*,*) i,iatom,abund(i,8),awt(8)
        end if
      close(unit=1)
!
      open(unit=1,file=trim(afile)//'.pho',status='unknown',               &
     &     action='write')
!
        write(1,*) icell
        do402 : do i=1,icell
          write(1,*) i,vel(i),xmass_cell_del(i)*xmass_sun
!         write(*,*) i,vel(i),xmass_cell_del(i)*xmass_sun
        end do do402

!-----------------------------------------------------------------------
        write(*,*)
        write(*,*) 'Getting the exponential fitted vefold and rhoc.'
        write(*,*) 'Conversion from mass fraction to number fraction.'
        write(*,*)
        write(*,*) 'n_i=X_i\rho/(A_i*amu)'
        write(*,*)
        write(*,*) 'f_i=(X_i/A_i)/(sum_j X_j/A_j)'
!-----------------------------------------------------------------------

          write(*,*)
!
        do410 : do i=1,icell
!
!            The elements beyond iatom from the Bravo model are initialized to zer
o
!            above.
!
             anorm=1.d0/sum(abund(i,:iatom)/awt(:iatom))
             fm(1:iatom)=( abund(i,:iatom)/awt(:iatom) )*anorm
             write(*,*) 'anorm=',anorm
```

```
!
          do420 : do j=1,nel
            fm_pho(j)=fm(index_pho(j))
          end do do420
!
          tmp=( radio(i,1,1)/awt(28) )*anorm     ! Ni-56   1st chain, 1st radio
nuclide.
          fm_pho(32)=tmp
          fm_pho(33)=fm_pho(33)-tmp
!
          tmp=( radio(i,2,1)/awt(27) )*anorm     ! Co-56   1st chain, 2nd radio
nuclide.
          tmp2=( radio(i,2,2)/awt(27) )*anorm    ! Co-57   2nd chain, 2nd radio
nuclide.
          tmp3=( radio(i,1,3)/awt(27) )*anorm    ! Co-55   3rd chain, 1st radio
nuclide.
          fm_pho(30)=tmp
!          fm_pho(29)=fm_pho(29)+(fm(27)-tmp-tmp2-tmp3)  ! Change stable Co to
Fe-56.
!          fm_pho(29)=fm_pho(29)+(fm(27)-tmp-tmp2)  ! Change stable Co and Co-5
5 to Fe-56.
!
          tmp4=( radio(i,1,2)/awt(28) )*anorm     ! Ni-57   2nd chain, 1st radi
onuclide.
          fm_pho(31)=tmp4+(fm(27)-tmp)            ! Change the Ni-57 and all no
n-Co-56 Co to Co-57.
!
          tmp=( radio(i,1,4)/awt(22) )*anorm     ! Ti-44   4th chain, 1st radio
nuclide.
          fm_pho(22)=tmp
          fm_pho(23)=fm_pho(23)-tmp
!
          fm_pho(:)=max(fm_pho(:),small_pho)
          fm_pho(:)=fm_pho(:)/sum(fm_pho(:))
!
          if(i .eq. icell-1) then   ! This allows the outer abundances to
              fm_pho2(:)=fm_pho(:)  !   to be extrapolated realistically.
            else if(i .eq. icell) then
              fm_pho(:)=fm_pho2(:)
          end if
!
          do430 : do j=1,irow   ! '(i6,6e21.13)' is Eddie's new format preferen
ce.
            write(1,'(i6,6e21.13)') i,(fm_pho(k),
  &
     &                       k=(j-1)*ncolumn+1,min(j*ncolumn,nel))
          end do do430
!
       end do do410
     close(unit=1)
!
     end program phow
!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12
!
! Copyright D.J. Jeffery, 2006jan01.
!
!  This is a set of subroutines for reading in supernova models.
!  They come in all different formats.
!
!   References:
!
!     \bibitem[Metcalf et al.(2004)]{metcalf2004} Metcalf, M., Reid, J.,
!          \&~Cohen, M. 2004,
!          Fortran 95/2003 Explained
!          (Oxford:  Oxford University Press) (MRC)
!          % Pretty good reference book.
!
!     \bibitem[Nomoto et al.(1984)]{nomoto1984}
!           Nomoto, K., Thielemann, F.-K., \&~Yokoi, \apj, 286, 644
```



```
!           % The original W7 paper.
!          % Here the iron peak yield is .86 solar masses
!
!     \bibitem[Press et al.(1992)]{press1992} Press, W. H., Teukolsky, S. A.,
!          Vetterling, W. T., \& Flannery, B. P. 1992,
!            Numerical Recipes in Fortran
!            (Cambridge:  Cambridge University Press)
!            % Bill should have given me a complimentary copy when
!            % I co-taught with him in 1993!
!
!      \bibitem[Thielemann et al.(1986)]{thielemann1986} Thielemann, F.-K.,
!            Nomoto, K.,
!            \&~Yokoi, A\&A, 158, 17
!             % The final W7 composition paper.
!             % I sum the iron-peak to be .79 solar masses.
!             % This is less than the old W7 which had .86 solar masses
!             % of iron-peak
!
!      \bibitem[Yoon \&~Langer(2005)]{yoon2005}
!        Yoon, S.-C., \&~Langer, N. 2005, A\&A, 435, 985
!        % On the evolution of rapidly rotating massive white dwarfs toward
!        % supernovae or collapses.
!        % On p. 977 they give a figure of binding energy for
!        % massive rotating WDs.
!        % They also give fitting formula for those binding energies.
!
!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12
!
      module model_mod
      use numerical
      use physical
      use astronomical
      use atomic_mod       ! The module is in /aalib/constant.f
      implicit none
!
      integer, parameter :: ncell=400
      real (kind=nprecision) :: abund(ncell,natom)=0.d0
      integer :: iatom    ! Model upper limit on atomic number.
      integer :: icell    ! Number of cells in model.
      integer :: iradio
      integer :: ivel_center=0  ! 0 for none (default);  1 for simple averaging;
!                               !   2 for tricky, but it's about the same as 1.
      integer :: ivel_kms=0     ! 0 for no conversion;  1 for conversion to km/s
.
      real (kind=nprecision) :: radio(ncell,nradio_isobar,nradio)=0.d0
      real (kind=nprecision) :: rho(ncell)
      real (kind=nprecision) :: rhoc
      real (kind=nprecision) :: time     ! Time of model in seconds.
      real (kind=nprecision) :: vefold
      real (kind=nprecision) :: vel(ncell)
      real (kind=nprecision) :: xke_erg
      real (kind=nprecision) :: xmass_cell(ncell)
      real (kind=nprecision) :: xmass_cell_del(ncell)
      real (kind=nprecision) :: xmass_gram
!
      end module model_mod
!
!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12
!
!
!
      subroutine model_diagnostic
      use numerical
      use physical
      use astronomical
      use atomic_mod       ! The module is in /aalib/constant.f
      use model_mod
      implicit none
!
```

```
      integer :: i,j,k,l,m,n
      real (kind=nprecision) :: vel_ipe
      real (kind=nprecision) :: xmass_ipe
      real (kind=nprecision) :: xmass_nico
!
!-----------------------------------------------------------------------
      write(*,*)
      write(*,*) 'Getting the exponential fitted vefold and rhoc.'
!-----------------------------------------------------------------------
!
      vefold=sqrt(sixth*xke_erg/xmass_gram)      ! Exponential model fit
!                                                !   for vefold.
      rhoc=xmass_gram/(8.d0*pi*(vefold*daysec)**3)  ! Exponential model
!                                                   !   fit for rhoc.
      write(*,*) 'Fitted vefold and rhoc for 1 day.'
      write(*,*) 'vefold,rhoc,xmass_gram,xke_erg'
      write(*,'(1p,4e25.15)') vefold,rhoc,xmass_gram,xke_erg
!
!-----------------------------------------------------------------------
      write(*,*)
      write(*,*) 'Checking the Ni/Co-56, IPE mass, and v_ipe.'
!-----------------------------------------------------------------------
!
! Sum up the Ni/Co-56 mass.
      xmass_nico=sum( ( radio(:icell,1,1)+radio(:icell,2,1) )           &
     &                *xmass_cell_del(:icell) )
!
! Sum up the IPE mass.
      xmass_ipe=0.d0
      do410 : do i=1,icell
        xmass_ipe=xmass_ipe+sum(abund(i,21:32))*xmass_cell_del(i)
      end do do410
!
! Determine the fiducial IPE mass core.
! Usually the velocities and masses are both at the cell walls I think.
!
      call interpolation(icell,xmass_cell(:icell),                    &
     &                    vel(:icell),1,1,xmass_ipe,k,vel_ipe)
!
      write(*,*) 'Ni/Co56 mass,IPE mass,g-factor,v_ipe,vel(k)'
      write(*,*) xmass_nico,xmass_ipe,xmass_nico/xmass_ipe,           &
     &           vel_ipe,vel(k)
      write(*,*) 'Si,S,IPE,Ni-56 at vel(k)'
      write(*,*) abund(k,14),abund(k,16),sum(abund(k,21:28)),         &
     &           radio(k,1,1),vel(k)
      write(*,*) 'Si,S,IPE at vel(k+1)'
      write(*,*) abund(k+1,14),abund(k+1,16),sum(abund(k+1,21:28)),
     &           radio(k+1,1,1),vel(k+1)
!
      end subroutine model_diagnostic
!
!
!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12
!
!  Subroutine bravo is for reading the model files of Bravo & Garcia-Senz.
!
      subroutine bravo(afile)
      use numerical
      use physical
      use astronomical
      use atomic_mod
      use model_mod
      implicit none
!
      integer, parameter :: nbravo=15
!
      character, intent(in) :: afile*(*)
!
      integer :: i,j,k,l,m,n
```



```
      integer :: ibravo(nbravo)=(/ 2, 6, 8,10,   12,14,16,18,             &
     &                           20,24,25,26,   27,28,30/)
      real (kind=nprecision) :: tmp
      real (kind=nprecision) :: tmp1,tmp2,tmp3
!
      namelist/param/iatom,time
!
      open(unit=1,file='/home/jeffery/public_html/aalib/atomic.dat',       &
     &     status='old',action='read')
        read(1,atomic_data)
      close(unit=1)
!
!-----------------------------------------------------------------------
      write(*,*)
      write(*,*) 'bravo:  Before zeroing and reading in.'
!-----------------------------------------------------------------------
!
      abund(:,:)=0.d0
      radio(:,:,:)=0.d0
      xke_erg=0.d0
      xmass_cell_del(:)=0.d0
!
      open(unit=1,file=afile,status='old',action='read') ! Open bravo data file.
        read(1,param)
        call readcopy(1,0,0,' ',icell)
        write(*,*)
        write(*,"('icell',3x,'vel (cm/s)',4x,'rho (cgs)',                  &
     &            2x,'m_i (m_sun)',5x,'ratio',                             &
     &            7x,'sum',3x,'X Ni-56')")
        do410 : do i=1,icell
          read(1,*) j,tmp,xmass_cell(i),vel(i),                            &
     &              (abund(i,ibravo(j)),j=1,nbravo),                       &
     &              (radio(i,1,j),radio(i,2,j),radio(i,3,j),j=1,2),        &
     &              radio(i,1,4),radio(i,2,4),radio(i,3,4),                &
     &              radio(i,2,3),radio(1,3,3)     ! The 1st nuclide Co-55 is
!                                                 !   not in Bravo.
          abund(i,28)=abund(i,28)+radio(i,1,1)+radio(i,1,2)
          abund(i,27)=abund(i,27)+radio(i,2,1)+radio(i,2,2)
          abund(i,26)=abund(i,26)+radio(i,3,1)+radio(i,3,2)
!
          abund(i,26)=abund(i,26)+radio(i,2,3)
          abund(i,25)=abund(i,25)+radio(i,3,3)
!
          abund(i,22)=abund(i,22)+radio(i,1,4)
          abund(i,21)=abund(i,21)+radio(i,2,4)
          abund(i,20)=abund(i,20)+radio(i,3,4)
!
! Note in radio(i,j,k) that the last index is the decay chain and the 2nd to
! last is the radionuclide in the decay chain.
! Bravo doesn't have Co-55 abundances in his table.
!
          if(i .gt. 1) then
              xmass_cell_del(i)=xmass_cell(i)-xmass_cell(i-1)
              tmp1=vel(i-1)
              tmp2=vel(i-1)**3
            else
              xmass_cell_del(i)=xmass_cell(i)
              tmp1=0.d0
              tmp2=0.d0
          end if
          rho(i)=(xmass_cell_del(i)*xmass_sun)                             &
     &           /(pi4L3*(vel(i)**3-tmp2)*daysec**3)
          xke_erg=xke_erg+.5d0*( .5d0*(vel(i)+tmp1) )**2                   &
     &            *xmass_cell_del(i)*xmass_sun
          tmp3=sum(abund(i,:iatom))
          write(*,910) i,vel(i),rho(i),                                    &
     &     xmass_cell(i),tmp/(vel(i)*time),tmp3,                           &
     &     abund(i,28)
  910     format(i5,1p,3e13.4,0p,3f10.4)
```

```
        end do do410
      close(unit=1)
!
      xmass_gram=xmass_cell(icell)*xmass_sun
!
      call model_diagnostic
      call velocity_centering
      call velocity_cms_to_kms
!
            write(*,*)
            i=1
            write(*,*) 'i,iatom,abund(i,8),awt(8)'
            write(*,*) i,iatom,abund(i,8),awt(8)
!
      end subroutine bravo
!
!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12
!
!  Subroutine hoeflich is for reading the old model files of Peter Hoeflich.
!
      subroutine hoeflich(afile)
      use numerical
      use physical
      use astronomical
      use atomic_mod
      use model_mod
      implicit none
!
      integer, parameter :: nnatom=30
!
      character, intent(in) :: afile*(*)
!
      integer :: i,j,k,l,m,n
      integer :: iiatom(nnatom)
      character :: record*180
      real (kind=nprecision) :: tmp
      real (kind=nprecision) :: tmp1,tmp2,tmp3
      real (kind=nprecision) :: xmass_mult
!
      namelist/param/iatom,iiatom,icell,rhoc,time,xmass_mult
!
      open(unit=1,file='/home/jeffery/public_html/aalib/atomic.dat',       &
     &     status='old',action='read')
        read(1,atomic_data)
      close(unit=1)
!
!-----------------------------------------------------------------------
      write(*,*)
      write(*,*) 'hoeflich:  Before zeroing and reading in.'
!-----------------------------------------------------------------------
!
      abund(:,:)=0.d0
      radio(:,:,:)=0.d0
      xke_erg=0.d0
      xmass_cell_del(:)=0.d0
!
      open(unit=1,file=afile,status='old',action='read') ! Open bravo data file.
        read(1,param)
        call readcopy(1,0,0,' ',k)
        read(1,*)
        read(1,*) record
        write(*,*) trim(record)
        write(*,"('icell',3x,'vel (cm/s)',4x,'rho (cgs)',                  &
     &            2x,'m_i (m_sun)',5x,'ratio',                             &
     &            7x,'sum',3x,'X Ni-56')")
        do410 : do i=1,icell
          read(1,*) j,xmass_cell(i),rho(i),vel(i),tmp2,tmp3,radio(i,1,1)
        end do do410
        do420 : do i=1,11
```



```
          read(1,*)
        end do do420
        do430 : do i=1,icell
           read(1,*) j
           write(*,*) i,j
           read(1,*) (abund(i,iiatom(j)),j=1,7)
           read(1,*) (abund(i,iiatom(j)),j=8,14)
           read(1,*) (abund(i,iiatom(j)),j=15,21)
           abund(i,28)=abund(i,28)+radio(i,1,1)
           abund(i,26)=abund(i,26)-radio(i,1,1)
           write(*,*) i,abund(i,26),abund(i,28),radio(i,1,1)
        end do do430
!
! Note in radio(i,j,k) that the last index is the decay chain, the 2nd to
! last is the radionuclide in the decay chain.
!
        rhoc=rhoc*(time/daysec)**3
        rho(:)=rho(:)*rhoc
        xmass_cell(:)=xmass_cell(:)*xmass_mult
!
        do440 : do i=1,icell
          if(i .gt. 1) then
              xmass_cell_del(i)=xmass_cell(i)-xmass_cell(i-1)
              tmp1=vel(i-1)
              tmp2=vel(i-1)**3
            else
              xmass_cell_del(i)=xmass_cell(i)
              tmp1=0.d0
              tmp2=0.d0
          end if
          tmp=(xmass_cell_del(i)*xmass_sun)                                &
     &           /(pi4L3*(vel(i)**3-tmp2)*daysec**3)
          xke_erg=xke_erg+.5d0*( .5d0*(vel(i)+tmp1) )**2                   &
     &            *xmass_cell_del(i)*xmass_sun
          tmp3=sum(abund(i,:iatom))
          write(*,910) i,vel(i),rho(i),                                    &
     &     xmass_cell(i),tmp/rho(i),tmp3,                                  &
     &     abund(i,28)
  910     format(i5,1p,3e13.4,0p,3f10.4)
        end do do440
      close(unit=1)
!
      xmass_gram=xmass_cell(icell)*xmass_sun
!
      call model_diagnostic
      call velocity_centering
      call velocity_cms_to_kms
!
            i=1
            write(*,*) 'i,iatom,abund(i,8),awt(8)'
            write(*,*) i,iatom,abund(i,8),awt(8)
!
      end subroutine hoeflich
!
!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12
!
!  Subroutine w7 is for reading the w7.dat file.  Not finished.
!
      subroutine w7(afile)
      use numerical
      use physical
      use astronomical
      use atomic_mod
      use model_mod
      implicit none
!
      character, intent(in) :: afile*(*)
      real (kind=nprecision) :: amass(nnuclide)
      real (kind=nprecision) :: frac(nnuclide)
```

```
      integer :: i,j,k,l,m,n
      integer :: inuclide(nnuclide)
      character :: record*80
      real (kind=nprecision) :: rho_cell
      real (kind=nprecision) :: sum1,sum2,sum3
      real (kind=nprecision) :: tmp
      real (kind=nprecision) :: tmp1,tmp2,tmp3
!
      real (kind=nprecision) :: acarbon=.4875
      real (kind=nprecision) :: aoxygen=.4875
      real (kind=nprecision) :: aneon=.025
      integer :: icell_comp
      integer :: iinuclide
      integer :: itmp
      real (kind=nprecision) :: xmass
      namelist/param/iatom,icell,icell_comp,iinuclide,                     &
     &                iradio,time,xmass
!
      character :: anuclide(nnuclide)*4
      real (kind=nprecision) :: radius(ncell)
!
      namelist/runs/anuclide,radius,rho,vel,xmass_cell
!
      open(unit=1,file='/home/jeffery/public_html/aalib/atomic.dat',       &
     &     status='old',action='read')
        read(1,atomic_data)
      close(unit=1)
!
      open(unit=1,file=afile,status='old',action='read') ! Open w7 data file.
      read(1,param)
      read(1,runs)
!
! Dealing with the first four nuclides.
      inuclide(1:4)=1             ! All hydrogen or they will be.
      amass(1:2)=1.d0
      amass(3)=2.d0
      amass(4)=3.d0
!
!-----------------------------------------------------------------------
      write(*,*)
      write(*,*) 'w7:  Before identifiying the nuclides with '//          &
     &          'their atoms.'
!-----------------------------------------------------------------------
!
      write(*,"('nuclide',' Atomic No.',' Atomic mass')")
      do402 : do i=5,nnuclide   ! The first four a N,P,D,T:  dealt with above.
         do404 : do j=1,natom
           if(anuclide(i)(1:2) .eq. aname2(j)) then
               inuclide(i)=j
               read(anuclide(i)(3:4),'(f3.0)') amass(i)
           end if
         end do do404
         write(*,'(i7,i11,f11.0)') i,inuclide(i),amass(i)
      end do do402
!
!-----------------------------------------------------------------------
      write(*,*)
      write(*,*) 'Before converting density and masses.'
!-----------------------------------------------------------------------
!
! Convert to density at day 1.
      rho(:)=10.d0**rho(:)*(time/daysec)**3
! Conversion to solar masses.  Masses must be in log of fractional mass.
      xmass_cell(:icell)=( 10.d0**xmass_cell(:icell) )*xmass
      xmass_gram=0.d0      ! Mass from the mass in the cells.
      xke_erg=0.d0         ! Kinetic energy.
      abund(:,:)=0.d0
      frac(:)=0.d0
      radio(:,:,:)=0.d0
```



```
!
!-----------------------------------------------------------------------
      write(*,*)
      write(*,*) 'Before calculating mass and kinetic energy and'
      write(*,*) 'doing some consistency checks which passed so-so.'
      write(*,*)
      write(*,"(' zone',16x,'rho given',11x,'rho calculated',              &
     &          1x,'rho ratio',1x,'cell mass')")
!
!-----------------------------------------------------------------------
!
      do410 : do i=1,icell
        if(i .eq. 1) then
            tmp=0.d0
            tmp1=0.d0
            tmp2=0.d0
            xmass_cell_del(i)=xmass_cell(i)
          else
            tmp=vel(i-1)**3
            tmp1=xmass_cell(i-1)
            tmp2=vel(i-1)
            xmass_cell_del(i)=xmass_cell(i)-xmass_cell(i-1)
        end if
!
        rho_cell=xmass_cell_del(i)*xmass_sun                               &
     &           /(pi4L3*(vel(i)**3-tmp)*daysec**3)
        write(*,'(i5,1p,2e25.15,0p,2f10.5)') i,rho(i),rho_cell,            &
     &                                  (rho(i)/rho_cell),                 &
     &                                  xmass_cell(i)
        xmass_gram=xmass_gram+rho(i)*(pi4L3*(vel(i)**3-tmp)*daysec**3)
!
        xke_erg=xke_erg+.5d0*( .5d0*(vel(i)+tmp2) )**2
  &
     &             *rho(i)*(pi4L3*(vel(i)**3-tmp)*daysec**3)
!
        if(i .le. icell_comp) then  ! w7 abundances only given up to icell_comp
 110        continue
            read(1,'(a80)') record
!            write(*,*) record
            if(record(2:4) .ne. 'ZO=') go to 110
            do412 : do j=5,iinuclide+1,5  ! Reading the nuclide fractions.
              read(1,'(2x,5(4x,e10.3))')                                   &
     &          (frac(k),k=j-4,min(j,iinuclide)) ! reading the nuclide fractions
            end do do412
            sum3=sum(frac(:)*amass(:))
            frac(:)=frac(:)*amass(:)/sum3
            do414 : do j=1,nnuclide
               abund(i,inuclide(j))=abund(i,inuclide(j))+frac(j)
            end do do414
            radio(i,1,1)=frac(204)     ! Ni-56  parent.
            radio(i,2,1)=frac(195)     ! Co-56  daughter.
          else                       ! Set outer layer abundances.
            abund(i,6)=acarbon
            abund(i,8)=aoxygen
            abund(i,10)=aneon
            radio(i,1,1)=0.d0
            radio(i,2,1)=0.d0
        end if
!
      end do do410
!
      close(unit=1)   ! Close w7 data file.
!
!-----------------------------------------------------------------------
      write(*,*)
      write(*,*) 'Checking the mass consistency and time'
!-----------------------------------------------------------------------
!
      sum1=xmass_gram/xmass_sun
```

```
      tmp=(xmass/sum1)**third*time
      write(*,*) 'xmass,xmass cal.,time,time cal.,kinetic energy'
      write(*,'(2f25.15,1p,3e25.15)') xmass,sum1,time,tmp,xke_erg
!
      xmass_gram=xmass*xmass_sun    ! This is just for w7 accuracy.
!
      call model_diagnostic
      call velocity_centering
      call velocity_cms_to_kms
!
      end subroutine w7
!
!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12
!
! Conversion to velocity centered in mass cell.
!
! The simple average agrees with the more elaborate result to
! better than 1 % almost always for any reasonably well resolved model
! based on one example test.  But it seems that this should be generally
! true.  One can always test if one is unsure.
!
      subroutine velocity_centering
      use numerical
      use physical
      use astronomical
      use atomic_mod
      use model_mod
      implicit none
!
      integer :: i,j,k,l,m,n
      real (kind=nprecision) :: tmp,tmp1
!
      if(ivel_center .eq. 0) return
!
      tmp=0.d0
      do410 : do i=1,icell
        tmp1=vel(i)
        if(ivel_center .eq. 1) then
            vel(i)=.5d0*(tmp+tmp1)
!            print*,'vel(i)'
!            print*,vel(i)
          else
            vel(i)=sqrt(xmass_cell_del(i)*xmass_sun                          &
     &                 /(pi4*rho(i)*daysec**3*(tmp1-tmp)))
!            print*,vel(i)
        end if
        tmp=tmp1
      end do do410
!
      end subroutine velocity_centering

!23456789a123456789b123456789c123456789d123456789e123456789f123456789g12
!
! Conversion of velocity from cm/s to km/s.
!
      subroutine velocity_cms_to_kms
      use numerical
      use physical
      use astronomical
      use atomic_mod
      use model_mod
      implicit none
!
      if(ivel_kms .eq. 0) return
!
      vel(:)=vel(:)*1.d-5
      vefold=vefold*1.d-5
!
      end subroutine velocity_cms_to_kms
```