/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 1

```
0001   !General numerical routines a
0002
0003
0004   !ccccccccccccccccccccccccccc
0005
0006         subroutine splder(y,d
0007         use Precision
0008   !  Splder fits a cubic spline
0009   !  the grid points in dy.  Dy
0010   !  difference formula for dy/
0011         implicit none
0012         integer, intent(in) :
0013         real(dl), intent(in)
0014         real(dl), intent(out)
0015         integer :: n1, i
0016         real(dl), allocatable
0017
0018         allocate(f(n))
0019         n1=n-1
0020   !  Quartic fit to dy/di at bo
0021         f(1)=(-10._dl*y(1)+15
0022         f(n)=(10._dl*y(n)-15.
0023   !  Solve the tridiagonal syst
0024   !  dy(i-1)+4*dy(i)+dy(i+1)=3*
0025   !  with dy(1)=f(1), dy(n)=f(n
0026         do i=2,n1
0027           f(i)=g(i)*(3._dl*(y
0028         end do
0029         dy(n)=f(n)
0030         do i=n1,1,-1
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 1

```
0001      !General numerical routin
0002
0003
0004      !ccccccccccccccccccccccccc
0005
0006      subroutine splder(y,dy,n,
0007      use Precision
0008      !  Splder fits a cubic sp
0009      !  the grid points in dy.
0010      !  difference formula for
0011      implicit none
0012      integer, intent(in) :: n
0013      real(dl), intent(in) :: y
0014      real(dl), intent(out) ::
0015      integer :: n1, i
0016      real(dl), allocatable, di
0017
0018      allocate(f(n))
0019      n1=n-1
0020      !  Quartic fit to dy/di a
0021      f(1)=(-10._dl*y(1)+15._dl
0022      f(n)=(10._dl*y(n)-15._dl*
0023      !  Solve the tridiagonal
0024      !  dy(i-1)+4*dy(i)+dy(i+1
0025      !  with dy(1)=f(1), dy(n)
0026      do i=2,n1
0027        f(i)=g(i)*(3._dl*(y(i
0028      end do
0029      dy(n)=f(n)
0030      do i=n1,1,-1
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 31

```
0031          dy(i)=f(i)-g(i)*dy(
0032        end do
0033        deallocate(f)
0034        end subroutine splder
0035  !ccccccccccccccccccccccccccc
0036        subroutine splini(g,n
0037        use Precision
0038  !  Splini must be called befo
0039        implicit none
0040        integer, intent(in) :
0041        real(dl), intent(out)
0042        integer :: i
0043
0044        g(1)=0._dl
0045        do i=2,n
0046          g(i)=1/(4._dl-g(i-1
0047        end do
0048        end subroutine splini
0049
0050
0051   !ccccccccccccccccccccccccccc
0052        function rombint2(f,a
0053        use precision
0054  !  Rombint returns the integr
0055  !  The method converges provi
0056  !  f must be real(dl) and mus
0057  !  routine.  tol indicates th
0058
0059  ! Modified by AL to specify m
0060  ! (min steps useful to stop w
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 31

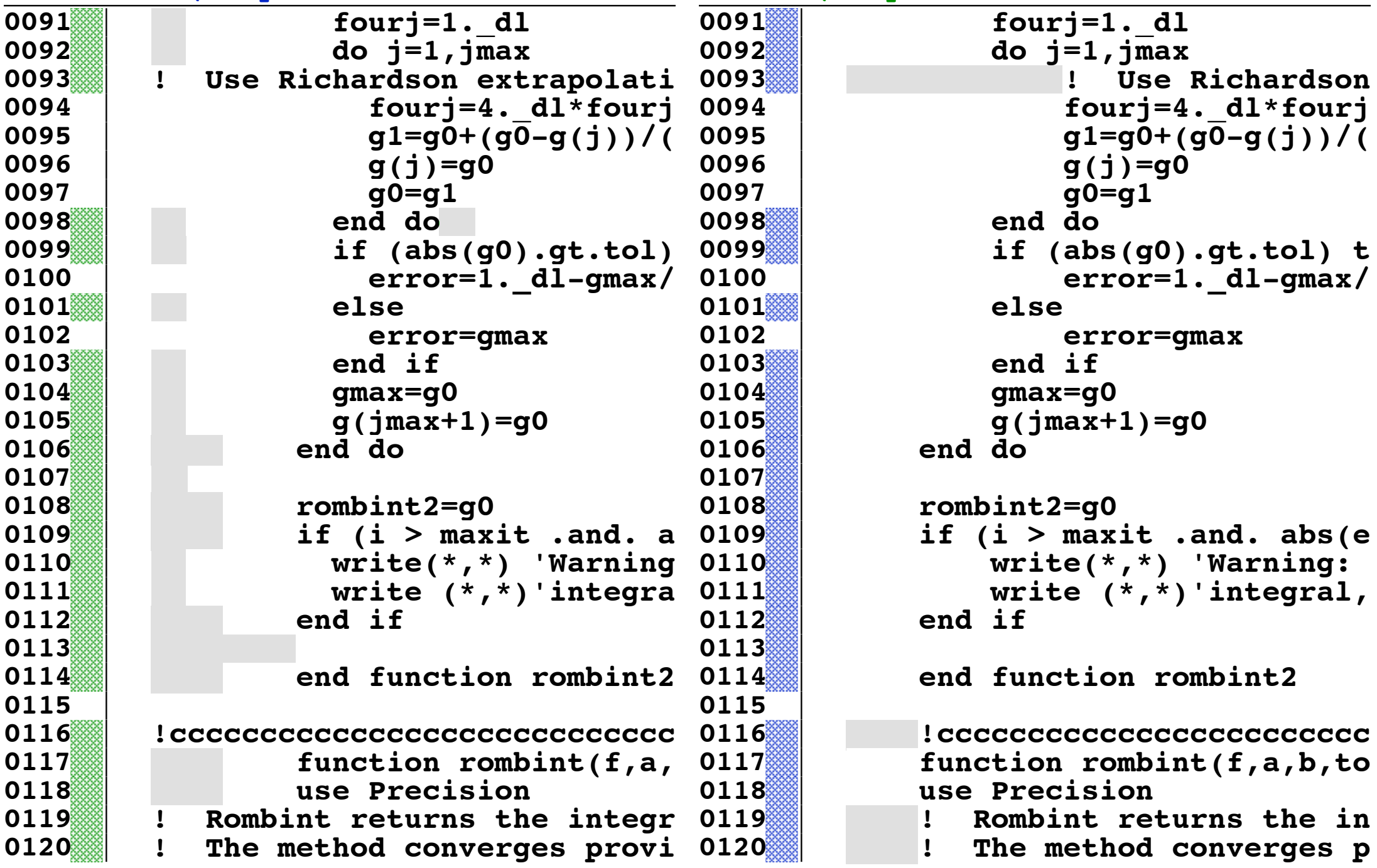
```
0031          dy(i)=f(i)-g(i)*dy(i+
0032        end do
0033        deallocate(f)
0034        end subroutine splder
0035    !ccccccccccccccccccccccccccc
0036        subroutine splini(g,n)
0037        use Precision
0038    !  Splini must be called
0039        implicit none
0040        integer, intent(in) :: n
0041        real(dl), intent(out):: g
0042        integer :: i
0043
0044        g(1)=0._dl
0045        do i=2,n
0046          g(i)=1/(4._dl-g(i-1))
0047        end do
0048        end subroutine splini
0049
0050
0051    !ccccccccccccccccccccccccccc
0052        function rombint2(f,a,b,t
0053        use precision
0054    !  Rombint returns the in
0055    !  The method converges p
0056    !  f must be real(dl) and
0057    !  routine.  tol indicate
0058
0059    ! Modified by AL to speci
0060    ! (min steps useful to st
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 61

```
0061          implicit none
0062          integer, parameter ::
0063          dimension g(MAXJ+1)
0064          real(dl) f
0065          external f
0066          real(dl) :: rombint2
0067          real(dl), intent(in)
0068          integer, intent(in)::
0069
0070          integer :: nint, i, k
0071          real(dl) :: h, gmax,
0072
0073          h=0.5d0*(b-a)
0074          gmax=h*(f(a)+f(b))
0075          g(1)=gmax
0076          nint=1
0077          error=1.0d20
0078          i=0
0079          do
0080            i=i+1
0081            if (i > maxit.or.(i
0082      !  Calculate next trapezoidal
0083            g0=0._dl
0084            do k=1,nint
0085              g0=g0+f(a+(k+k-1)
0086            end do
0087            g0=0.5d0*g(1)+h*g0
0088            h=0.5d0*h
0089            nint=nint+nint
0090            jmax=min(i,MAXJ)
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 61

```
0061          implicit none
0062          integer, parameter :: MAX
0063          dimension g(MAXJ+1)
0064          real(dl) f
0065          external f
0066          real(dl) :: rombint2
0067          real(dl), intent(in) :: a
0068          integer, intent(in):: max
0069
0070          integer :: nint, i, k, jm
0071          real(dl) :: h, gmax, erro
0072
0073          h=0.5d0*(b-a)
0074          gmax=h*(f(a)+f(b))
0075          g(1)=gmax
0076          nint=1
0077          error=1.0d20
0078          i=0
0079          do
0080              i=i+1
0081              if (i > maxit.or.(i >
0082              !  Calculate next tra
0083              g0=0._dl
0084              do k=1,nint
0085                g0=g0+f(a+(k+k-1)
0086              end do
0087              g0=0.5d0*g(1)+h*g0
0088              h=0.5d0*h
0089              nint=nint+nint
0090              jmax=min(i,MAXJ)
```


/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 91
/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 91
0091 fourj=1._dl
0092 do j=1,jmax
0093 ! Use Richardson extrapolati
0094 fourj=4._dl*fourj
0095 g1=g0+(g0-g(j))/(
0096 g(j)=g0
0097 g0=g1
0098 end do
0099 if (abs(g0).gt.tol)
0100 error=1._dl-gmax/
0101 else
0102 error=gmax
0103 end if
0104 gmax=g0
0105 g(jmax+1)=g0
0106 end do
0107
0108 rombint2=g0
0109 if (i > maxit .and. a
0110 write(*,*) 'Warning
0111 write (*,*)'integra
0112 end if
0113
0114 end function rombint2
0115
0116 !ccccccccccccccccccccccccc
0117 function rombint(f,a,
0118 use Precision
0119 ! Rombint returns the integr
0120 ! The method converges provi
0091 fourj=1._dl
0092 do j=1,jmax
0093 ! Use Richardson
0094 fourj=4._dl*fourj
0095 g1=g0+(g0-g(j))/(
0096 g(j)=g0
0097 g0=g1
0098 end do
0099 if (abs(g0).gt.tol) t
0100 error=1._dl-gmax/
0101 else
0102 error=gmax
0103 end if
0104 gmax=g0
0105 g(jmax+1)=g0
0106 end do
0107
0108 rombint2=g0
0109 if (i > maxit .and. abs(e
0110 write(*,*) 'Warning:
0111 write (*,*)'integral,
0112 end if
0113
0114 end function rombint2
0115
0116 !ccccccccccccccccccccccc
0117 function rombint(f,a,b,to
0118 use Precision
0119 ! Rombint returns the in
0120 ! The method converges p

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 121

```
0121    !  f must be real(dl) and mus
0122    !  routine.  tol indicates th
0123    !
0124          implicit none
0125          integer, parameter ::
0126          integer, parameter ::
0127          dimension g(MAXJ+1)
0128          real(dl) f
0129          external f
0130          real(dl) :: rombint
0131          real(dl), intent(in)
0132          integer :: nint, i, k
0133          real(dl) :: h, gmax,
0134    !
0135
0136          h=0.5d0*(b-a)
0137          gmax=h*(f(a)+f(b))
0138          g(1)=gmax
0139          nint=1
0140          error=1.0d20
0141          i=0
0142    10      i=i+1
0143            if (i.gt.MAXITER.or
0144              go to 40
0145    !  Calculate next trapezoidal
0146            g0=0._dl
0147              do 20 k=1,nint
0148              g0=g0+f(a+(k+k-1)
0149    20      continue
0150            g0=0.5d0*g(1)+h*g0
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 121

```
0121          !  f must be real(dl) and
0122          !  routine.  tol indicate
0123          !
0124          implicit none
0125          integer, parameter :: MAX
0126          integer, parameter :: MAX
0127          dimension g(MAXJ+1)
0128          real(dl) f
0129          external f
0130          real(dl) :: rombint
0131          real(dl), intent(in) :: a
0132          integer :: nint, i, k, jm
0133          real(dl) :: h, gmax, erro
0134          !
0135
0136          h=0.5d0*(b-a)
0137          gmax=h*(f(a)+f(b))
0138          g(1)=gmax
0139          nint=1
0140          error=1.0d20
0141          i=0
0142    10    i=i+1
0143          if (i.gt.MAXITER.or.(i.gt
0144              go to 40
0145          !  Calculate next trapezo
0146          g0=0._dl
0147          do 20 k=1,nint
0148            g0=g0+f(a+(k+k-1)*h)
0149    20    continue
0150          g0=0.5d0*g(1)+h*g0
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 151

```
0151          h=0.5d0*h
0152          nint=nint+nint
0153          jmax=min(i,MAXJ)
0154          fourj=1._dl
0155            do 30 j=1,jmax
0156    !  Use Richardson extrapolati
0157            fourj=4._dl*fourj
0158            g1=g0+(g0-g(j))/(
0159            g(j)=g0
0160            g0=g1
0161    30      continue
0162          if (abs(g0).gt.tol)
0163            error=1._dl-gmax/
0164          else
0165            error=gmax
0166          end if
0167          gmax=g0
0168          g(jmax+1)=g0
0169        go to 10
0170    40    rombint=g0
0171          if (i.gt.MAXITER.and.
0172            write(*,*) 'Warning
0173            write (*,*)'integra
0174        end if
0175
0176        end function rombint
0177
0178
0179    !ccccccccccccccccccccccccccc
0180        function rombint_obj(
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 151

```
0151        h=0.5d0*h
0152        nint=nint+nint
0153        jmax=min(i,MAXJ)
0154        fourj=1._dl
0155        do 30 j=1,jmax
0156          !  Use Richardson ext
0157          fourj=4._dl*fourj
0158          g1=g0+(g0-g(j))/(four
0159          g(j)=g0
0160          g0=g1
0161    30  continue
0162        if (abs(g0).gt.tol) then
0163          error=1._dl-gmax/g0
0164        else
0165          error=gmax
0166        end if
0167        gmax=g0
0168        g(jmax+1)=g0
0169        go to 10
0170    40  rombint=g0
0171        if (i.gt.MAXITER.and.abs(
0172          write(*,*) 'Warning:
0173          write (*,*)'integral,
0174        end if
0175
0176        end function rombint
0177
0178
0179      !ccccccccccccccccccccccccc
0180        function rombint_obj(obj,
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 181

```
0181        use Precision
0182    !  Rombint returns the integr
0183    !  The method converges provi
0184    !  f must be real(dl) and mus
0185    !  routine.  tol indicates th
0186    !
0187        implicit none
0188        integer, intent(in),
0189        integer :: MAXITER=20
0190        integer, parameter ::
0191        dimension g(MAXJ+1)
0192        real obj !dummy
0193        real(dl) f
0194        external f
0195        real(dl) :: rombint_o
0196        real(dl), intent(in)
0197        integer :: nint, i, k
0198        real(dl) :: h, gmax,
0199    !
0200
0201        if (present(maxit)) t
0202            MaxIter = maxit
0203        end if
0204        h=0.5d0*(b-a)
0205        gmax=h*(f(obj,a)+f(ob
0206        g(1)=gmax
0207        nint=1
0208        error=1.0d20
0209        i=0
0210    10    i=i+1
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 181

```
0181        use Precision
0182    !  Rombint returns the in
0183    !  The method converges p
0184    !  f must be real(dl) and
0185    !  routine.  tol indicate
0186    !
0187        implicit none
0188        integer, intent(in), opti
0189        integer :: MAXITER=20
0190        integer, parameter :: MAX
0191        dimension g(MAXJ+1)
0192        real obj !dummy
0193        real(dl) f
0194        external f
0195        real(dl) :: rombint_obj
0196        real(dl), intent(in) :: a
0197        integer :: nint, i, k, jm
0198        real(dl) :: h, gmax, erro
0199    !
0200
0201        if (present(maxit)) then
0202            MaxIter = maxit
0203        end if
0204        h=0.5d0*(b-a)
0205        gmax=h*(f(obj,a)+f(obj,b)
0206        g(1)=gmax
0207        nint=1
0208        error=1.0d20
0209        i=0
0210    10  i=i+1
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 211

```
0211        if (i.gt.MAXITER.or
0212          go to 40
0213    !  Calculate next trapezoidal
0214        g0=0._dl
0215          do 20 k=1,nint
0216          g0=g0+f(obj,a+(k+
0217    20    continue
0218        g0=0.5d0*g(1)+h*g0
0219        h=0.5d0*h
0220        nint=nint+nint
0221        jmax=min(i,MAXJ)
0222        fourj=1._dl
0223          do 30 j=1,jmax
0224    !  Use Richardson extrapolati
0225          fourj=4._dl*fourj
0226          g1=g0+(g0-g(j))/(
0227          g(j)=g0
0228          g0=g1
0229    30    continue
0230        if (abs(g0).gt.tol)
0231          error=1._dl-gmax/
0232        else
0233          error=gmax
0234        end if
0235        gmax=g0
0236        g(jmax+1)=g0
0237      go to 10
0238    40  rombint_obj=g0
0239        if (i.gt.MAXITER.and.
0240          write(*,*) 'Warning
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 211

```
0211        if (i.gt.MAXITER.or.(i.gt
0212            go to 40
0213    !  Calculate next trapezo
0214        g0=0._dl
0215        do 20 k=1,nint
0216            g0=g0+f(obj,a+(k+k-1)
0217    20  continue
0218        g0=0.5d0*g(1)+h*g0
0219        h=0.5d0*h
0220        nint=nint+nint
0221        jmax=min(i,MAXJ)
0222        fourj=1._dl
0223        do 30 j=1,jmax
0224          !  Use Richardson ext
0225            fourj=4._dl*fourj
0226            g1=g0+(g0-g(j))/(four
0227            g(j)=g0
0228            g0=g1
0229    30  continue
0230        if (abs(g0).gt.tol) then
0231            error=1._dl-gmax/g0
0232        else
0233            error=gmax
0234        end if
0235        gmax=g0
0236        g(jmax+1)=g0
0237        go to 10
0238    40  rombint_obj=g0
0239        if (i.gt.MAXITER.and.abs(
0240            write(*,*) 'Warning:
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 241

```
0241          write (*,*)'integra
0242        end if
0243
0244        end function rombint_
0245
0246
0247    !ccccccccccccccccccccccccccc
0248    ! calculates array of second
0249    ! interpolation. y2 is array
0250    ! derivatives at end points.
0251
0252    !Thanks Martin Reinecke
0253        subroutine spline(x,y,n,d
0254          use Precision
0255          integer, intent(in) ::
0256          real(dl), intent(in) ::
0257          real(dl), intent(out) :
0258          integer i
0259          real(dl) xp,qn,sig,un,x
0260
0261          d1r= (y(2)-y(1))/(x(2)-
0262          if (d11>.99e30_dl) then
0263            d2(1)=0._dl
0264            u(1)=0._dl
0265          else
0266            d2(1)=-0.5_dl
0267            u(1)=(3._dl/(x(2)-x(1
0268          endif
0269
0270          do i=2,n-1
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 241

```
0241          write (*,*)'integral,
0242        end if
0243
0244        end function rombint_obj
0245
0246
0247    !ccccccccccccccccccccccccccc
0248    ! calculates array of sec
0249    ! interpolation. y2 is ar
0250    ! derivatives at end poin
0251
0252    !Thanks Martin Reinecke
0253        subroutine spline(x,y,n,d
0254        use Precision
0255        integer, intent(in) :: n
0256        real(dl), intent(in) :: x
0257        real(dl), intent(out) ::
0258        integer i
0259        real(dl) xp,qn,sig,un,xxd
0260
0261        d1r= (y(2)-y(1))/(x(2)-x(
0262        if (d11>.99e30_dl) then
0263          d2(1)=0._dl
0264          u(1)=0._dl
0265        else
0266          d2(1)=-0.5_dl
0267          u(1)=(3._dl/(x(2)-x(1
0268        endif
0269
0270        do i=2,n-1
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 271

```
0271              d1l=d1r
0272              d1r=(y(i+1)-y(i))/(x(
0273              xxdiv=1._dl/(x(i+1)-x
0274              sig=(x(i)-x(i-1))*xxd
0275              xp=1._dl/(sig*d2(i-1)
0276
0277              d2(i)=(sig-1._dl)*xp
0278
0279              u(i)=(6._dl*(d1r-d1l)
0280           end do
0281           d1l=d1r
0282
0283           if (d1n>.99e30_dl) then
0284              qn=0._dl
0285              un=0._dl
0286           else
0287              qn=0.5_dl
0288              un=(3._dl/(x(n)-x(n-1
0289           endif
0290
0291           d2(n)=(un-qn*u(n-1))/(q
0292           do i=n-1,1,-1
0293              d2(i)=d2(i)*d2(i+1)+u
0294           end do
0295        end subroutine spline
0296
0297         SUBROUTINE spline_deriv(
0298         !Get derivative y1 given
0299          use Precision
0300          implicit none
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 271

```
0271              d1l=d1r
0272              d1r=(y(i+1)-y(i))/(x(
0273              xxdiv=1._dl/(x(i+1)-x
0274              sig=(x(i)-x(i-1))*xxd
0275              xp=1._dl/(sig*d2(i-1)
0276
0277              d2(i)=(sig-1._dl)*xp
0278
0279              u(i)=(6._dl*(d1r-d1l)
0280           end do
0281           d1l=d1r
0282
0283           if (d1n>.99e30_dl) then
0284              qn=0._dl
0285              un=0._dl
0286           else
0287              qn=0.5_dl
0288              un=(3._dl/(x(n)-x(n-1
0289           endif
0290
0291           d2(n)=(un-qn*u(n-1))/(qn*
0292           do i=n-1,1,-1
0293              d2(i)=d2(i)*d2(i+1)+u
0294           end do
0295           end subroutine spline
0296
0297           SUBROUTINE spline_deriv(x
0298           !Get derivative y1 given
0299           use Precision
0300           implicit none
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 301

```
0301        INTEGER, intent(in) ::
0302        real(dl), intent(in) ::
0303        real(dl), intent(out) :
0304        INTEGER i
0305        real(dl) dx
0306
0307        do i=1, n-1
0308
0309           dx = (x(i+1) - x(i))
0310           y1(i) = (y(i+1) - y(
0311        end do
0312         dx = x(n) - x(n-1)
0313         y1(n) = (y(n) - y(n-1)
0314
0315        END SUBROUTINE spline_d
0316
0317        subroutine spline_integ
0318         !Cumulative integral o
0319         use Precision
0320         integer, intent(in) ::
0321         real(dl), intent(in) :
0322         real(dl), intent(out)
0323         real(dl) dx
0324         integer i
0325
0326         yint(1) = 0
0327         do i=2, n
0328
0329           dx = (x(i) - x(i-1))
0330           yint(i) = yint(i-1)
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 301

```
0301        INTEGER, intent(in) :: n
0302        real(dl), intent(in) :: x
0303        real(dl), intent(out) ::
0304        INTEGER i
0305        real(dl) dx
0306
0307        do i=1, n-1
0308
0309           dx = (x(i+1) - x(i))
0310           y1(i) = (y(i+1) - y(i
0311        end do
0312        dx = x(n) - x(n-1)
0313        y1(n) = (y(n) - y(n-1))/d
0314
0315        END SUBROUTINE spline_der
0316
0317        subroutine spline_integra
0318        !Cumulative integral of c
0319        use Precision
0320        integer, intent(in) :: n
0321        real(dl), intent(in) :: x
0322        real(dl), intent(out) ::
0323        real(dl) dx
0324        integer i
0325
0326        yint(1) = 0
0327        do i=2, n
0328
0329           dx = (x(i) - x(i-1))
0330           yint(i) = yint(i-1) +
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 331

```
0331
0332          end do
0333
0334        end subroutine spline_i
0335
0336
0337
0338    !ccccccccccccccccccccccccccc
0339    !  this is not the splint giv
0340
0341
0342          subroutine splint(y,z
0343          use Precision
0344    !  Splint integrates a cubic
0345    !  z = integral from 1 to n o
0346    !  to y(i).
0347    !
0348          implicit none
0349          integer, intent(in) :
0350          real(dl), intent(in)
0351          real(dl), intent(out)
0352
0353          integer :: n1
0354          real(dl) :: dy1, dyn
0355    !
0356          n1=n-1
0357    !  Cubic fit to dy/di at boun
0358    !        dy1=(-11._dl*y(1)+18.
0359          dy1=0._dl
0360          dyn=(11._dl*y(n)-18._
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 331

```
0331
0332          end do
0333
0334        end subroutine spline_int
0335
0336
0337
0338        !ccccccccccccccccccccccccc
0339        !  this is not the splint
0340
0341
0342          subroutine splint(y,z,n)
0343          use Precision
0344        !  Splint integrates a cu
0345        !  z = integral from 1 to
0346        !  to y(i).
0347        !
0348          implicit none
0349          integer, intent(in) :: n
0350          real(dl), intent(in) :: y
0351          real(dl), intent(out) ::
0352
0353          integer :: n1
0354          real(dl) :: dy1, dyn
0355        !
0356          n1=n-1
0357        !  Cubic fit to dy/di at
0358        !        dy1=(-11._dl*y(1)
0359          dy1=0._dl
0360          dyn=(11._dl*y(n)-18._dl*y
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 361

```
0361    !
0362          z=0.5d0*(y(1)+y(n))+(
0363          z= z + sum(y(2:n1))
0364          end subroutine splint
0365
0366
0367    !This version is modified to
0368    !Fortunately Fortran doesn't
0369    !passed object parameter (EV)
0370          subroutine dverk (EV,n,
0371          use Precision
0372          use AMLUtils
0373          integer n, ind, nw, k
0374          real(dl) x, y(n), xend,
0375          real EV !It isn't, but
0376    !     class(*) EV !seems to b
0377    !
0378    !*****************************
0379    !
0380    ! note added 11/14/85.
0381    !
0382    ! if you discover any errors
0383    !
0384    !        kenneth r. jackson
0385    !        department of comput
0386    !        university of toront
0387    !        toronto, ontario,
0388    !        canada   m5s 1a4
0389    !
0390    !        phone: 416-978-7075
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 361

```
0361    !
0362          z=0.5d0*(y(1)+y(n))+(dy1-
0363          z= z + sum(y(2:n1))
0364          end subroutine splint
0365
0366
0367    !This version is modified
0368    !Fortunately Fortran does
0369    !passed object parameter
0370          subroutine dverk (EV,n, f
0371          use Precision
0372          use AMLUtils
0373          integer n, ind, nw, k
0374          real(dl) x, y(n), xend, t
0375          real EV !It isn't, but as
0376    !     class(*) EV !seems
0377    !
0378    !*************************
0379    !
0380    ! note added 11/14/85.
0381    !
0382    ! if you discover any err
0383    !
0384    !        kenneth r. jacks
0385    !        department of co
0386    !        university of to
0387    !        toronto, ontario
0388    !        canada   m5s 1a4
0389    !
0390    !        phone: 416-978-7
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 391

```
0391    !
0392    !         electronic mail:
0393    !         uucp:   {cornell,dec
0394    !         csnet:  krj@toronto
0395    !         arpa:   krj.toronto@
0396    !         bitnet: krj%toronto@
0397    !
0398    ! dverk is written in fortran
0399    !
0400    ! the constants dwarf and rre
0401    ! set for a  vax  in  double
0402    ! described below, if this pr
0403    !
0404    ! the c array is declared in
0405    ! although  more  elements  a
0406    ! causes some compilers to is
0407    ! no  error  provided  c is d
0408    ! program, as described below
0409    !
0410    ! the following external stat
0411    ! warning  message  from  the
0412    ! comments and code follow it
0413    !
0414    !******************************
0415    !
0416          external fcn
0417    !
0418    !******************************
0419    !
0420    !     purpose - this is a run
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 391

```
0391    !
0392    !         electronic mail:
0393    !         uucp:   {cornell
0394    !         csnet:  krj@toro
0395    !         arpa:   krj.toro
0396    !         bitnet: krj%toro
0397    !
0398    ! dverk is written in for
0399    !
0400    ! the constants dwarf and
0401    ! set for a  vax  in  dou
0402    ! described below, if thi
0403    !
0404    ! the c array is declared
0405    ! although  more  element
0406    ! causes some compilers t
0407    ! no  error  provided  c
0408    ! program, as described b
0409    !
0410    ! the following external
0411    ! warning  message  from
0412    ! comments and code follo
0413    !
0414    !**************************
0415    !
0416      external fcn
0417    !
0418    !**************************
0419    !
0420    !     purpose - this is a
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 421

```
0421    !  fifth and sixth order pair
0422    !  the solution of  a  system
0423    !  equations  with  initial  c
0424    !  error proportional to  a  t
0425    !  proportionality  depends  o
0426    !  as well as the differential
0427    !
0428    !     various options are ava
0429    !  kinds  of  error control, r
0430    !  which permit the user to ex
0431    !  perhaps make modifications)
0432    !
0433    !     the program is efficien
0434    !  variable-order-adams  metho
0435    !  function evaluations are ve
0436    !  more suitable if one wanted
0437    !  solution values by interpol
0438    !  with graphical output.
0439    !
0440    !
0441    !
0442    !*****************************
0443    !
0444    !     use - the user must spe
0445    !
0446    !     n  number of equations
0447    !
0448    !   fcn  name of subroutine f
0449    !           itself must also
0450    !           the following for
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 421

```
0421    !  fifth and sixth order p
0422    !  the solution of  a  sys
0423    !  equations  with  initia
0424    !  error proportional to
0425    !  proportionality  depend
0426    !  as well as the differen
0427    !
0428    !     various options are
0429    !  kinds  of  error contro
0430    !  which permit the user t
0431    !  perhaps make modificati
0432    !
0433    !     the program is effi
0434    !  variable-order-adams  m
0435    !  function evaluations ar
0436    !  more suitable if one wa
0437    !  solution values by inte
0438    !  with graphical output.
0439    !
0440    !
0441    !
0442    !*************************
0443    !
0444    !     use - the user must
0445    !
0446    !     n  number of equati
0447    !
0448    !   fcn  name of subrouti
0449    !           itself must a
0450    !           the following
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 451

```
0451    !              subroutine fcn
0452    !              integer n
0453    !              real(dl) x, y(
0454    !                       *** et
0455    !           and it should eva
0456    !
0457    !     x  independent variable
0458    !
0459    !     y  dependent variable -
0460    !           ..., y(n) supplie
0461    !
0462    !  xend  value of x to which
0463    !           be less than the
0464    !
0465    !   tol  tolerance - the subr
0466    !           local  error  in
0467    !           proportional to t
0468    !           damping  of  erro
0469    !           the global error
0470    !           control   can   b
0471    !           calculated value
0472    !           to  the  problem
0473    !           is proportional t
0474    !           weights  that  de
0475    !           by the user.  the
0476    !           1/max(1,abs(y(k))
0477    !           absolute and rela
0478    !
0479    !   ind  indicator - on initi
0480    !           1  or  2. if the
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 451

```
0451    !              subroutine
0452    !              integer n
0453    !              real(dl) x
0454    !                       **
0455    !           and it should
0456    !
0457    !     x  independent vari
0458    !
0459    !     y  dependent variab
0460    !           ..., y(n) sup
0461    !
0462    !  xend  value of x to wh
0463    !           be less than
0464    !
0465    !   tol  tolerance - the
0466    !           local  error
0467    !           proportional
0468    !           damping  of
0469    !           the global er
0470    !           control   can
0471    !           calculated va
0472    !           to  the  prob
0473    !           is proportion
0474    !           weights  that
0475    !           by the user.
0476    !           1/max(1,abs(y
0477    !           absolute and
0478    !
0479    !   ind  indicator - on i
0480    !           1  or  2. if
```

```
0481    !              should set ind to
0482    !              then  is  to  dec
0483    !              may also  select
0484    !              setting ind = 2 a
0485    !              c as described in
0486    !              the  subroutine
0487    !              any event, the su
0488    !                 3 after a norm
0489    !                 4, 5, or 6 aft
0490    !                 -1, -2, or -3
0491    !
0492    !       c  communications vecto
0493    !              equal to 24, unle
0494    !              case the dimensio
0495    !
0496    !      nw  first dimension of w
0497    !              equal to n
0498    !
0499    !       w  workspace matrix - f
0500    !              be greater than o
0501    !
0502    !       the subroutine  will  n
0503    ! replaced the initial values
0504    ! of xend and an approximatio
0505    ! called  repeatedly  with ne
0506    ! any other argument.  howeve
0507    ! described below, may also b
0508    !
0509    !       three error returns are
0510    ! will be the most recently a
```



```
0481    !              should set in
0482    !              then  is  to
0483    !              may also  sel
0484    !              setting ind =
0485    !              c as describe
0486    !              the  subrouti
0487    !              any event, th
0488    !                 3 after a
0489    !                 4, 5, or 6
0490    !                 -1, -2, or
0491    !
0492    !       c  communications v
0493    !              equal to 24,
0494    !              case the dime
0495    !
0496    !      nw  first dimension
0497    !              equal to n
0498    !
0499    !       w  workspace matrix
0500    !              be greater th
0501    !
0502    !       the subroutine  wil
0503    ! replaced the initial va
0504    ! of xend and an approxim
0505    ! called  repeatedly  wit
0506    ! any other argument.  ho
0507    ! described below, may al
0508    !
0509    !       three error returns
0510    ! will be the most recent
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 511

```
0511    !     with ind = -3 the subro
0512    !        requirement  with a
0513    !        equal to hmin, which
0514    !     with ind = -2 the value
0515    !        probably  means  tha
0516    !        calculation of hmin)
0517    !     with ind = -1 the allow
0518    !        been  exceeded,  but
0519    !        described in the nex
0520    !
0521    !     there are several circu
0522    ! to  be  terminated,  along
0523    ! the user determine the caus
0524    ! involve  entry with illegal
0525    ! such as attempting a normal
0526    ! value of xend, or atteмptin
0527    !
0528    !*****************************
0529    !
0530    !     options - if the subrou
0531    ! components of the communica
0532    ! the subroutine uses only de
0533    ! subroutine  is  entered  wi
0534    ! these 9 components - normal
0535    ! and  then  make  non-zero
0536    ! options he wishes to select
0537    ! re-entry  to  the  subrouti
0538    ! options, or tol, in the cou
0539    ! about  how  such changes af
0540    ! restart with ind = 1 or 2.
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 511

```
0511    !     with ind = -3 the s
0512    !        requirement  wit
0513    !        equal to hmin, w
0514    !     with ind = -2 the v
0515    !        probably  means
0516    !        calculation of h
0517    !     with ind = -1 the a
0518    !        been  exceeded,
0519    !        described in the
0520    !
0521    !     there are several c
0522    ! to  be  terminated,  al
0523    ! the user determine the
0524    ! involve  entry with ill
0525    ! such as attempting a no
0526    ! value of xend, or attem
0527    !
0528    !*************************
0529    !
0530    !     options - if the su
0531    ! components of the commu
0532    ! the subroutine uses onl
0533    ! subroutine  is  entered
0534    ! these 9 components - no
0535    ! and  then  make  non-ze
0536    ! options he wishes to se
0537    ! re-entry  to  the  subr
0538    ! options, or tol, in the
0539    ! about  how  such change
0540    ! restart with ind = 1 or
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 541

```
0541     ! program  -  the information
0542     ! normally be changed by him.
0543     !
0544     !  c(1)  error control indica
0545     !           max  norm  of  th
0546     !           weights being det
0547     !              if c(1)=1 the
0548     !              if c(1)=2 the
0549     !                 control)
0550     !              if c(1)=3 the
0551     !                 (relative
0552     !                 than the fl
0553     !              if c(1)=4 the
0554     !                 (here indiv
0555     !              if c(1)=5 the
0556     !              for all other
0557     !                 default  va
0558     !                 1/max(1,abs
0559     !           (in the two cases
0560     !           dimension of c to
0561     !           components c(31),
0562     !
0563     !  c(2)  floor value - used w
0564     !
0565     !  c(3)  hmin specification -
0566     !           to be abs(c(3)) -
0567     !              10*max(dwarf,r
0568     !           where dwarf is a
0569     !           rreb is the relat
0570     !
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 541

```
0541     ! program  -  the informa
0542     ! normally be changed by
0543     !
0544     !  c(1)  error control in
0545     !           max  norm  of
0546     !           weights being
0547     !              if c(1)=1
0548     !              if c(1)=2
0549     !                 control
0550     !              if c(1)=3
0551     !                 (relati
0552     !                 than th
0553     !              if c(1)=4
0554     !                 (here i
0555     !              if c(1)=5
0556     !              for all ot
0557     !                 default
0558     !                 1/max(1
0559     !           (in the two c
0560     !           dimension of
0561     !           components c(
0562     !
0563     !  c(2)  floor value - us
0564     !
0565     !  c(3)  hmin specificati
0566     !           to be abs(c(3
0567     !              10*max(dwa
0568     !           where dwarf i
0569     !           rreb is the r
0570     !
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 571

```
0571     !  c(4)  hstart specification
0572     !           an  initial  hmag
0573     !           the restrictions
0574     !           uses the default
0575     !
0576     !  c(5)  scale specification
0577     !           scale of the prob
0578     !           the method more r
0579     !           hmax  (as  descri
0580     !           acceptance requir
0581     !           of  1  is  used.
0582     !           constant coeffici
0583     !           norm  of  the  as
0584     !           approximation to
0585     !           jacobian along th
0586     !
0587     !  c(6)  hmax specification -
0588     !           if c(6).ne.0 and
0589     !              min(abs(c(6)),
0590     !           if c(6).ne.0 and
0591     !           if c(6).eq.0 and
0592     !              2/abs(c(5))
0593     !           if c(6).eq.0 and
0594     !              of 2
0595     !
0596     !  c(7)  maximum number of fu
0597     !           error  return wit
0598     !           of function evalu
0599     !
0600     !  c(8)  interrupt number  1
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 571

```
0571     !  c(4)  hstart specifica
0572     !           an  initial
0573     !           the restricti
0574     !           uses the defa
0575     !
0576     !  c(5)  scale specificat
0577     !           scale of the
0578     !           the method mo
0579     !           hmax  (as  de
0580     !           acceptance re
0581     !           of  1  is  us
0582     !           constant coef
0583     !           norm  of  the
0584     !           approximation
0585     !           jacobian alon
0586     !
0587     !  c(6)  hmax specificati
0588     !           if c(6).ne.0
0589     !              min(abs(c(
0590     !           if c(6).ne.0
0591     !           if c(6).eq.0
0592     !              2/abs(c(5)
0593     !           if c(6).eq.0
0594     !              of 2
0595     !
0596     !  c(7)  maximum number o
0597     !           error  return
0598     !           of function e
0599     !
0600     !  c(8)  interrupt number
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 601

```
0601    !              interrupt   the
0602    !              preliminary value
0603    !              and  xtrial  in
0604    !              differ from hmag
0605    !              xend  is  near) -
0606    !              will resume calcu
0607    !              re-entered with i
0608    !
0609    !  c(9)  interrupt number  2
0610    !              interrupt   the
0611    !              decided whether o
0612    !              recent  trial ste
0613    !              ind = 6 if it pla
0614    !              accepted  result,
0615    !              value, and w(*,2)
0616    !              the  subroutine
0617    !              interruption on r
0618    !              change ind in thi
0619    !              acceptance of a s
0620    !              vice versa. he ca
0621    !
0622    !*****************************
0623    !
0624    !  summary of the components
0625    !
0626    !     prescribed at the optio
0627    !           of the user
0628    !
0629    !
0630    !     c(1) error control indi
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 601

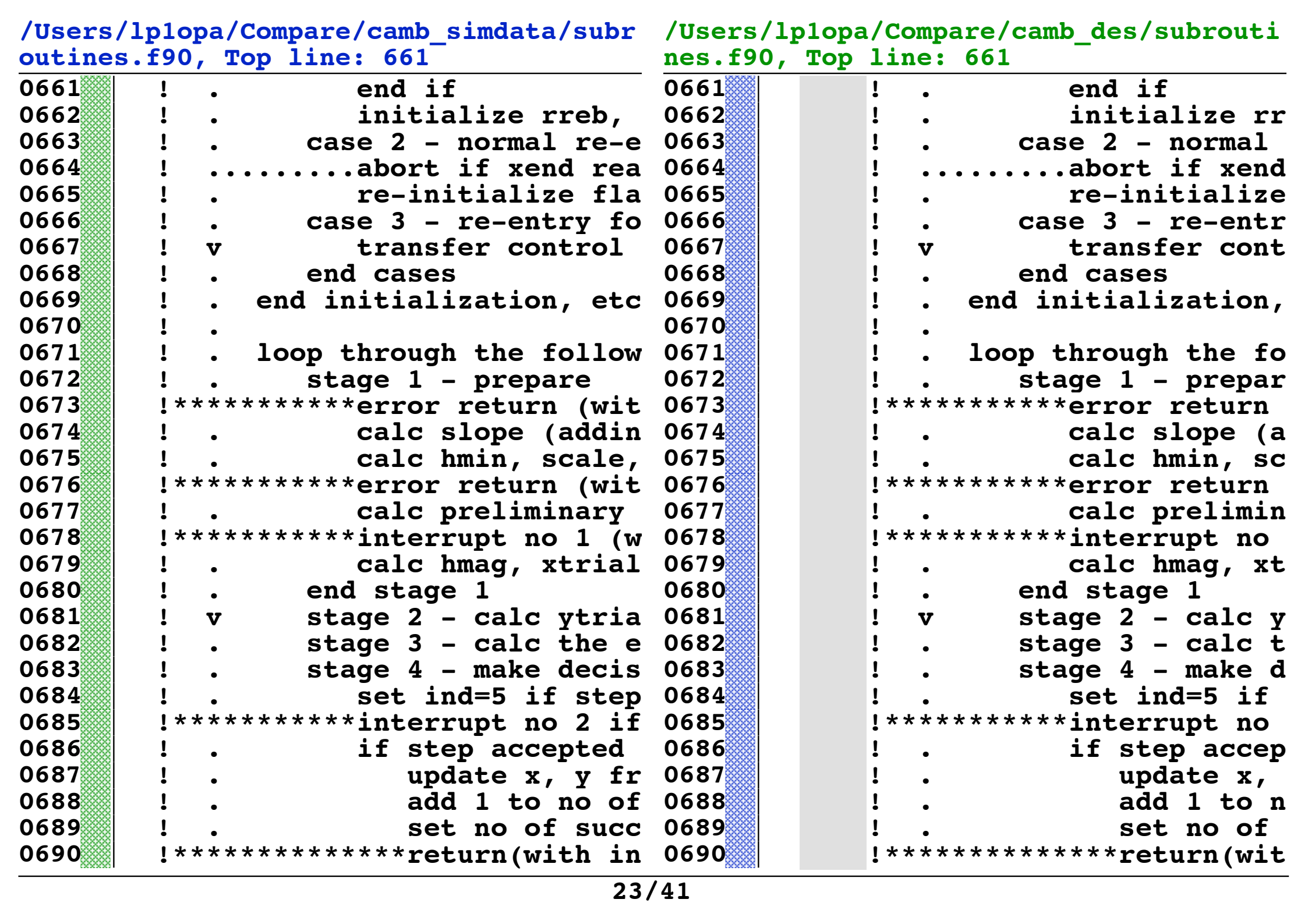
```
0601    !              interrupt   t
0602    !              preliminary v
0603    !              and  xtrial
0604    !              differ from h
0605    !              xend  is  nea
0606    !              will resume c
0607    !              re-entered wi
0608    !
0609    !  c(9)  interrupt number
0610    !              interrupt   t
0611    !              decided wheth
0612    !              recent  trial
0613    !              ind = 6 if it
0614    !              accepted  res
0615    !              value, and w(
0616    !              the  subrouti
0617    !              interruption
0618    !              change ind in
0619    !              acceptance of
0620    !              vice versa. h
0621    !
0622    !*************************
0623    !
0624    !  summary of the compone
0625    !
0626    !     prescribed at the o
0627    !           of the user
0628    !
0629    !
0630    !     c(1) error control
```

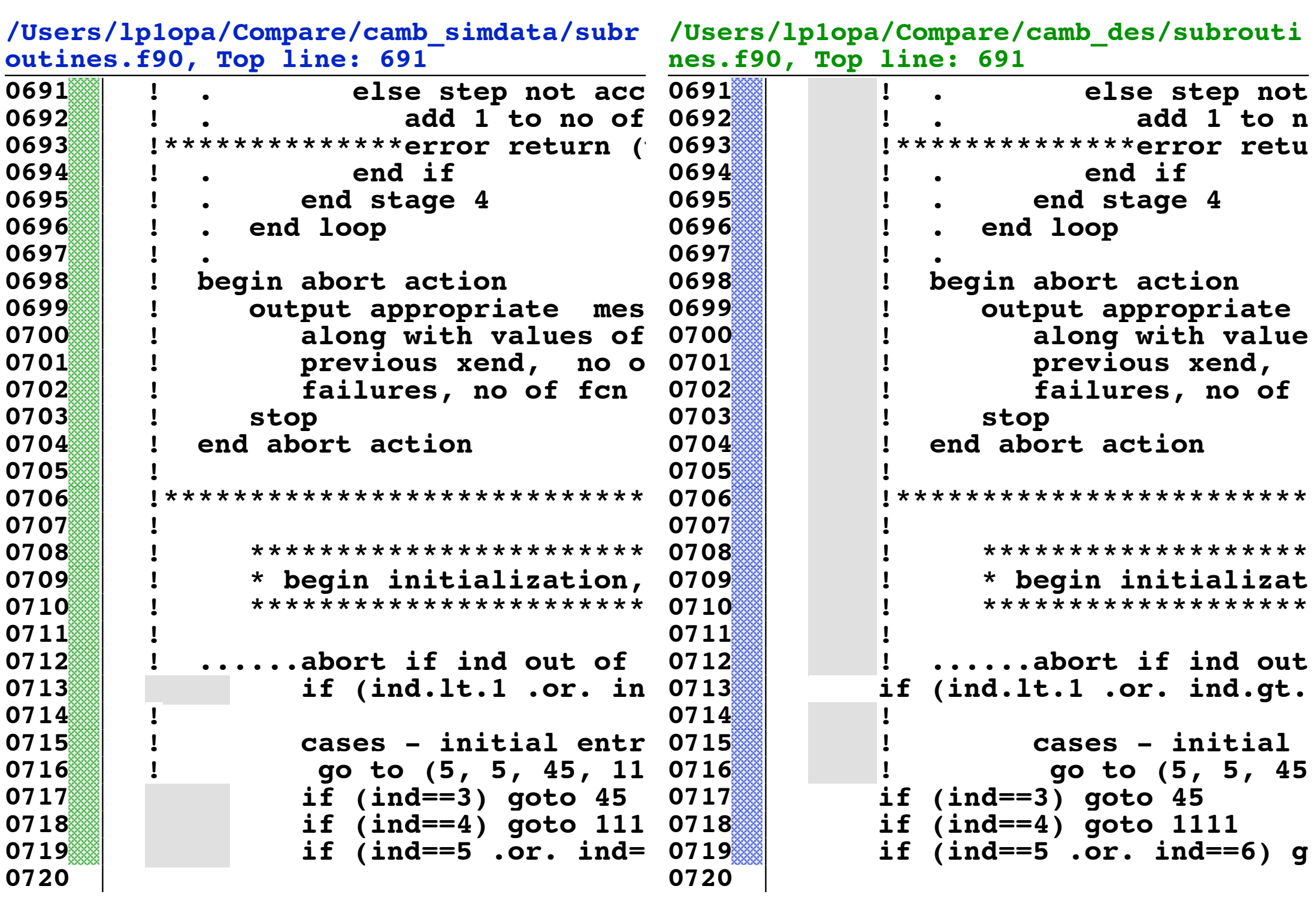
/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 631

```
0631    !     c(2) floor value
0632    !     c(3) hmin specification
0633    !     c(4) hstart specificati
0634    !     c(5) scale specificatio
0635    !     c(6) hmax specification
0636    !     c(7) max no of fcn eval
0637    !     c(8) interrupt no 1
0638    !     c(9) interrupt no 2
0639    !
0640    !
0641    !
0642    !
0643    !
0644    !
0645    !  if c(1) = 4 or 5, c(31), c
0646    !
0647    !*****************************
0648    !
0649    !  an overview of the program
0650    !
0651    !     begin initialization, p
0652    !  ......abort if ind out of
0653    !  .     cases - initial entr
0654    !  .     case 1 - initial ent
0655    !  v.......abort if n.gt.nw
0656    !  .         if initial entry
0657    !  .            set c(1) to c(
0658    !  .         else initial entr
0659    !  .            make c(1) to c
0660    !  .            make floor val
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 631

```
0631    !     c(2) floor value
0632    !     c(3) hmin specifica
0633    !     c(4) hstart specifi
0634    !     c(5) scale specific
0635    !     c(6) hmax specifica
0636    !     c(7) max no of fcn
0637    !     c(8) interrupt no 1
0638    !     c(9) interrupt no 2
0639    !
0640    !
0641    !
0642    !
0643    !
0644    !
0645    !  if c(1) = 4 or 5, c(31
0646    !
0647    !*************************
0648    !
0649    !  an overview of the pro
0650    !
0651    !     begin initializatio
0652    !  ......abort if ind out
0653    !  .     cases - initial
0654    !  .     case 1 - initial
0655    !  v.......abort if n.gt
0656    !  .         if initial en
0657    !  .            set c(1) t
0658    !  .         else initial
0659    !  .            make c(1)
0660    !  .            make floor
```


/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 661
/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 661
0661 ! . end if
0662 ! . initialize rreb,
0663 ! . case 2 - normal re-e
0664 ! .....abort if xend rea
0665 ! . re-initialize fla
0666 ! . case 3 - re-entry fo
0667 ! v transfer control
0668 ! . end cases
0669 ! . end initialization, etc
0670 ! .
0671 ! . loop through the follow
0672 ! . stage 1 - prepare
0673 !**********error return (wit
0674 ! . calc slope (addin
0675 ! . calc hmin, scale,
0676 !**********error return (wit
0677 ! . calc preliminary
0678 !**********interrupt no 1 (w
0679 ! . calc hmag, xtrial
0680 ! . end stage 1
0681 ! v stage 2 - calc ytria
0682 ! . stage 3 - calc the e
0683 ! . stage 4 - make decis
0684 ! . set ind=5 if step
0685 !**********interrupt no 2 if
0686 ! . if step accepted
0687 ! . update x, y fr
0688 ! . add 1 to no of
0689 ! . set no of succ
0690 !**********return(with in
0661 ! . end if
0662 ! . initialize rr
0663 ! . case 2 - normal
0664 ! .....abort if xend
0665 ! . re-initialize
0666 ! . case 3 - re-entr
0667 ! v transfer cont
0668 ! . end cases
0669 ! . end initialization,
0670 ! .
0671 ! . loop through the fo
0672 ! . stage 1 - prepar
0673 !**********error return
0674 ! . calc slope (a
0675 ! . calc hmin, sc
0676 !**********error return
0677 ! . calc prelimin
0678 !**********interrupt no
0679 ! . calc hmag, xt
0680 ! . end stage 1
0681 ! v stage 2 - calc y
0682 ! . stage 3 - calc t
0683 ! . stage 4 - make d
0684 ! . set ind=5 if
0685 !**********interrupt no
0686 ! . if step accep
0687 ! . update x,
0688 ! . add 1 to n
0689 ! . set no of
0690 !**********return(wit


/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 691
/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 691
0691 ! . else step not acc
0692 ! . add 1 to no of
0693 !**********error return (
0694 ! . end if
0695 ! . end stage 4
0696 ! . end loop
0697 ! .
0698 ! begin abort action
0699 ! output appropriate mes
0700 ! along with values of
0701 ! previous xend, no o
0702 ! failures, no of fcn
0703 ! stop
0704 ! end abort action
0705 !
0706 !**********
0707 !
0708 ! **********
0709 ! * begin initialization,
0710 ! **********
0711 !
0712 ! .....abort if ind out of
0713 if (ind.lt.1 .or. in
0714 !
0715 ! cases - initial entr
0716 ! go to (5, 5, 45, 11
0717 if (ind==3) goto 45
0718 if (ind==4) goto 111
0719 if (ind==5 .or. ind=
0720
0691 ! . else step not
0692 ! . add 1 to n
0693 !**********error retu
0694 ! . end if
0695 ! . end stage 4
0696 ! . end loop
0697 ! .
0698 ! begin abort action
0699 ! output appropriate
0700 ! along with value
0701 ! previous xend,
0702 ! failures, no of
0703 ! stop
0704 ! end abort action
0705 !
0706 !**********
0707 !
0708 ! **********
0709 ! * begin initializat
0710 ! **********
0711 !
0712 ! .....abort if ind out
0713 if (ind.lt.1 .or. ind.gt.
0714 !
0715 ! cases - initial
0716 ! go to (5, 5, 45
0717 if (ind==3) goto 45
0718 if (ind==4) goto 1111
0719 if (ind==5 .or. ind==6) g
0720

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 721

```
0721     !         case 1 - initial ent
0722     !  .........abort if n.gt.nw
0723              if (n.gt.nw .or.
0724              if (ind.eq. 2) go
0725     !           initial entry
0726     !           set c(1) to c(
0727              do k = 1, 9
0728                 c(k) = 0._d
0729              end do
0730              go to 35
0731        15    continue
0732     !           initial entry
0733     !           make c(1) to c
0734              do k = 1, 9
0735                 c(k) = dabs
0736              end do
0737     !           make floor val
0738              if (c(1).ne.4.
0739                 do k = 1, n
0740                    c(k+30)
0741                 end do
0742        30    continue
0743        35    continue
0744     !           initialize rreb,
0745              c(10) = 2._dl**(-
0746              c(11) = 1.d-35
0747     !           set previous xend
0748              c(20) = x
0749              do k = 21, 24
0750                 c(k) = 0._dl
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 721

```
0721     !         case 1 - initial
0722     !  .........abort if n.gt
0723           if (n.gt.nw .or. tol.le.0
0724           if (ind.eq. 2) go to 15
0725     !           initial en
0726     !           set c(1) t
0727           do k = 1, 9
0728              c(k) = 0._dl
0729           end do
0730           go to 35
0731     15    continue
0732     !           initial en
0733     !           make c(1)
0734           do k = 1, 9
0735              c(k) = dabs(c(k))
0736           end do
0737     !           make floor
0738           if (c(1).ne.4._dl .and. c
0739           do k = 1, n
0740              c(k+30) = dabs(c(k+30
0741           end do
0742     30    continue
0743     35    continue
0744     !           initialize rr
0745           c(10) = 2._dl**(-56)
0746           c(11) = 1.d-35
0747     !           set previous
0748           c(20) = x
0749           do k = 21, 24
0750              c(k) = 0._dl
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 751

```
0751            end do
0752            go to 50
0753      !   case 2 - normal re-e
0754      !  .........abort if xend rea
0755         45   if (c(21).ne.0._d
0756
0757      !         re-initialize fla
0758            c(21) = 0._dl
0759            go to 50
0760      !   case 3 - re-entry fo
0761      !      transfer control
0762      !      this has already
0763      !   end cases
0764         50 continue
0765      !
0766      !     end initialization, etc
0767      !
0768      !     ************************
0769      !     * loop through the foll
0770      !     * until the occurrence
0771      !     *    (a) the normal ret
0772      !     *        stage 4
0773      !     *    (b) an error retur
0774      !     *    (c) an interrupt r
0775      !     *        requested, in
0776      !     ************************
0777      !
0778      99999 continue
0779      !
0780      !        ********************
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 751

```
0751            end do
0752            go to 50
0753      !   case 2 - normal
0754      !  .........abort if xend
0755      45  if (c(21).ne.0._dl .and.
0756           (x.ne.c(20) .or. xend
0757      !         re-initialize
0758            c(21) = 0._dl
0759            go to 50
0760      !   case 3 - re-entr
0761      !      transfer cont
0762      !      this has alre
0763      !   end cases
0764      50  continue
0765      !
0766      !     end initialization,
0767      !
0768      !     ********************
0769      !     * loop through the
0770      !     * until the occurre
0771      !     *    (a) the normal
0772      !     *        stage 4
0773      !     *    (b) an error r
0774      !     *    (c) an interru
0775      !     *        requested,
0776      !     ********************
0777      !
0778      99999 continue
0779      !
0780      !        ****************
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 781

```
0781    !          * stage 1 - prepare
0782    !          * and some parameter
0783    !          * values of hmag, xt
0784    !          * an integration ste
0785    !          **********************
0786    !
0787    !***********error return (wit
0788                if (c(7).eq.0._dl
0789                   ind = -1
0790                   return
0791      100       continue
0792    !
0793    !           calculate slope (
0794                if (ind .eq. 6) g
0795                   call fcn(EV,n,
0796                   c(24) = c(24)
0797      105       continue
0798    !
0799    !           calculate hmin -
0800                c(13) = c(3)
0801                if (c(3) .ne. 0._
0802    !              calculate defa
0803    !              first calculat
0804    !              by the error c
0805                   temp = 0._dl
0806                   if (c(1) .ne.
0807    !                 absolute er
0808                      do 110 k =
0809                         temp = d
0810      110             continue
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 781

```
0781          !          * stage 1 - prep
0782          !          * and some param
0783          !          * values of hmag
0784          !          * an integration
0785          !          ****************
0786          !
0787          !***********error return
0788             if (c(7).eq.0._dl .or. c(
0789             ind = -1
0790             return
0791        100  continue
0792          !
0793          !           calculate slo
0794             if (ind .eq. 6) go to 105
0795             call fcn(EV,n, x, y, w(1,
0796             c(24) = c(24) + 1._dl
0797        105  continue
0798          !
0799          !           calculate hmi
0800             c(13) = c(3)
0801             if (c(3) .ne. 0._dl) go t
0802          !              calculate
0803          !              first calc
0804          !              by the err
0805             temp = 0._dl
0806             if (c(1) .ne. 1._dl) go t
0807          !                 absolut
0808             do 110 k = 1, n
0809                temp = dmax1(temp, da
0810        110  continue
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 811

```
0811        c(12) = tem
0812        go to 160
0813  115   if (c(1) .ne.
0814  !        relative er
0815  !        weighted no
0816        c(12) = 1._
0817        go to 160
0818  120   if (c(1) .ne.
0819  !        weights are
0820        do 125 k =
0821           temp = d
0822  125   continue
0823        c(12) = dmi
0824        go to 160
0825  130   if (c(1) .ne.
0826  !        weights are
0827        do 135 k =
0828           temp = d
0829  135   continue
0830        c(12) = dmi
0831        go to 160
0832  140   if (c(1) .ne.
0833  !        weights are
0834        do 145 k =
0835           temp = d
0836  145   continue
0837        c(12) = tem
0838        go to 160
0839  150   continue
0840  !        default cas
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 811

```
0811        c(12) = temp
0812        go to 160
0813  115   if (c(1) .ne. 2._dl) go t
0814        !                  relativ
0815        !                  weighte
0816        c(12) = 1._dl
0817        go to 160
0818  120   if (c(1) .ne. 3._dl) go t
0819        !                  weights
0820        do 125 k = 1, n
0821           temp = dmax1(temp, da
0822  125   continue
0823        c(12) = dmin1(temp, 1._dl
0824        go to 160
0825  130   if (c(1) .ne. 4._dl) go t
0826        !                  weights
0827        do 135 k = 1, n
0828           temp = dmax1(temp, da
0829  135   continue
0830        c(12) = dmin1(temp, 1._dl
0831        go to 160
0832  140   if (c(1) .ne. 5._dl) go t
0833        !                  weights
0834        do 145 k = 1, n
0835           temp = dmax1(temp, da
0836  145   continue
0837        c(12) = temp
0838        go to 160
0839  150   continue
0840        !                  default
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 841

```
0841                        do 155 k =
0842                           temp = d
0843     155                continue
0844                        c(12) = dmi
0845     160              continue
0846                        c(13) = 10._dl
0847     165             continue
0848     !
0849     !                 calculate scale -
0850                      c(15) = c(5)
0851                      if (c(5) .eq. 0._
0852     !
0853     !                 calculate hmax -
0854     !                 case 1 both hmax
0855                        if (c(6).ne.0.
0856
0857     !                 case 2 - hmax pre
0858                        if (c(6).ne.0.
0859     !                 case 3 - hmax not
0860                        if (c(6).eq.0.
0861     !                 case 4 - neither
0862                        if (c(6).eq.0.
0863     !
0864     !***********error return (wit
0865                   if (c(13) .le. c(
0866                      ind = -2
0867                      return
0868     170             continue
0869     !
0870     !                 calculate prelimi
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 841

```
0841          do 155 k = 1, n
0842             temp = dmax1(temp, da
0843     155  continue
0844          c(12) = dmin1(temp, 1._dl
0845     160  continue
0846          c(13) = 10._dl*dmax1(c(11
0847     165  continue
0848          !
0849          !                 calculate sca
0850          c(15) = c(5)
0851          if (c(5) .eq. 0._dl) c(15
0852          !
0853          !                 calculate hma
0854          !                 case 1 both h
0855          if (c(6).ne.0._dl .and. c
0856             c(16) = dmin1(c(6), 2
0857          !                 case 2 - hmax
0858          if (c(6).ne.0._dl .and. c
0859          !                 case 3 - hmax
0860          if (c(6).eq.0._dl .and. c
0861          !                 case 4 - neit
0862          if (c(6).eq.0._dl .and. c
0863          !
0864          !***********error return
0865          if (c(13) .le. c(16)) go
0866          ind = -2
0867          return
0868     170  continue
0869          !
0870          !                 calculate pre
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 871

```
0871        if (ind .gt. 2) g
0872     !  case 1 - initial
0873     !     any, else defa
0874           c(14) = c(4)
0875           if (c(4) .eq.
0876           go to 185
0877     175   if (c(23) .gt. 1.
0878     !     case 2 - after a
0879     !     use min(2, .9*
0880     !     overflow. then
0881           temp = 2._dl*c
0882           if (tol .lt. (
0883                 temp =
0884           c(14) = dmax1(
0885           go to 185
0886     180   continue
0887     !     case 3 - after tw
0888           c(14) = .5d0*c
0889     185   continue
0890     !
0891     !     check against hma
0892           c(14) = dmin1(c(1
0893     !
0894     !     check against hmi
0895           c(14) = dmax1(c(1
0896     !
0897     !***********interrupt no 1 (w
0898           if (c(8) .eq. 0._
0899           ind = 4
0900           return
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 871

```
0871        if (ind .gt. 2) go to 175
0872     !     case 1 - init
0873     !        any, else
0874        c(14) = c(4)
0875        if (c(4) .eq. 0._dl) c(14
0876        go to 185
0877     175 if (c(23) .gt. 1._dl) go
0878     !     case 2 - afte
0879     !        use min(2,
0880     !        overflow.
0881        temp = 2._dl*c(14)
0882        if (tol .lt. (2._dl/.9d0)
0883            temp = .9d0*(tol/c(19
0884        c(14) = dmax1(temp, .5d0*
0885        go to 185
0886     180 continue
0887     !     case 3 - afte
0888        c(14) = .5d0*c(14)
0889     185 continue
0890     !
0891     !     check against
0892        c(14) = dmin1(c(14), c(16
0893     !
0894     !     check against
0895        c(14) = dmax1(c(14), c(13
0896     !
0897     !***********interrupt no
0898        if (c(8) .eq. 0._dl) go t
0899        ind = 4
0900        return
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 901

```
0901      !               resume here on re
0902     1111             continue
0903      !
0904      !               calculate hmag, x
0905                      if (c(14) .ge. da
0906      !                  do not step mo
0907                        c(14) = dmin1(
0908                        c(17) = x + ds
0909                        go to 195
0910        190           continue
0911      !                 hit xend exact
0912                        c(14) = dabs(x
0913                        c(17) = xend
0914        195           continue
0915      !
0916      !               calculate htrial
0917                      c(18) = c(17) - x
0918      !
0919      !             end stage 1
0920      !
0921      !             ********************
0922      !             * stage 2 - calculat
0923      !             * w(*,2), ... w(*,8)
0924      !             * stage 3. w(*,9) is
0925      !             * ytrial.
0926      !             ********************
0927      !
0928                      temp = c(18)/1398
0929      !
0930                      do 200 k = 1, n
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 901

```
0901      !               resume here o
0902     1111 continue
0903      !
0904      !               calculate hma
0905      if (c(14) .ge. dabs(xend
0906      !                  do not ste
0907      c(14) = dmin1(c(14), .5d0
0908      c(17) = x + dsign(c(14),
0909      go to 195
0910     190 continue
0911      !                 hit xend e
0912      c(14) = dabs(xend - x)
0913      c(17) = xend
0914     195 continue
0915      !
0916      !               calculate htr
0917      c(18) = c(17) - x
0918      !
0919      !             end stage 1
0920      !
0921      !             ****************
0922      !             * stage 2 - calc
0923      !             * w(*,2), ... w(
0924      !             * stage 3. w(*,9
0925      !             * ytrial.
0926      !             ****************
0927      !
0928      temp = c(18)/139816908000
0929      !
0930      do 200 k = 1, n
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 931

```
0931            w(k,9) = y(k)
0932     200    continue
0933            call fcn(EV,n, x
0934     !
0935            do 205 k = 1, n
0936               w(k,9) = y(k)
0937
0938     205    continue
0939            call fcn(EV,n, x
0940     !
0941            do 210 k = 1, n
0942               w(k,9) = y(k)
0943
0944
0945     210    continue
0946            call fcn(EV,n, x
0947     !
0948            do 215 k = 1, n
0949               w(k,9) = y(k)
0950
0951
0952
0953     215    continue
0954            call fcn(EV,n, x
0955     !
0956            do 220 k = 1, n
0957               w(k,9) = y(k)
0958
0959
0960
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 931

```
0931            w(k,9) = y(k) + temp*
0932     200    continue
0933            call fcn(EV,n, x + c(18)/
0934     !
0935            do 205 k = 1, n
0936               w(k,9) = y(k) + temp*
0937                  + w(k,2)*29827607
0938     205    continue
0939            call fcn(EV,n, x + c(18)*
0940     !
0941            do 210 k = 1, n
0942               w(k,9) = y(k) + temp*
0943                  - w(k,2)*37284508
0944                  + w(k,3)*34954227
0945     210    continue
0946            call fcn(EV,n, x + c(18)*
0947     !
0948            do 215 k = 1, n
0949               w(k,9) = y(k) + temp*
0950                  + w(k,2)*12816549
0951                  - w(k,3)*92847165
0952                  + w(k,4)*12379622
0953     215    continue
0954            call fcn(EV,n, x + c(18)*
0955     !
0956            do 220 k = 1, n
0957               w(k,9) = y(k) + temp*
0958                  - w(k,2)*11185352
0959                  + w(k,3)*91726288
0960                  - w(k,4)*42721833
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 961

```
0961
0962      220         continue
0963                  call fcn(EV,n, x
0964     !
0965                  do 225 k = 1, n
0966                     w(k,9) = y(k)
0967
0968
0969
0970
0971      225         continue
0972                  call fcn(EV,n, x
0973     !
0974                  do 230 k = 1, n
0975                     w(k,9) = y(k)
0976
0977
0978
0979
0980
0981      230         continue
0982                  call fcn(EV,n, x
0983     !
0984     !            calculate ytrial,
0985     !               in w(*,9)
0986                  do 235 k = 1, n
0987                     w(k,9) = y(k)
0988
0989
0990
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 961

```
0961                     + w(k,5)*48250540
0962      220 continue
0963          call fcn(EV,n, x + c(18),
0964         !
0965          do 225 k = 1, n
0966              w(k,9) = y(k) + temp*
0967                   + w(k,2)*23116395
0968                   - w(k,3)*13220922
0969                   - w(k,4)*45300678
0970                   + w(k,5)*32687548
0971      225 continue
0972          call fcn(EV,n, x + c(18)/
0973         !
0974          do 230 k = 1, n
0975              w(k,9) = y(k) + temp*
0976                   - w(k,2)*97546680
0977                   + w(k,3)*78971103
0978                   - w(k,4)*19208266
0979                   + w(k,5)*40029897
0980                   + w(k,7)*20158600
0981      230 continue
0982          call fcn(EV,n, x + c(18),
0983         !
0984         !            calculate ytr
0985         !               in w(*,9)
0986          do 235 k = 1, n
0987              w(k,9) = y(k) + temp*
0988                   + w(k,3)*54518625
0989                   + w(k,4)*44663734
0990                   + w(k,5)*18880646
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 991

```
0991
0992
0993      235         continue
0994     !
0995     !              add 7 to the no o
0996                    c(24) = c(24) + 7
0997     !
0998     !           end stage 2
0999     !
1000     !           **********************
1001     !           * stage 3 - calculat
1002     !           * the  unweighted  a
1003     !           * step) for the unex
1004     !           * w(*,2).  then  cal
1005     !           * specified by the e
1006     !           * modify  this resul
1007     !           * unit step) for the
1008     !           **********************
1009     !
1010     !              calculate the unw
1011                    do 300 k = 1, n
1012                       w(k,2) = (   w
1013                                  + w
1014                                  - w
1015                                  + w
1016                                  + w
1017                                  - w
1018                                  - w
1019      300         continue
1020     !
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 991

```
0991                  + w(k,7)*15076875
0992                  + w(k,8)*97599465
0993     235 continue
0994         !
0995         !              add 7 to the
0996             c(24) = c(24) + 7._dl
0997         !
0998         !           end stage 2
0999         !
1000         !           **********************
1001         !           * stage 3 - calc
1002         !           * the  unweighte
1003         !           * step) for the
1004         !           * w(*,2).  then
1005         !           * specified by t
1006         !           * modify  this r
1007         !           * unit step) for
1008         !           **********************
1009         !
1010         !              calculate the
1011             do 300 k = 1, n
1012                 w(k,2) = (   w(k,1)*8
1013                      + w(k,3)*97354687
1014                      - w(k,4)*97095075
1015                      + w(k,5)*85821120
1016                      + w(k,6)*95329710
1017                      - w(k,7)*15076875
1018                      - w(k,8)*97599465
1019     300 continue
1020         !
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 1021

```
1021      !              calculate the wei
1022      !              the error control
1023                     temp = 0._dl
1024                     if (c(1) .ne. 1._
1025      !                 absolute error
1026                     do 305 k = 1,
1027                        temp = dmax
1028        305          continue
1029                     go to 360
1030        310          if (c(1) .ne. 2._
1031      !                 relative error
1032                     do 315 k = 1,
1033                        temp = dmax
1034        315          continue
1035                     go to 360
1036        320          if (c(1) .ne. 3._
1037      !                 weights are 1/
1038                     do 325 k = 1,
1039                        temp = dmax
1040
1041        325          continue
1042                     go to 360
1043        330          if (c(1) .ne. 4._
1044      !                 weights are 1/
1045                     do 335 k = 1,
1046                        temp = dmax
1047
1048        335          continue
1049                     go to 360
1050        340          if (c(1) .ne. 5._
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 1021

```
1021      !              calculate the
1022      !              the error con
1023                     temp = 0._dl
1024                     if (c(1) .ne. 1._dl) go t
1025      !                 absolute e
1026                     do 305 k = 1, n
1027                        temp = dmax1(temp,dab
1028        305          continue
1029                     go to 360
1030        310          if (c(1) .ne. 2._dl) go t
1031      !                 relative e
1032                     do 315 k = 1, n
1033                        temp = dmax1(temp, da
1034        315          continue
1035                     go to 360
1036        320          if (c(1) .ne. 3._dl) go t
1037      !                 weights ar
1038                     do 325 k = 1, n
1039                        temp = dmax1(temp, da
1040                           / dmax1(c(2), dab
1041        325          continue
1042                     go to 360
1043        330          if (c(1) .ne. 4._dl) go t
1044      !                 weights ar
1045                     do 335 k = 1, n
1046                        temp = dmax1(temp, da
1047                           / dmax1(c(k+30),
1048        335          continue
1049                     go to 360
1050        340          if (c(1) .ne. 5._dl) go t
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 1051

```
1051      !                  weights are 1/
1052                         do 345 k = 1,
1053                            temp = dmax
1054        345              continue
1055                         go to 360
1056        350            continue
1057      !                  default case -
1058                         do 355 k = 1,
1059                            temp = dmax
1060
1061        355              continue
1062        360            continue
1063      !
1064      !                calculate est - (
1065      !                   - est is inten
1066      !                   step in ytrial
1067                       c(19) = temp*c(14
1068      !
1069      !              end stage 3
1070      !
1071      !              **********************
1072      !              * stage 4 - make dec
1073      !              **********************
1074      !
1075      !                set ind=5 if step
1076                       ind = 5
1077                       if (c(19) .gt. to
1078      !
1079      !************interrupt no 2 if
1080                       if (c(9) .eq. 0._
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 1051

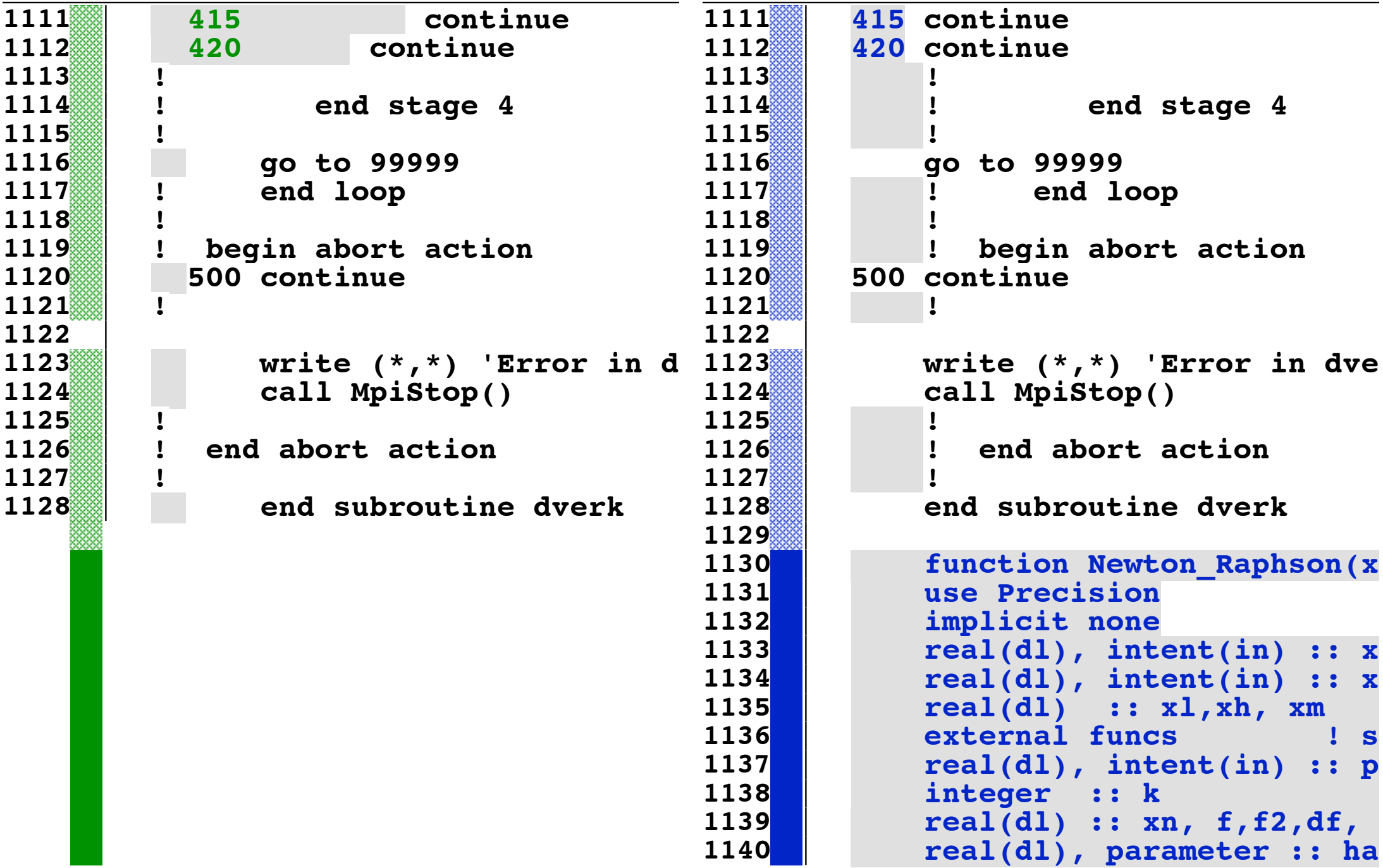
```
1051      !                  weights ar
1052            do 345 k = 1, n
1053                temp = dmax1(temp, da
1054        345 continue
1055            go to 360
1056        350 continue
1057      !                  default ca
1058            do 355 k = 1, n
1059                temp = dmax1(temp, da
1060                   / dmax1(1._dl, da
1061        355 continue
1062        360 continue
1063      !
1064      !                calculate est
1065      !                   - est is i
1066      !                   step in yt
1067            c(19) = temp*c(14)*c(15)
1068      !
1069      !              end stage 3
1070      !
1071      !              **********************
1072      !              * stage 4 - make
1073      !              **********************
1074      !
1075      !                set ind=5 if
1076            ind = 5
1077            if (c(19) .gt. tol) ind =
1078      !
1079      !************interrupt no
1080            if (c(9) .eq. 0._dl) go t
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 1081

```
1081                    return
1082    !             resume here on re
1083     2222         continue
1084    !
1085                 if (ind .eq. 6) g
1086    !               step accepted
1087    !                  ytrial, add
1088    !                  the no of s
1089                    x = c(17)
1090                    do 400 k = 1,
1091                       y(k) = w(k,
1092       400          continue
1093                    c(22) = c(22)
1094                    c(23) = 0._dl
1095    !***************return(with in
1096                    if (x .ne. xen
1097                       ind = 3
1098                       c(20) = xen
1099                       c(21) = 1._
1100                       return
1101       405          continue
1102                    go to 420
1103       410       continue
1104    !               step not accep
1105    !                  successive
1106                    c(23) = c(23)
1107    !***************error return (
1108                    if (c(14) .gt.
1109                       ind = -3
1110                       return
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 1081

```
1081                    return
1082         !             resume here o
1083      2222 continue
1084         !
1085                    if (ind .eq. 6) go to 410
1086         !                step accep
1087         !                   ytrial,
1088         !                   the no
1089                    x = c(17)
1090                    do 400 k = 1, n
1091                        y(k) = w(k,9)
1092      400 continue
1093                    c(22) = c(22) + 1._dl
1094                    c(23) = 0._dl
1095         !***************return(wit
1096                    if (x .ne. xend) go to 40
1097                    ind = 3
1098                    c(20) = xend
1099                    c(21) = 1._dl
1100                    return
1101      405 continue
1102                    go to 420
1103      410 continue
1104         !                step not a
1105         !                   success
1106                    c(23) = c(23) + 1._dl
1107         !***************error retu
1108                    if (c(14) .gt. c(13)) go
1109                    ind = -3
1110                    return
```


/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 1111
/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 1111
415 continue
420 continue
! end stage 4
go to 99999
! end loop
! begin abort action
500 continue
write (*,*) 'Error in d
call MpiStop()
! end abort action
end subroutine dverk
write (*,*) 'Error in dve
function Newton_Raphson(x
use Precision
implicit none
real(dl), intent(in) :: x
real(dl), intent(in) :: x
real(dl) :: xl,xh, xm
external funcs ! s
real(dl), intent(in) :: p
integer :: k
real(dl) :: xn, f,f2,df,
real(dl), parameter :: ha

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 1129

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 1141

```
1141                integer, parameter :: ITE
1142                real(dl), parameter :: to
1143
1144                xl =xxl
1145                xh = xxh
1146                call funcs(f,df,xl, param
1147                call funcs(f2,df,xh, para
1148                if (f*f2 > 0._dl) then
1149                    error stop 'Newton_Ra
1150                endif
1151                if (f > 0._dl) then
1152                    xm = xl
1153                    xl = xh
1154                    xh = xm
1155                endif
1156
1157                error = abs(xh-xl)
1158                xm = half*(xl+xh)
1159                k = 0
1160                do while (error > tol .an
1161                    k = k+1
1162                    call funcs(f,df,xm, p
1163                    if (f > 0._dl) then
1164                        xh = xm
1165                    else
1166                        xl = xm
1167                    endif
1168                    xn = xm - f/df
1169                    if ( (xn-xl)*(xn-xh)
1170                        xm = half* (xh+xl
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 1129

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 1171

```
1171                 error = abs(xh-xl
1172             else
1173                 error = abs(xn-xm
1174                 xm = xn
1175             endif
1176         enddo
1177
1178         if (error > tol) then
1179             write(*,*) 'Newton_Ra
1180             write(*,*) xn, f, err
1181         endif
1182
1183         end function Newton_Raphs
1184
1185
1186         subroutine Gauss_Legendre
1187         !Get Gauss-Legendre point
1188         use constants
1189         implicit none
1190         integer, intent(in) :: n
1191         real(dl), intent(out) ::
1192         real(dl), parameter :: ep
1193         integer i, j, m
1194         real(dl) p1, p2, p3, pp,
1195
1196         m=(n+1)/2
1197         !$OMP PARALLEL DO DEFAULT
1198         do i=1,m
1199             z=cos(const_pi*(i-0.2
1200             z1 = 0._dl
```

/Users/lp1opa/Compare/camb_simdata/subroutines.f90, Top line: 1129

```
1129
```

/Users/lp1opa/Compare/camb_des/subroutines.f90, Top line: 1201

```
1201          do while (abs(z-z1) >
1202              p1=1._dl
1203              p2=0._dl
1204              do j=1,n
1205                  p3=p2
1206                  p2=p1
1207                  p1=((2*j-1)*z
1208              end do
1209              pp=n*(z*p1-p2)/(z
1210              z1=z
1211              z=z1-p1/pp
1212          end do
1213          x(i)=-z
1214          x(n+1-i)=z
1215          w(i)=2/((1._dl-z**2)*
1216          w(n+1-i)=w(i)
1217      end do
1218      !$OMP END PARALLEL DO
1219
1220      end subroutine Gauss_Lege
1221
1222
```