

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1

```
0001 !Module of generally useful r
0002 !Antony Lewis, http://cosmolo
0003
0004 !April 2006: fix to TList_Rea
0005 !March 2008: fix to Ranges
0006 !June 2010: fixed bug in Del
0007
0008 module Ranges
0009 !A collection of ranges, con
0010 implicit none
0011
0012 integer, parameter :: Max_R
0013 double precision, parameter
0014     !fraction of bin width we
0015
0016 Type Region
0017     integer start_index
0018     integer steps
0019     logical :: IsLog
0020     double precision Low, Hig
0021     double precision delta
0022     double precision delta_ma
0023 end Type Region
0024
0025 Type Regions
0026
0027     integer count
0028     integer npoints
0029     double precision Lowest,
0030     Type(Region) :: R(Max_ra
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1

```
0001 !Module of generally usef
0002 !Antony Lewis, http://cos
0003
0004 !April 2006: fix to TList
0005 !March 2008: fix to Range
0006 !June 2010: fixed bug in
0007
0008 module Ranges
0009 !A collection of ranges,
0010 implicit none
0011
0012 integer, parameter :: Max
0013 double precision, paramet
0014     !fraction of bin width we
0015
0016 Type Region
0017     integer start_index
0018     integer steps
0019     logical :: IsLog
0020     double precision Low,
0021     double precision delta
0022     double precision delt
0023 end Type Region
0024
0025 Type Regions
0026
0027     integer count
0028     integer npoints
0029     double precision Lowe
0030     Type(Region) :: R(Max
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 31

```
0031      logical :: has_dpoints
0032      double precision, dimens
0033          !dpoints is (points(i+
0034
0035      end Type Regions
0036
0037      contains
0038
0039      subroutine Ranges_Init(R)
0040          Type(Regions) R
0041
0042          call Ranges_Free(R)
0043
0044      end subroutine Ranges_Ini
0045
0046      subroutine Ranges_Free(R)
0047          Type(Regions) R
0048          integer status
0049
0050          deallocate(R%points,stat
0051          deallocate(R%dpoints,sta
0052          call Ranges_Nullify(R)
0053
0054      end subroutine Ranges_Fre
0055
0056
0057      subroutine Ranges_Nullify(R
0058          Type(Regions) R
0059
0060          nullify(R%points)
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 31

```
0031      logical :: has_dpoint
0032      double precision, dim
0033          !dpoints is (points(i+
0034
0035      end Type Regions
0036
0037      contains
0038
0039      subroutine Ranges_Init(R)
0040          Type(Regions) R
0041
0042          call Ranges_Free(R)
0043
0044      end subroutine Ranges_In
0045
0046      subroutine Ranges_Free(R)
0047          Type(Regions) R
0048          integer status
0049
0050          deallocate(R%points,stat
0051          deallocate(R%dpoints,stat
0052          call Ranges_Nullify(R)
0053
0054      end subroutine Ranges_Fr
0055
0056
0057      subroutine Ranges_Nullify(R
0058          Type(Regions) R
0059
0060          nullify(R%points)
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 61

```
0061      nullify(R%dpoints)
0062      R%count = 0
0063      R%npoints = 0
0064      R%has_dpoints = .false.
0065
0066
0067      end subroutine Ranges_Nulli
0068
0069      subroutine Ranges_Assign(R,
0070          Type(Regions) R, Rin
0071
0072          call Ranges_Init(R)
0073          R = Rin
0074          nullify(R%points,R%dpoint
0075          if (associated(Rin%points
0076              call Ranges_GetArray(R,
0077          end if
0078
0079      end subroutine Ranges_Assig
0080
0081      function Ranges_IndexOf(Re
0082          Type(Regions), intent(i
0083          Type(Region), pointer ::
0084          double precision :: tau
0085          integer pointstep
0086          integer i
0087
0088          pointstep=0
0089          do i=1,Reg%count
```

/Users/lp1lopa/Compare/camb\_des/utils.F9
0, Top line: 61

```
0061      nullify(R%dpoints)
0062      R%count = 0
0063      R%npoints = 0
0064      R%has_dpoints = .false.
0065
0066
0067      end subroutine Ranges_Nul
0068
0069      subroutine Ranges_Assign(
0070          Type(Regions) R, Rin
0071
0072          call Ranges_Init(R)
0073          R = Rin
0074          nullify(R%points,R%dpoint
0075          if (associated(Rin%points
0076              call Ranges_GetArray(
0077          end if
0078
0079      end subroutine Ranges_Ass
0080
0081      function Ranges_IndexOf(R
0082          Type(Regions), intent(in)
0083          Type(Region), pointer :::
0084          double precision :: tau
0085          integer pointstep
0086          integer i
0087
0088          pointstep=0
0089          do i=1,Reg%count
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 91

```
0091      AReg => Reg%R(i)
0092
0093      if (tau < AReg%High
0094          if (AReg%IsLog)
0095              pointstep = ARe
0096          else
0097              pointstep = ARe
0098          end if
0099          return
0100      end if
0101
0102      end do
0103
0104      if (tau >= Reg%Highest)
0105          pointstep = Reg%npoi
0106      else
0107          write (*,*) 'Ranges_Index0'
0108          stop
0109      end if
0110
0111  end function Ranges_Index0
0112
0113
0114  subroutine Ranges_GetArray
0115      Type(Regions), target :: 
0116      Type(Region), pointer :: 
0117      logical, intent(in), opt
0118      integer status,i,j,ix
0119
0120
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 91

```
0091      AReg => Reg%R(i)
0092
0093      if (tau < AReg%High .
0094          if (AReg%IsLog) t
0095              pointstep = A
0096          else
0097              pointstep = A
0098          end if
0099          return
0100      end if
0101
0102      end do
0103
0104      if (tau >= Reg%Highest) t
0105          pointstep = Reg%npoi
0106      else
0107          write (*,*) 'Ranges_Index'
0108          stop
0109      end if
0110
0111  end function Ranges_Index
0112
0113
0114  subroutine Ranges_GetArra
0115      Type(Regions), target :: 
0116      Type(Region), pointer :: 
0117      logical, intent(in), opt
0118      integer status,i,j,ix
0119
0120
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 121

```
0121      if (present(want_dpoints)
0122        Reg%has_dpoints = want_
0123      else
0124        Reg%has_dpoints = .true
0125      end if

0126

0127      deallocate(Reg%points,st
0128      allocate(Reg%points(Reg%
0129

0130      ix=0
0131      do i=1, Reg%count
0132        AReg => Reg%R(i)
0133        do j = 0, AReg%steps-1
0134          ix=ix+1
0135          if (AReg%IsLog) then
0136            Reg%points(ix) = ARe
0137          else
0138            Reg%points(ix) = ARe
0139          end if
0140        end do
0141      end do
0142      ix =ix+1
0143      Reg%points(ix) = Reg%High
0144      if (ix /= Reg%npoints) s
0145
0146      if (Reg%has_dpoints) cal
0147
0148    end subroutine Ranges_GetA
0149
0150
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 121

```
0121      if (present(want_dpoints)
0122        Reg%has_dpoints = wan
0123      else
0124        Reg%has_dpoints = .tr
0125      end if

0126

0127      deallocate(Reg%points,sta
0128      allocate(Reg%points(Reg%n
0129

0130      ix=0
0131      do i=1, Reg%count
0132        AReg => Reg%R(i)
0133        do j = 0, AReg%steps-
0134          ix=ix+1
0135          if (AReg%IsLog) t
0136            Reg%points(ix)
0137          else
0138            Reg%points(ix)
0139          end if
0140        end do
0141      end do
0142      ix =ix+1
0143      Reg%points(ix) = Reg%High
0144      if (ix /= Reg%npoints) st
0145
0146      if (Reg%has_dpoints) call
0147
0148    end subroutine Ranges_Get
0149
0150
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 151

```
0151 subroutine Ranges_Getdpoi  
0152     Type(Regions), target ::  
0153     logical, intent(in), opti  
0154     integer i, status  
0155     logical halfs  
0156  
0157     if (present(half_ends))  
0158         halfs = half_ends  
0159     else  
0160         halfs = .true.  
0161     end if  
0162  
0163     deallocate(Reg%dpoints,  
0164     allocate(Reg%dpoints(Re  
0165  
0166     do i=2, Reg%npoints-1  
0167         Reg%dpoints(i) = (Reg  
0168     end do  
0169     if (halfs) then  
0170         Reg%dpoints(1) = (Reg%  
0171         Reg%dpoints(Reg%npoint  
0172     else  
0173         Reg%dpoints(1) = (Reg%  
0174         Reg%dpoints(Reg%npoint  
0175     end if  
0176 end subroutine Ranges_Getd  
0177  
0178  
0179 subroutine Ranges_Add_delt  
0180     Type(Regions), target ::
```

/Users/lp1lopa/Compare/camb\_des/utils.F9  
0, Top line: 151

```
0151 subroutine Ranges_Getdpoi  
0152     Type(Regions), target ::  
0153     logical, intent(in), opti  
0154     integer i, status  
0155     logical halfs  
0156  
0157     if (present(half_ends)) t  
0158         halfs = half_ends  
0159     else  
0160         halfs = .true.  
0161     end if  
0162  
0163     deallocate(Reg%dpoints,st  
0164     allocate(Reg%dpoints(Reg%  
0165  
0166     do i=2, Reg%npoints-1  
0167         Reg%dpoints(i) = (Reg  
0168     end do  
0169     if (halfs) then  
0170         Reg%dpoints(1) = (Reg%  
0171         Reg%dpoints(Reg%npoint  
0172     else  
0173         Reg%dpoints(1) = (Reg%  
0174         Reg%dpoints(Reg%npoint  
0175     end if  
0176 end subroutine Ranges_Get  
0177  
0178  
0179 subroutine Ranges_Add_delt  
0180     Type(Regions), target ::
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 181

```
0181      logical, intent(in), opt
0182      double precision, intent
0183      integer n
0184      logical :: WantLog
0185
0186      if (present(IsLog)) then
0187          WantLog = IsLog
0188      else
0189          WantLog = .false.
0190      end if
0191
0192      if (t_end <= t_start) &
0193          stop 'Ranges_Add_delta
0194      if (t_approx_delta <=0)
0195
0196      if (WantLog) then
0197          n = max(1,int(log(t_en
0198      else
0199          n = max(1,int((t_end-t
0200      end if
0201      call Ranges_Add(Reg,t_st
0202
0203  end subroutine Ranges_Add_
0204
0205
0206  subroutine Ranges_Add(Reg,
0207      Type(Regions), target :::
0208      logical, intent(in), opt
0209      double precision, intent
0210      integer, intent(in) ::: n
```

/Users/lp1lopa/Compare/camb\_des/utils.F9
0, Top line: 181

```
0181      logical, intent(in), opt
0182      double precision, intent(
0183      integer n
0184      logical :: WantLog
0185
0186      if (present(IsLog)) then
0187          WantLog = IsLog
0188      else
0189          WantLog = .false.
0190      end if
0191
0192      if (t_end <= t_start) &
0193          stop 'Ranges_Add_delt
0194      if (t_approx_delta <=0) s
0195
0196      if (WantLog) then
0197          n = max(1,int(log(t_
0198      else
0199          n = max(1,int((t_end-
0200      end if
0201      call Ranges_Add(Reg,t_st
0202
0203  end subroutine Ranges_Add_
0204
0205
0206  subroutine Ranges_Add(Reg
0207      Type(Regions), target :::
0208      logical, intent(in), opt
0209      double precision, intent(
0210      integer, intent(in) ::: ns
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 211

```
0211      Type(Region), pointer ::  
0212      Type(Region), target ::  
0213      double precision EndPoin  
0214      integer ixin, nreg, ix,  
0215      double precision delta  
0216      logical WantLog  
0217      double precision min_req  
0218      double precision Request  
0219  
0220      if (present(IsLog)) then  
0221          WantLog = IsLog  
0222      else  
0223          WantLog = .false.  
0224      end if  
0225  
0226      if (WantLog) then  
0227          delta = log(t_end/t_sta  
0228      else  
0229          delta = (t_end - t_star  
0230      end if  
0231  
0232      if (t_end <= t_start) st  
0233      if (nstep <=0) stop 'Ran  
0234      if (Reg%Count>= Max_Rang  
0235  
0236      !avoid IBM compiler bug, from  
0237      !      if (Reg%count > 0) NewRe  
0238      if (Reg%count > 0) THEN  
0239          DO i=1,Reg%count  
0240          NewRegions(i) = Reg%
```

/Users/lp1lopa/Compare/camb\_des/utils.F9  
0, Top line: 211

```
0211      Type(Region), pointer ::  
0212      Type(Region), target :: N  
0213      double precision EndPoint  
0214      integer ixin, nreg, ix, i  
0215      double precision delta  
0216      logical WantLog  
0217      double precision min_requ  
0218      double precision RequestD  
0219  
0220      if (present(IsLog)) then  
0221          WantLog = IsLog  
0222      else  
0223          WantLog = .false.  
0224      end if  
0225  
0226      if (WantLog) then  
0227          delta = log(t_end/t_s  
0228      else  
0229          delta = (t_end - t_st  
0230      end if  
0231  
0232      if (t_end <= t_start) sto  
0233      if (nstep <=0) stop 'Rang  
0234      if (Reg%Count>= Max_Range  
0235  
0236      !avoid IBM compiler bug,  
0237      !      if (Reg%count > 0) N  
0238      if (Reg%count > 0) THEN  
0239          DO i=1,Reg%count  
0240          NewRegions(i) = R
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 241

```
0241      END DO
0242      END IF
0243      nreg = Reg%count + 1
0244      AReg=> NewRegions(nreg)
0245      AReg%Low = t_start
0246      AReg%High = t_end
0247      AReg%delta = delta
0248      AReg%steps = nstep
0249      AReg%IsLog = WantLog
0250
0251 !Get end point in order
0252     ix = 0
0253     do i=1, nreg
0254
0255         AReg => NewRegions(i)
0256         if (ix==0) then
0257             ix = 1
0258             EndPoints(ix) = ARe
0259             ix = 2
0260             EndPoints(ix) = ARe
0261         else
0262             ixin = ix
0263             do j=1,ixin
0264                 if (AReg%Low < EndPo
0265                     EndPoints(j+1:ix+1
0266                     EndPoints(j) = ARe
0267                     ix=ix+1
0268                     exit
0269                 end if
0270             end do
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 241

```
0241      END DO
0242      END IF
0243      nreg = Reg%count + 1
0244      AReg=> NewRegions(nreg)
0245      AReg%Low = t_start
0246      AReg%High = t_end
0247      AReg%delta = delta
0248      AReg%steps = nstep
0249      AReg%IsLog = WantLog
0250
0251 !Get end point in order
0252     ix = 0
0253     do i=1, nreg
0254
0255         AReg => NewRegions(i)
0256         if (ix==0) then
0257             ix = 1
0258             EndPoints(ix) = A
0259             ix = 2
0260             EndPoints(ix) = A
0261         else
0262             ixin = ix
0263             do j=1,ixin
0264                 if (AReg%Low < EndPo
0265                     EndPoints(j+1:ix+1
0266                     EndPoints(j) = ARe
0267                     ix=ix+1
0268                     exit
0269                 end if
0270             end do
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 271

```
0271      if (ixin == ix) then
0272          ix = ix+1
0273          EndPoints(ix) = ARe
0274          ix = ix+1
0275          EndPoints(ix) = ARe
0276      else
0277          ixin = ix
0278          do j=1,ixin
0279              if (AReg%High <
0280                  EndPoints(j+1):
0281                  EndPoints(j) =
0282                      ix=ix+1
0283                      exit
0284                      end if
0285                      end do
0286          if (ixin == ix) t
0287              ix = ix+1
0288              EndPoints(ix) =
0289          end if
0290
0291      end if
0292  end if
0293
0294  end do
0295
0296 !remove duplicate points
0297     ixin = ix
0298     ix = 1
0299     do i=2, ixin
0300         if (EndPoints(i) /= En
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 271

```
0271      if (ixin == ix) t
0272          ix = ix+1
0273          EndPoints(ix)
0274          ix = ix+1
0275          EndPoints(ix)
0276      else
0277          ixin = ix
0278          do j=1,ixin
0279              if (AReg%
0280                  EndPo
0281                  EndPo
0282                      ix=ix
0283                      exit
0284                      end if
0285                      end do
0286          if (ixin == i
0287              ix = ix+1
0288              EndPoints
0289          end if
0290
0291      end if
0292  end if
0293
0294  end do
0295
0296 !remove duplicate points
0297     ixin = ix
0298     ix = 1
0299     do i=2, ixin
0300         if (EndPoints(i) /= E
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 301

```
0301          ix=ix+1
0302          EndPoints(ix) = EndPo
0303          end if
0304        end do
0305
0306
0307 !ix is the number of end poin
0308 Reg%Lowest = EndPoints(1
0309 Reg%Highest = EndPoints(
0310 Reg%count = 0
0311
0312 max_delta = Reg%Highest
0313
0314 do i=1, ix - 1
0315   AReg => Reg%R(i)
0316   AReg%Low = EndPoint
0317   AReg%High = EndPoin
0318 !
0319   max_delta = EndPoi
0320   delta = max_delta
0321   AReg%IsLog = .false
0322
0323   do j=1, nreg
0324     if (AReg%Low >= Ne
0325       if (NewRegions(j
0326         if (AReg%IsLo
0327           delta = min(
0328         else
0329           min_log_step
0330           if (min_log_
```

/Users/lp1lopa/Compare/camb\_des/utils.F9
0, Top line: 301

```
0301          ix=ix+1
0302          EndPoints(ix) = E
0303          end if
0304        end do
0305
0306
0307 !ix is the number of end
0308 Reg%Lowest = EndPoints(1)
0309 Reg%Highest = EndPoints(i
0310 Reg%count = 0
0311
0312 max_delta = Reg%Highest -
0313
0314 do i=1, ix - 1
0315   AReg => Reg%R(i)
0316   AReg%Low = EndPoints(
0317   AReg%High = EndPoints
0318 !
0319   max_delta = max_delta
0320   delta = max_delta
0321   AReg%IsLog = .false.
0322
0323   do j=1, nreg
0324     if (AReg%Low >= N
0325       if (NewRegion
0326         if (AReg%
0327           delta = min(
0328         else
0329           min_log_step
0330           if (min_log_
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 331

```
0331          max_log_st
0332          if (delta
0333              delta =
0334          else
0335              AReg%IsL
0336              delta =
0337          end if
0338      end if
0339  end if
0340  else !NewRegion
0341  if (AReg%IsLog)
0342      max_log_step
0343  if (NewRegion
0344      min_log_ste
0345      if (min_log
0346          AReg%IsL
0347          delta =
0348      else
0349          delta =
0350      end if
0351  end if
0352  else
0353      delta = min(de
0354  end if
0355  end if
0356  end if
0357 end do
0358
0359 if (AReg%IsLog) then
0360     Diff = log(AReg%Hi
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 331

```
0331          max_log_st
0332          if (delta
0333              delta =
0334          else
0335              AReg%IsL
0336              delta =
0337          end if
0338      end if
0339  end if
0340  else !NewRegion
0341  if (AReg%IsLog)
0342      max_log_step
0343  if (NewRegion
0344      min_log_ste
0345      if (min_log
0346          AReg%IsL
0347          delta =
0348      else
0349          delta =
0350      end if
0351  end if
0352  else
0353      delta = min(de
0354  end if
0355  end if
0356  end if
0357 end do
0358
0359 if (AReg%IsLog) then
0360     Diff = log(AReg%Hi
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 361

```
0361      else  
0362          Diff = AReg%High -  
0363      endif  
0364      if (delta >= Diff) t  
0365          AReg%delta = Diff  
0366          AReg%steps = 1  
0367      else  
0368          AReg%steps = max(  
0369              AReg%delta = Diff  
0370      end if  
0371  
0372      Reg%count = Reg%coun  
0373      RequestDelta(Reg%Cou  
0374  
0375      if (AReg%IsLog) then  
0376          if (AReg%steps ==1)  
0377              AReg%Delta_min = A  
0378              AReg%Delta_max = A  
0379          else  
0380              AReg%Delta_min = A  
0381              AReg%Delta_max = A  
0382          end if  
0383      else  
0384          AReg%Delta_max = A  
0385          AReg%Delta_min = A  
0386      end if  
0387  end do  
0388  
0389  
0390 !Get rid of tiny regions
```

/Users/lp1lopa/Compare/camb\_des/utils.F9  
0, Top line: 361

```
0361      else  
0362          Diff = AReg%High  
0363      endif  
0364      if (delta >= Diff) th  
0365          AReg%delta = Diff  
0366          AReg%steps = 1  
0367      else  
0368          AReg%steps = max  
0369          AReg%delta = Diff  
0370      end if  
0371  
0372      Reg%count = Reg%count  
0373      RequestDelta(Reg%Coun  
0374  
0375      if (AReg%IsLog) then  
0376          if (AReg%steps ==  
0377              AReg%Delta_mi  
0378              AReg%Delta_ma  
0379          else  
0380              AReg%Delta_mi  
0381              AReg%Delta_ma  
0382          end if  
0383      else  
0384          AReg%Delta_max =  
0385          AReg%Delta_min =  
0386      end if  
0387  end do  
0388  
0389  
0390 !Get rid of tiny regions
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 391

```
0391      ix = Reg%Count
0392      do i=ix, 1, -1
0393          AReg => Reg%R(i)
0394          if (AReg%steps ==1)
0395              Diff = AReg%High
0396              if (AReg%IsLog)
0397                  min_request =
0398                  max_request =
0399              else
0400                  min_request =
0401                  max_request =
0402              end if
0403              if (i/= Reg%Cou
0404                  LastReg => Reg
0405                  if (RequestDel
0406                      .an
0407
0408                      LastReg%Lo
0409                      if (Diff >
0410                          LastReg
0411                      end if
0412                      if (LastRe
0413                          LastReg
0414                      else
0415                          LastReg
0416                      end if
0417                      Reg%R(i:Re
0418                      Reg%Count
0419                      cycle
0420      end if
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 391

```
0391      ix = Reg%Count
0392      do i=ix, 1, -1
0393          AReg => Reg%R(i)
0394          if (AReg%steps ==1) t
0395              Diff = AReg%High
0396              if (AReg%IsLog) t
0397                  min_request =
0398                  max_request =
0399              else
0400                  min_request =
0401                  max_request =
0402              end if
0403              if (i/= Reg%Count
0404                  LastReg => Re
0405                  if (RequestDe
0406                      .and. Las
0407
0408                      LastReg%L
0409                      if (Diff >
0410                          LastR
0411                      end if
0412                      if (LastR
0413                          LastR
0414                      else
0415                          LastR
0416                      end if
0417                      Reg%R(i:R
0418                      Reg%Count
0419                      cycle
0420      end if
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 421

```
0421      end if
0422      if (i/=1) then
0423          LastReg => Reg
0424          if (RequestDel
0425              .an
0426              LastReg%Hi
0427              !AlMat08 L
0428              if (Diff >
0429                  LastReg
0430              end if
0431              if (LastRe
0432                  LastReg
0433              else
0434                  LastReg
0435              end if
0436              Reg%R(i:Re
0437              Reg%Count
0438          end if
0439      end if
0440  end if
0441 end do

0444 !Set up start indices and get
0445 nsteps = 1
0446 do i = 1, Reg%Count
0447     AReg => Reg%R(i)
0448     AReg%Start_index = n
0449     nsteps = nsteps + AR
0450     if (AReg%IsLog) then
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 421

```
0421      end if
0422      if (i/=1) then
0423          LastReg => Re
0424          if (RequestDe
0425              .and. Las
0426              LastReg%H
0427              !AlMat08
0428              if (Diff
0429                  LastR
0430              end if
0431              if (LastR
0432                  LastR
0433              else
0434                  LastR
0435              end if
0436              Reg%R(i:R
0437              Reg%Count
0438          end if
0439      end if
0440  end if
0441 end do

0444 !Set up start indices and
0445 nsteps = 1
0446 do i = 1, Reg%Count
0447     AReg => Reg%R(i)
0448     AReg%Start_index = ns
0449     nsteps = nsteps + ARe
0450     if (AReg%IsLog) then
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 451

```
0451         if (AReg%steps ==1)  
0452             AReg%Delta_min = A  
0453             AReg%Delta_max = A  
0454         else  
0455             AReg%Delta_min = A  
0456             AReg%Delta_max = A  
0457         end if  
0458     else  
0459         AReg%Delta_max = A  
0460         AReg%Delta_min = A  
0461     end if  
0462 end do  
0463  
0464 Reg%npoints = nsteps  
0465  
0466 end subroutine Ranges_Add  
0467  
0468  
0469 subroutine Ranges_Write(Re  
0470 Type(Regions), intent(i  
0471 Type(Region), pointer :  
0472 integer i  
0473  
0474 do i=1,Reg%count  
0475     AReg => Reg%R(i)  
0476     if (AReg%ISLog) the  
0477         Write (*,'("Range  
0478     else  
0479         Write (*,'("Range  
0480     end if
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 451

```
0451         if (AReg%steps ==  
0452             AReg%Delta_mi  
0453             AReg%Delta_ma  
0454         else  
0455             AReg%Delta_mi  
0456             AReg%Delta_ma  
0457         end if  
0458     else  
0459         AReg%Delta_max =  
0460         AReg%Delta_min =  
0461     end if  
0462 end do  
0463  
0464 Reg%npoints = nsteps  
0465  
0466 end subroutine Ranges_Add  
0467  
0468  
0469 subroutine Ranges_Write(R  
0470 Type(Regions), intent(in)  
0471 Type(Region), pointer ::  
0472 integer i  
0473  
0474 do i=1,Reg%count  
0475     AReg => Reg%R(i)  
0476     if (AReg%IsLog) then  
0477         Write (*,'("Range  
0478     else  
0479         Write (*,'("Range  
0480     end if
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 481

```
0481      end do
0482      end subroutine Ranges_Writ
0483
0484
0485      end module Ranges
0486
0487
0488      module Lists
0489          !Currently implements lists
0490          implicit none
0491
0492          type real_pointer
0493              real, dimension(:), point
0494          end type real_pointer
0495
0496          type double_pointer
0497              double precision, dimensi
0498          end type double_pointer
0499
0500          type String_pointer
0501              character, dimension(:),
0502          end type String_pointer
0503
0504
0505          Type TList_RealArr
0506              integer Count
0507              integer Delta
0508              integer Capacity
0509              type(Real_Pointer), dimen
0510          end Type TList_RealArr
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 481

```
0481      end do
0482      end subroutine Ranges_Wri
0483
0484
0485      end module Ranges
0486
0487
0488      module Lists
0489          !Currently implements lis
0490          implicit none
0491
0492          type real_pointer
0493              real, dimension(:), p
0494          end type real_pointer
0495
0496          type double_pointer
0497              double precision, dim
0498          end type double_pointer
0499
0500          type String_pointer
0501              character, dimension(
0502          end type String_pointer
0503
0504
0505          Type TList_RealArr
0506              integer Count
0507              integer Delta
0508              integer Capacity
0509              type(Real_Pointer), d
0510          end Type TList_RealArr
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 511

```
0511
0512      Type TStringList
0513          integer Count
0514          integer Delta
0515          integer Capacity
0516          type(String_Pointer), dim
0517      end Type TStringList
0518
0519      contains
0520
0521      subroutine TList_RealArr_I
0522          Type (TList_RealArr) :: L
0523
0524          L%Count = 0
0525          L%Capacity = 0
0526          L%Delta = 1024
0527          nullify(L%items)
0528
0529      end subroutine TList_RealA
0530
0531      subroutine TList_RealArr_C
0532          Type (TList_RealArr) :: L
0533          integer i, status
0534
0535          do i=L%Count,1,-1
0536              deallocate (L%Items(i))
0537          end do
0538          deallocate (L%Items, stat
0539          nullify(L%Items)
0540          L%Count = 0
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 511

```
0511
0512      Type TStringList
0513          integer Count
0514          integer Delta
0515          integer Capacity
0516          type(String_Pointer), dim
0517      end Type TStringList
0518
0519      contains
0520
0521      subroutine TList_RealArr_I
0522          Type (TList_RealArr) :: L
0523
0524          L%Count = 0
0525          L%Capacity = 0
0526          L%Delta = 1024
0527          nullify(L%items)
0528
0529      end subroutine TList_RealA
0530
0531      subroutine TList_RealArr_C
0532          Type (TList_RealArr) :: L
0533          integer i, status
0534
0535          do i=L%Count,1,-1
0536              deallocate (L%Items(i))
0537          end do
0538          deallocate (L%Items, stat
0539          nullify(L%Items)
0540          L%Count = 0
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 541

```
0541      L%Capacity = 0
0542
0543      end subroutine TList_RealA
0544
0545
0546      subroutine TList_RealArr_A
0547      Type (TList_RealArr) :: L
0548      real, intent(in) :: P(:)
0549      integer s
0550
0551      if (L%Count == L%Capacity)
0552      s = size(P)
0553      L%Count = L%Count + 1
0554      allocate(L%Items(L%Count))
0555      L%Items(L%Count)%P = P
0556
0557      end subroutine TList_RealA
0558
0559      subroutine TList_RealArr_S
0560      Type (TList_RealArr) :: L
0561      integer C
0562      type(Real_Pointer), dimen
0563
0564      if (L%Count > 0) then
0565          if (C < L%Count) stop '
0566          allocate(TmpItems(L%Cou
0567          TmpItems = L%Items(1:L%
0568          deallocate(L%Items)
0569          allocate(L%Items(C))
0570          L%Items(1:L%Count) = Tm
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 541

```
0541      L%Capacity = 0
0542
0543
0544
0545
0546      subroutine TList_RealArr_A
0547      Type (TList_RealArr) :: L
0548      real, intent(in) :: P(:)
0549      integer s
0550
0551      if (L%Count == L%Capacity)
0552      s = size(P)
0553      L%Count = L%Count + 1
0554      allocate(L%Items(L%Count))
0555      L%Items(L%Count)%P = P
0556
0557      end subroutine TList_RealA
0558
0559      subroutine TList_RealArr_S
0560      Type (TList_RealArr) :: L
0561      integer C
0562      type(Real_Pointer), dimen
0563
0564      if (L%Count > 0) then
0565          if (C < L%Count) stop '
0566          allocate(TmpItems(L%Cou
0567          TmpItems = L%Items(1:L%
0568          deallocate(L%Items)
0569          allocate(L%Items(C))
0570          L%Items(1:L%Count) = Tm
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 571

```
0571      deallocate(TmpItems)
0572      else
0573      allocate(L%Items(C))
0574      end if
0575      L%Capacity = C
0576      end subroutine TList_RealA

0578      subroutine TList_RealArr_D
0579      Type (TList_RealArr) :: L
0580      integer, intent(in) :: i
0581      integer status

0582      deallocate(L%items(i)%P,
0583      if (L%Count > 1) L%Items
0584      L%Count = L%Count -1

0587      end subroutine TList_RealA

0589      subroutine TList_RealArr_S
0590      Type (TList_RealArr) :: L
0591      integer, intent(in) :: fi
0592      integer i

0594      write (fid) L%Count
0595      do i=1,L%Count
0596          write(fid) size(L%Item
0597          write(fid) L%Items(i)%
0598      end do

0600      end subroutine TList_RealA
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 571

```
0571      deallocate(TmpItems)
0572      else
0573      allocate(L%Items(C))
0574      end if
0575      L%Capacity = C
0576      end subroutine TList_RealA

0578      subroutine TList_RealArr_
0579      Type (TList_RealArr) :: L
0580      integer, intent(in) :: i
0581      integer status

0582      deallocate(L%items(i)%P,
0583      if (L%Count > 1) L%Items(
0584      L%Count = L%Count -1

0587      end subroutine TList_RealA

0589      subroutine TList_RealArr_
0590      Type (TList_RealArr) :: L
0591      integer, intent(in) :: fi
0592      integer i

0594      write (fid) L%Count
0595      do i=1,L%Count
0596          write(fid) size(L%Item
0597          write(fid) L%Items(i)%
0598      end do

0600      end subroutine TList_RealA
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 601

```
0601
0602 subroutine TList_RealArr_R
0603 Type (TList_RealArr) :: L
0604 integer, intent(in) :: fi
0605 integer num,i,sz
0606
0607 call TList_RealArr_Clea
0608 read (fid) num
0609 call TList_RealArr_SetC
0610 do i=1,num
0611     read(fid) sz
0612     allocate(L%Items(i)%P(
0613         read(fid) L%Items(i)%P
0614     end do
0615     L%Count = num
0616
0617 end subroutine TList_RealA
0618
0619
0620 subroutine TList_RealArr_T
0621 Type (TList_RealArr) :: L
0622 integer, intent(in) :: i
0623 integer newCount
0624 type(Real_Pointer), dimen
0625
0626 if (L%Count > 1) then
0627     newCount = (L%Count-1)/
0628     allocate(TmpItems(newCo
0629     TmpItems = L%Items(1:L%
0630     deallocate(L%Items)
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 601

```
0601
0602 subroutine TList_RealArr_R
0603 Type (TList_RealArr) :: L
0604 integer, intent(in) :: fi
0605 integer num,i,sz
0606
0607 call TList_RealArr_Clear(
0608 read (fid) num
0609 call TList_RealArr_SetCap
0610 do i=1,num
0611     read(fid) sz
0612     allocate(L%Items(i)%P(
0613         read(fid) L%Items(i)%P
0614     end do
0615     L%Count = num
0616
0617 end subroutine TList_RealA
0618
0619
0620 subroutine TList_RealArr_T
0621 Type (TList_RealArr) :: L
0622 integer, intent(in) :: i
0623 integer newCount
0624 type(Real_Pointer), dimen
0625
0626 if (L%Count > 1) then
0627     newCount = (L%Count-1)
0628     allocate(TmpItems(newCo
0629     TmpItems = L%Items(1:L%
0630     deallocate(L%Items)
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 631

```
0631      L%Capacity = newCount
0632      allocate(L%Items(L%Capa
0633      L%Items = TmpItems
0634      L%Count = newCount
0635      deallocate(TmpItems)
0636      end if
0637  end subroutine TList_RealA
0638
0639  subroutine TList_RealArr_C
0640  !Taking the ix'th entry in
0641  !limfrac of the items betw
0642  !e.g. if limfrac = 0.05 ge
0643  Type (TList_RealArr) :: :
0644  integer, intent(IN) :: i
0645  real, intent(IN) :: limf
0646  real, intent(OUT), option
0647  integer, intent(IN), opti
0648  integer b,t,samps
0649  real pos, d
0650  type(Real_Pointer), dime
0651
0652  b=1
0653  t=L%Count
0654  if (present(ix1)) b = ix1
0655  if (present(ix2)) t = ix2
0656  samps = t - b + 1
0657
0658  allocate(SortItems(samps)
0659  SortItems = L%Items(b:t)
0660  call QuickSortArr_Real(S
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 631

```
0631      L%Capacity = newCount
0632      allocate(L%Items(L%Ca
0633      L%Items = TmpItems
0634      L%Count = newCount
0635      deallocate(TmpItems)
0636      end if
0637  end subroutine TList_Real
0638
0639  subroutine TList_RealArr_
0640  !Taking the ix'th entry i
0641  !limfrac of the items bet
0642  !e.g. if limfrac = 0.05 g
0643  Type (TList_RealArr) :: L
0644  integer, intent(IN) :: ix
0645  real, intent(IN) :: limfr
0646  real, intent(OUT), option
0647  integer, intent(IN), opti
0648  integer b,t,samps
0649  real pos, d
0650  type(Real_Pointer), dimen
0651
0652  b=1
0653  t=L%Count
0654  if (present(ix1)) b = ix1
0655  if (present(ix2)) t = ix2
0656  samps = t - b + 1
0657
0658  allocate(SortItems(samps)
0659  SortItems = L%Items(b:t)
0660  call QuickSortArr_Real(S
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 661

```
0661      if (present(Lower)) then
0662          pos = (samps-1)*limfra
0663          b = max(int(pos),1)
0664          Lower = SortItems(b)%P
0665          if (b < samps .and. pos
0666              d = pos - b
0667              Lower = Lower*(1 - d)
0668          end if
0669      end if
0670      if (present(Upper)) then
0671          pos = (samps-1)*(1.-lim
0672          b = max(int(pos),1)
0673          Upper = SortItems(b)%P(
0674              if (b < samps .and. pos
0675                  d = pos - b
0676                  Upper = Upper*(1 - d)
0677              end if
0678          end if
0679
0680      deallocate(SortItems)
0681
0682  end subroutine TList_Real
0683
0684 subroutine TStringList_Ini
0685     Type (TStringList) :: L
0686
0687     L%Count = 0
0688     L%Capacity = 0
0689     L%Delta = 128
0690     nullify(L%items)
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 661

```
0661      if (present(Lower)) then
0662          pos = (samps-1)*limfr
0663          b = max(int(pos),1)
0664          Lower = SortItems(b)%
0665          if (b < samps .and. p
0666              d = pos - b
0667              Lower = Lower*(1 -
0668          end if
0669      end if
0670      if (present(Upper)) then
0671          pos = (samps-1)*(1.-l
0672          b = max(int(pos),1)
0673          Upper = SortItems(b)%
0674              if (b < samps .and. p
0675                  d = pos - b
0676                  Upper = Upper*(1 -
0677              end if
0678      end if
0679
0680      deallocate(SortItems)
0681
0682  end subroutine TList_Real
0683
0684 subroutine TStringList_In
0685     Type (TStringList) :: L
0686
0687     L%Count = 0
0688     L%Capacity = 0
0689     L%Delta = 128
0690     nullify(L%items)
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 691

```
0691
0692      end subroutine TStringList
0693
0694      subroutine TStringList_Cle
0695          Type (TStringList) :: L
0696          integer i, status
0697
0698          do i=L%Count,1,-1
0699              deallocate (L%Items(i))
0700          end do
0701          deallocate (L%Items, stat
0702          call TStringList_Init(L)
0703
0704      end subroutine TStringList
0705
0706      subroutine TStringList_Set
0707          Type (TStringList) :: L
0708          character(Len=*), intent(
0709          character(Len=*), intent(
0710          character(LEN=1024) item
0711          integer i,j
0712          character(LEN=256) valid_
0713
0714          if (present(valid_chars_i
0715              valid_chars = valid_ch
0716          else
0717              valid_chars='abcdefghijklm
0718          endif
0719
0720          call TStringList_Clear(L
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 691

```
0691
0692      end subroutine TStringList
0693
0694      subroutine TStringList_Cle
0695          Type (TStringList) :: L
0696          integer i, status
0697
0698          do i=L%Count,1,-1
0699              deallocate (L%Items(i))
0700          end do
0701          deallocate (L%Items, stat
0702          call TStringList_Init(L)
0703
0704      end subroutine TStringList
0705
0706      subroutine TStringList_Set
0707          Type (TStringList) :: L
0708          character(Len=*), intent(
0709          character(Len=*), intent(
0710          character(LEN=1024) item
0711          integer i,j
0712          character(LEN=256) valid_
0713
0714          if (present(valid_chars_i
0715              valid_chars = valid_ch
0716          else
0717              valid_chars='abcdefghijklm
0718          endif
0719
0720          call TStringList_Clear(L
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 721

```
0721      item = ''  
0722      j=0  
0723      do i=1, len_trim(S)  
0724          if (verify(S(i:i),tri)  
0725              j=j+1  
0726              item(j:j) = S(i:i)  
0727          else  
0728              if (trim(S(i:i)) /='  
0729                  write (*,*) 'Inval  
0730                  end if  
0731                  if (j>0) call TStri  
0732                      j=0  
0733                  end if  
0734              end do  
0735              if (j>0) call TStringLis  
0736  
0737      end subroutine TStringList  
0738  
0739  
0740  
0741      subroutine TStringList_Add  
0742          Type (TStringList) :: L  
0743          character(LEN=*), intent(  
0744              integer s,i  
0745  
0746          if (L%Count == L%Capacity  
0747              s = len_trim(P)  
0748              L%Count = L%Count + 1  
0749              allocate(L%Items(L%Count)  
0750              do i=1,s
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 721

```
0721      item = ''  
0722      j=0  
0723      do i=1, len_trim(S)  
0724          if (verify(S(i:i),tri)  
0725              j=j+1  
0726              item(j:j) = S(i:i)  
0727          else  
0728              if (trim(S(i:i)) /'  
0729                  write (*,*) '  
0730                  end if  
0731                  if (j>0) call TSt  
0732                      j=0  
0733                  end if  
0734              end do  
0735              if (j>0) call TStringList  
0736  
0737      end subroutine TStringList  
0738  
0739  
0740  
0741      subroutine TStringList_Ad  
0742          Type (TStringList) :: L  
0743          character(LEN=*), intent(  
0744              integer s,i  
0745  
0746          if (L%Count == L%Capacity  
0747              s = len_trim(P)  
0748              L%Count = L%Count + 1  
0749              allocate(L%Items(L%Count)  
0750              do i=1,s
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 751

```
0751      L%Items(L%Count)%P(i) = P
0752      end do
0753      end subroutine TStringList
0754
0755      function TStringList_Item(
0756          Type (TStringList) :: L
0757          integer, intent(in) :: i
0758          integer j
0759          character(LEN=1024) S
0760
0761          S=' '
0762          if (i<=L%Count .and. i>0)
0763              do j=1,size(L%Items(i))%P
0764                  S(j:j)=L%Items(i)%P(j)
0765              end do
0766          end if
0767      end function TStringList_I
0768
0769      subroutine TStringList_Set
0770          Type (TStringList) :: L
0771          integer C
0772          type(String_Pointer), dim
0773
0774          if (L%Count > 0) then
0775              if (C < L%Count) stop '
0776              allocate(TmpItems(L%Co
0777              TmpItems = L%Items(1:L%
0778              deallocate(L%Items)
0779              allocate(L%Items(C))
0780              L%Items(1:L%Count) = Tm
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 751

```
0751      L%Items(L%Count)%P(i)
0752      end do
0753      end subroutine TStringList
0754
0755      function TStringList_Item
0756          Type (TStringList) :: L
0757          integer, intent(in) :: i
0758          integer j
0759          character(LEN=1024) S
0760
0761          S=' '
0762          if (i<=L%Count .and. i>0)
0763              do j=1,size(L%Items(i))
0764                  S(j:j)=L%Items(i)
0765              end do
0766          end if
0767      end function TStringList_I
0768
0769      subroutine TStringList_Set
0770          Type (TStringList) :: L
0771          integer C
0772          type(String_Pointer), dim
0773
0774          if (L%Count > 0) then
0775              if (C < L%Count) stop '
0776              allocate(TmpItems(L%C
0777              TmpItems = L%Items(1:L%
0778              deallocate(L%Items)
0779              allocate(L%Items(C))
0780              L%Items(1:L%Count) = Tm
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 781

```
0781      deallocate(TmpItems)
0782      else
0783      allocate(L%Items(C))
0784      end if
0785      L%Capacity = C
0786
0787      end subroutine TStringList
0788
0789      subroutine TStringList_Del
0790      Type (TStringList) :: L
0791      integer, intent(in) :: i
0792      integer status
0793
0794      deallocate(L%items(i)%P,
0795      if (L%Count > 1) L%Items
0796      L%Count = L%Count -1
0797
0798      end subroutine TStringList
0799
0800      function TStringList_Index
0801      Type (TStringList) :: L
0802      character(LEN=*), intent(
0803      integer TStringList_Index
0804
0805      slen = len_trim(S)
0806      do i=1,L%Count
0807          if ( size(L%Items(i)%P)=
0808              !Yes, comparing strings and
0809              j=1
0810              do while (L%Items(i)%P
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 781

```
0781      deallocate(TmpItems)
0782      else
0783      allocate(L%Items(C))
0784      end if
0785      L%Capacity = C
0786
0787      end subroutine TStringList
0788
0789      subroutine TStringList_Del
0790      Type (TStringList) :: L
0791      integer, intent(in) :: i
0792      integer status
0793
0794      deallocate(L%items(i)%P,
0795      if (L%Count > 1) L%Items(
0796      L%Count = L%Count -1
0797
0798      end subroutine TStringList
0799
0800      function TStringList_Index
0801      Type (TStringList) :: L
0802      character(LEN=*), intent(
0803      integer TStringList_Index
0804
0805      slen = len_trim(S)
0806      do i=1,L%Count
0807          if ( size(L%Items(i)%P)=
0808              !Yes, comparing s
0809              j=1
0810              do while (L%Items(i)%P
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 811

```
0811          j=j+1
0812          if (j>slen) then
0813              TStringList_Index
0814                  return
0815          end if
0816          end do
0817      end if
0818      end do
0819      TStringList_IndexOf=-1
0820
0821      end function TStringList_I
0822
0823
0824      recursive subroutine Qu
0825          !Sorts an array of poin
0826          integer, intent(in) :: :
0827 #ifdef __GFORTRAN
0828             type(real_pointer), dim
0829 #else
0830             type(real_pointer), dim
0831 #endif
0832             integer I, J, L
0833             real P
0834             type(real_pointer) :: t
0835
0836             L = Lin
0837             do
0838
0839                 I = L
0840                 J = R
```

/Users/lp1lopa/Compare/camb\_des/utils.F9  
0, Top line: 811

```
0811          j=j+1
0812          if (j>slen) t
0813              TStringLi
0814                  return
0815          end if
0816          end do
0817      end if
0818      end do
0819      TStringList_IndexOf=-1
0820
0821      end function TStringList_
0822
0823
0824      recursive subroutine Quic
0825          !Sorts an array of pointe
0826          integer, intent(in) :: Li
0827 #ifdef __GFORTRAN
0828             type(real_pointer), dimen
0829 #else
0830             type(real_pointer), dimen
0831 #endif
0832             integer I, J, L
0833             real P
0834             type(real_pointer) :: tem
0835
0836             L = Lin
0837             do
0838
0839                 I = L
0840                 J = R
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 841

```
0841      P = Arr((L + R)/2)%p(in
0842
0843      do
0844      do while (Arr(I)%p(index)
0845          I = I + 1
0846      end do
0847
0848      do while (Arr(J)%p(index)
0849          J = J - 1
0850      end do
0851
0852      if (I <= J) then
0853
0854          Temp%p => Arr(I)%p
0855          Arr(I)%p => Arr(J)%p
0856          Arr(J)%p => Temp%p
0857          I = I + 1
0858          J = J - 1
0859      end if
0860      if (I > J) exit
0861
0862      end do
0863      if (L < J) call QuickSort
0864      L = I
0865      if (I >= R) exit
0866      end do
0867
0868      end subroutine QuickSortA
0869
0870
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 841

```
0841      P = Arr((L + R)/2)%p(
0842
0843      do
0844      do while (Arr(I)%p(index)
0845          I = I + 1
0846      end do
0847
0848      do while (Arr(J)%p(index)
0849          J = J - 1
0850      end do
0851
0852      if (I <= J) then
0853
0854          Temp%p => Arr(I)%p
0855          Arr(I)%p => A
0856          Arr(J)%p => T
0857          I = I + 1
0858          J = J - 1
0859      end if
0860      if (I > J) exit
0861
0862      end do
0863      if (L < J) call Quick
0864      L = I
0865      if (I >= R) exit
0866      end do
0867
0868      end subroutine QuickSortA
0869
0870
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 871

```
0871  
0872      recursive subroutine Quic  
0873      !Sorts an array of pointe  
0874      integer, intent(in) ::  
0875      #ifdef __GFORTRAN  
0876      type(double_pointer), d  
0877      #else  
0878      type(double_pointer), d  
0879      #endif  
0880      integer I, J, L  
0881      double precision P  
0882      type(double_pointer) ::  
0883  
0884      L = Lin  
0885      do  
0886  
0887      I = L  
0888      J = R  
0889      P = Arr((L + R)/2)%p(in  
0890  
0891      do  
0892      do while (Arr(I)%p(inde  
0893          I = I + 1  
0894      end do  
0895  
0896      do while (Arr(J)%p(inde  
0897          J = J - 1  
0898      end do  
0899  
0900      if (I <= J) then
```

/Users/lp1lopa/Compare/camb\_des/utils.F9  
0, Top line: 871

```
0871  
0872  
0873      recursive subroutine Quic  
0874      !Sorts an array of pointe  
0875      integer, intent(in) :: Li  
0876      #ifdef __GFORTRAN  
0877      type(double_pointer), dim  
0878      #else  
0879      type(double_pointer), dim  
0880      #endif  
0881      integer I, J, L  
0882      double precision P  
0883      type(double_pointer) :: t  
0884  
0885      L = Lin  
0886      do  
0887      I = L  
0888      J = R  
0889      P = Arr((L + R)/2)%p(  
0890  
0891      do  
0892      do while (Arr(I)%p(inde  
0893          I = I + 1  
0894      end do  
0895  
0896      do while (Arr(J)%p(inde  
0897          J = J - 1  
0898      end do  
0899  
0900      if (I <= J) then
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 901

```
0901
0902      Temp%p => Arr(I)%p
0903      Arr(I)%p => Arr(J)%p
0904      Arr(J)%p => Temp%p
0905      I = I + 1
0906      J = J - 1
0907      end if
0908      if (I > J) exit
0909
0910      end do
0911      if (L < J) call QuickSort
0912      L = I
0913      if (I >= R) exit
0914      end do
0915
0916      end subroutine QuickSortA
0917
0918
0919      end module Lists
0920
0921      module AMLutils
0922      use Lists
0923
0924      #ifdef DEONLY
0925      !Comment out if linking to
0926      !CXML only has LAPACK 2.0
0927      include 'CXML_INCLUDE.F90'
0928      #endif
0929
0930
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 901

```
0901
0902      Temp%p => Arr
0903      Arr(I)%p => A
0904      Arr(J)%p => T
0905      I = I + 1
0906      J = J - 1
0907      end if
0908      if (I > J) exit
0909
0910      end do
0911      if (L < J) call Quick
0912      L = I
0913      if (I >= R) exit
0914      end do
0915
0916      end subroutine QuickSortA
0917
0918
0919      end module Lists
0920
0921      module AMLutils
0922      use Lists
0923
0924      #ifdef DEONLY
0925      !Comment out if linking t
0926      !CXML only has LAPACK 2.0
0927      include 'CXML_INCLUDE.F90'
0928      #endif
0929
0930
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 931

```
0931 #ifdef NAGF95
0932     use F90_UNIX
0933 #endif
0934
0935     implicit none
0936
0937 #ifndef NAGF95
0938 #ifndef GFC
0939 #ifndef __INTEL_COMPILER_BUIL
0940 #ifndef __GFORTRAN__
0941     integer iargc
0942     external iargc
0943 #endif
0944 #endif
0945 #endif
0946 #endif
0947
0948 #ifdef MPI
0949     include "mpif.h"
0950 #endif
0951
0952     integer :: Feedback = 1
0953     integer, parameter :: tmp_f
0954
0955
0956
0957     double precision, parameter
0958         twopi=2*pi, fourpi=4*pi
0959     double precision, parameter
0960     double precision, parameter
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 931

```
0931 #ifdef NAGF95
0932     use F90_UNIX
0933 #endif
0934
0935     implicit none
0936
0937 #ifndef NAGF95
0938 #ifndef GFC
0939 #ifndef __INTEL_COMPILER_BUIL
0940 #ifndef __GFORTRAN__
0941     integer iargc
0942     external iargc
0943 #endif
0944 #endif
0945 #endif
0946 #endif
0947
0948 #ifdef MPI
0949     include "mpif.h"
0950 #endif
0951
0952     integer :: Feedback = 1
0953     integer, parameter :: tmp_f
0954
0955
0956
0957     double precision, parameter
0958         twopi=2*pi, fourpi=4*pi
0959     double precision, parameter
0960     double precision, parameter
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 961

```
0961
0962      real, parameter :: pi_r = 3
0963
0964      logical :: flush_write = .t
0965          !True means no data lost
0966
0967      integer, parameter :: file_
0968      integer, parameter :: file_
0969
0970      logical file_units(file_uni
0971
0972 INTERFACE CONCAT
0973     module procedure concat_s
0974 END INTERFACE
0975
0976 INTERFACE RealToStr
0977     module procedure SingleToStr
0978 END INTERFACE RealToStr
0979
0980 contains
0981
0982 function new_file_unit()
0983     integer i, new_file_unit
0984     logical, save :: file_units
0985     logical notfree
0986
0987     if (.not. file_units_initiated)
0988         file_units = .false.
0989         file_units_initiated = .true.
0990     end if
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 961

```
0961
0962      real, parameter :: pi_r =
0963
0964      logical :: flush_write =
0965          !True means no data lost
0966
0967      integer, parameter :: fil
0968      integer, parameter :: fil
0969
0970      logical file_units(file_u
0971
0972 INTERFACE CONCAT
0973     module procedure concat_s
0974 END INTERFACE
0975
0976 INTERFACE RealToStr
0977     module procedure SingleToStr
0978 END INTERFACE RealToStr
0979
0980 contains
0981
0982 function new_file_unit()
0983     integer i, new_file_unit
0984     logical, save :: file_units
0985     logical notfree
0986
0987     if (.not. file_units_initiated)
0988         file_units = .false.
0989         file_units_initiated = .
0990     end if
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 991

```
0991
0992      do i=file_units_start, file
0993        if (.not. file_units(i) .a
0994          inquire(i,opened=notfree)
0995          if (notfree) cycle
0996          file_units(i)=.true.
0997          new_file_unit = i
0998          return
0999        end if
1000      end do
1001
1002      call mpiStop('No unused fil
1003
1004    end function new_file_unit
1005
1006
1007    subroutine CloseFile(i)
1008      integer, intent(in) :: i
1009
1010      close(i)
1011      file_units(i) = .false.
1012
1013    end subroutine CloseFile
1014
1015    subroutine ClearFileUnit(i)
1016      integer, intent(in) :: i
1017
1018      file_units(i) = .false.
1019
1020    end subroutine ClearFileUnit
```

/Users/lp1lopa/Compare/camb\_des/utils.F9  
0, Top line: 991

```
0991
0992      do i=file_units_start, fi
0993        if (.not. file_units(
0994          inquire(i,opened=
0995          if (notfree) cycl
0996          file_units(i)=.tr
0997          new_file_unit = i
0998          return
0999        end if
1000      end do
1001
1002      call mpiStop('No unused f
1003
1004    end function new_file_uni
1005
1006
1007    subroutine CloseFile(i)
1008      integer, intent(in) :: i
1009
1010      close(i)
1011      file_units(i) = .false.
1012
1013    end subroutine CloseFile
1014
1015    subroutine ClearFileUnit(
1016      integer, intent(in) :: i
1017
1018      file_units(i) = .false.
1019
1020    end subroutine ClearFileU
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1021

```
1021
1022      function GetParamCount()
1023          integer GetParamCount
1024
1025          GetParamCount = iargc()
1026
1027      end function GetParamCount
1028
1029      function GetMpIRank()
1030          integer GetMpIRank
1031 #ifdef MPI
1032          integer ierror
1033          call mpi_comm_rank(MPI_COMM_WORLD, GetMpIRank, ierror)
1034 #else
1035          GetMpIRank=0
1036 #endif
1037
1038      end function GetMpIRank
1039
1040      function IsMainMPI()
1041          logical IsMainMPI
1042
1043          IsMainMPI = GetMpIRank()
1044
1045      end function IsMainMPI
1046
1047      subroutine MpiStop(Msg)
1048          character(LEN=*), intent(in) :: Msg
1049          integer i
1050 #ifdef MPI
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1021

```
1021
1022      function GetParamCount()
1023          integer GetParamCount
1024
1025          GetParamCount = iargc()
1026
1027      end function GetParamCount
1028
1029      function GetMpIRank()
1030          integer GetMpIRank
1031 #ifdef MPI
1032          integer ierror
1033          call mpi_comm_rank(MPI_COMM_WORLD, GetMpIRank, ierror)
1034 #else
1035          GetMpIRank=0
1036 #endif
1037
1038      end function GetMpIRank
1039
1040      function IsMainMPI()
1041          logical IsMainMPI
1042
1043          IsMainMPI = GetMpIRank()
1044
1045      end function IsMainMPI
1046
1047      subroutine MpiStop(Msg)
1048          character(LEN=*), intent(in) :: Msg
1049          integer i
1050 #ifdef MPI
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1051

```
1051      integer ierror, MpIRank
1052      #endif
1053
1054      if (present(Msg)) write(*,
1055
1056      #ifdef MPI
1057          call mpi_comm_rank(MPI_COMM_WORLD, ierror)
1058          write (*,*) 'MpIStop: ', ierror
1059          call MPI_ABORT(MPI_COMM_WORLD, ierror)
1060      #endif
1061      i=1      !put breakpoint o
1062      stop
1063
1064      end subroutine MpIStop
1065
1066      subroutine MpIStat(MPIID,
1067      implicit none
1068      integer MPIID,MpISize
1069      #ifdef MPI
1070          integer ierror
1071          call mpi_comm_rank(MPI_COMM_WORLD, ierror)
1072          if (ierror/=MPI_SUCCESS) then
1073              call mpi_comm_size(MPI_COMM_WORLD, MpISize)
1074          else
1075              MPIID=0
1076              MpISize=1
1077      #endif
1078      end subroutine MpIStat
1079
1080      subroutine MpIQuietWait
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1051

```
1051      integer ierror, MpIRank
1052      #endif
1053
1054      if (present(Msg)) write(*,
1055
1056      #ifdef MPI
1057          call mpi_comm_rank(MPI_COMM_WORLD, ierror)
1058          write (*,*) 'MpIStop: ', ierror
1059          call MPI_ABORT(MPI_COMM_WORLD, ierror)
1060      #endif
1061      i=1      !put breakpoint o
1062      error stop
1063
1064      end subroutine MpIStop
1065
1066      subroutine MpIStat(MPIID,
1067      implicit none
1068      integer MPIID,MpISize
1069      #ifdef MPI
1070          integer ierror
1071          call mpi_comm_rank(MPI_COMM_WORLD, ierror)
1072          if (ierror/=MPI_SUCCESS) then
1073              call mpi_comm_size(MPI_COMM_WORLD, MpISize)
1074          else
1075              MPIID=0
1076              MpISize=1
1077      #endif
1078      end subroutine MpIStat
1079
1080      subroutine MpIQuietWait
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1081

```
1081      !Set MPI thread to sleep, e
1082      #ifdef MPI
1083          integer flag, ierr, STAT
1084          integer i, MpiId, MpiSiz
1085
1086          call MpiStat(MpiID, MpiS
1087          if (MpiID/=0) then
1088              do
1089                  call MPI_IPROBE(0,0,MP
1090                  if (flag/=0) then
1091                      call MPI_RECV(i,
1092                      exit
1093                  end if
1094                  call sleep(1)
1095              end do
1096          end if
1097      #endif
1098      end subroutine
1099
1100      subroutine MpiWakeQuietWait
1101      #ifdef MPI
1102          integer j, MpiId, MpiSize
1103
1104          call MpiStat(MpiID, MpiS
1105          if (MpiID==0) then
1106              do j=1, MpiSize-1
1107                  call MPI_ISSEND(Mp
1108              end do
1109          end if
1110      #endif
```

/Users/lp1lopa/Compare/camb\_des/utils.F9  
0, Top line: 1081

```
1081      !Set MPI thread to sleep,
1082      #ifdef MPI
1083          integer flag, ierr, STATU
1084          integer i, MpiId, MpiSize
1085
1086          call MpiStat(MpiID, MpiSi
1087          if (MpiID/=0) then
1088              do
1089                  call MPI_IPROBE(0
1090                  if (flag/=0) then
1091                      call MPI_RECV
1092                      exit
1093                  end if
1094                  call sleep(1)
1095              end do
1096          end if
1097      #endif
1098      end subroutine
1099
1100      subroutine MpiWakeQuietWa
1101      #ifdef MPI
1102          integer j, MpiId, MpiSize
1103
1104          call MpiStat(MpiID, MpiSi
1105          if (MpiID==0) then
1106              do j=1, MpiSize-1
1107                  call MPI_ISSEND(M
1108              end do
1109          end if
1110      #endif
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1111

```
1111 end subroutine MpiWakeQuiet
1112
1113 #ifdef __GFORTRAN__
1114
1115 ! =====
1116 function iargc ()
1117   ! =====
1118   integer iargc
1119   ! =====
1120
1121   iargc=command_argument_c
1122 end function iargc
1123
1124 ! =====
1125 subroutine getarg(num, res)
1126   ! =====
1127   integer, intent(in) :: nu
1128   character(len=*), intent(
1129   integer l, err
1130   ! =====
1131   call get_command_argument
1132 end subroutine getarg
1133
1134 #endif
1135
1136
1137 function GetParam(i)
1138
1139   character(LEN=512) GetPara
1140   integer, intent(in) :: i
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1111

```
1111 end subroutine MpiWakeQui
1112
1113 #ifdef __GFORTRAN__
1114
1115 ! =====
1116 function iargc ()
1117   ! =====
1118   integer iargc
1119   ! =====
1120
1121   iargc=command_argument_c
1122 end function iargc
1123
1124 ! =====
1125 subroutine getarg(num, re
1126   ! =====
1127   integer, intent(in) :: nu
1128   character(len=*), intent(
1129   integer l, err
1130   ! =====
1131   call get_command_argument
1132 end subroutine getarg
1133
1134 #endif
1135
1136
1137 function GetParam(i)
1138
1139   character(LEN=512) GetPar
1140   integer, intent(in) :: i
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1141

```
1141
1142      if (iargc() < i) then
1143          GetParam = ''
1144      else
1145          call getarg(i,GetParam)
1146      end if
1147  end function GetParam
1148
1149  function concat_s(S1,S2,S3,
1150      character(LEN=*), intent(i
1151      character(LEN=*), intent(i
1152      character(LEN = 1000) conc
1153
1154      concat = trim(S1) // S2
1155      if (present(S3)) then
1156          concat = trim(concat) //
1157          if (present(S4)) then
1158              concat = trim(concat)
1159              if (present(S5)) then
1160                  concat = trim(concat)
1161                  if (present(S6)) t
1162                      concat = trim(co
1163                      if (present(S7))
1164                          concat = trim(
1165                          if (present(S
1166                              concat = tr
1167                              end if
1168                                  end if
1169                                      end if
1170  end if
```

/Users/lp1lopa/Compare/camb\_des/utils.F9
0, Top line: 1141

```
1141
1142      if (iargc() < i) then
1143          GetParam = ''
1144      else
1145          call getarg(i,GetParam)
1146      end if
1147  end function GetParam
1148
1149  function concat_s(S1,S2,S
1150      character(LEN=*), intent(
1151      character(LEN=*), intent(
1152      character(LEN = 1000) con
1153
1154      concat = trim(S1) // S2
1155      if (present(S3)) then
1156          concat = trim(concat)
1157          if (present(S4)) then
1158              concat = trim(concat)
1159              if (present(S5))
1160                  concat = trim(concat)
1161                  if (present(S6)) t
1162                      concat = trim(co
1163                      if (present(S7))
1164                          concat = trim(
1165                          if (present(S
1166                              concat = tr
1167                              end if
1168                                  end if
1169                                      end if
1170  end if
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1171

```
1171      end if
1172      end if
1173
1174      end function concat_s
1175
1176      function concat_s_n(SS1,N2,S
1177          character(LEN=*>), intent(i
1178          integer, intent(in) :: N2
1179          character(LEN=*>), intent(i
1180          integer, intent(in), optio
1181          character(LEN = 1000) conc
1182
1183          concat = trim(SS1) //trim(
1184          if (present(SS3)) then
1185              concat = trim(concat) //
1186              if (present(N4)) then
1187                  concat = trim(concat)
1188                  if (present(SS5)) then
1189                      concat = trim(concat)
1190                      if (present(N6)) t
1191                          concat = trim(co
1192                          if (present(SS7))
1193                          concat = trim(co
1194                          if (present(N8))
1195                              concat = trim(
1196                              if (present(SS9)
1197                                  concat = trim(
1198                                      if (present(N
1199                                      concat = trim(
1200                                          if (present
```

/Users/lp1lopa/Compare/camb\_des/utils.F9
0, Top line: 1171

```
1171      end if
1172      end if
1173
1174      end function concat_s
1175
1176      function concat_s_n(SS1,N
1177          character(LEN=*>), intent(
1178          integer, intent(in) :: N2
1179          character(LEN=*>), intent(
1180          integer, intent(in), opti
1181          character(LEN = 1000) con
1182
1183          concat = trim(SS1) //trim(
1184          if (present(SS3)) then
1185              concat = trim(concat) //
1186              if (present(N4)) then
1187                  concat = trim(concat)
1188                  if (present(SS5))
1189                      concat = trim(concat)
1190                      if (present(N6)) t
1191                          concat = trim(co
1192                          if (present(SS7))
1193                          concat = trim(co
1194                          if (present(N8))
1195                              concat = trim(
1196                              if (present(SS9)
1197                                  concat = trim(
1198                                      if (present(N
1199                                      concat = trim(
1200                                          if (present
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1201

```
1201           concat = t
1202           end if
1203           end if
1204           end if
1205           end if
1206           end if
1207           end if
1208           end if
1209           end if
1210           end if
1211
1212       end function concat_s_n
1213
1214       subroutine Exchange(i1,i2)
1215           integer i1,i2,tmp
1216
1217           tmp=i1
1218           i1=i2
1219           i2=tmp
1220
1221   end subroutine Exchange
1222
1223       subroutine WriteS(S)
1224           character(LEN=*), intent(i
1225
1226               write (*,*) trim(S)
1227
1228   end subroutine WriteS
1229
1230       subroutine StringReplace(Fi
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1201

```
1201           e
1202           end i
1203
1204           end if
1205
1206           end if
1207           end if
1208           end if
1209           end if
1210           end if
1211
1212   end function concat_s_n
1213
1214       subroutine Exchange(i1,i2)
1215           integer i1,i2,tmp
1216
1217           tmp=i1
1218           i1=i2
1219           i2=tmp
1220
1221   end subroutine Exchange
1222
1223       subroutine WriteS(S)
1224           character(LEN=*), intent(
1225
1226               write (*,*) trim(S)
1227
1228   end subroutine WriteS
1229
1230   subroutine StringReplace(
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1231

```
1231      character(LEN=*), intent(i
1232      character(LEN=*), intent(i
1233      integer i
1234
1235      i = index(S,Finds)
1236      if (i>0) then
1237          S = S(1:i-1)//trim(RepS)
1238      end if
1239
1240  end subroutine StringReplac
1241
1242  function numcat(S, num)
1243      character(LEN=*) S
1244      character(LEN=1024) numcat
1245      integer num
1246
1247      write (numstr, *) num
1248      numcat = trim(S) // trim(a
1249      !OK, so can probably do wi
1250  end function numcat
1251
1252  function LogicalToInt(B)
1253      integer LogicalToInt
1254      logical, intent(in) :: B
1255
1256      if (B) then
1257          LogicalToInt=1
1258      else
1259          LogicalToInt=0
1260      end if
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1231

```
1231      character(LEN=*), intent(i
1232      character(LEN=*), intent(i
1233      integer i
1234
1235      i = index(S,Finds)
1236      if (i>0) then
1237          S = S(1:i-1)//trim(Re
1238      end if
1239
1240  end subroutine StringRepl
1241
1242  function numcat(S, num)
1243      character(LEN=*) S
1244      character(LEN=1024) numca
1245      integer num
1246
1247      write (numstr, *) num
1248      numcat = trim(S) // trim(
1249      !OK, so can probably do w
1250  end function numcat
1251
1252  function LogicalToInt(B)
1253      integer LogicalToInt
1254      logical, intent(in) :: B
1255
1256      if (B) then
1257          LogicalToInt=1
1258      else
1259          LogicalToInt=0
1260      end if
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1261

```
1261
1262      end function LogicalToInt
1263
1264      function IntToLogical(I)
1265          integer, intent(in) :: I
1266          logical IntToLogical
1267
1268          IntToLogical = I /= 0
1269
1270      end function IntToLogical
1271
1272      function IntToStr(I, minlen
1273          integer , intent(in) :: I
1274          character(LEN=30) IntToStr
1275          integer, intent(in), optio
1276          integer n
1277          character (LEN=20) :: form
1278
1279          if (present(minlen)) then
1280              n = minlen
1281              if (I<0) n=n+1
1282              form = concat('(',I,',n,'.','
1283              write (IntToStr,form) i
1284          else
1285              write (IntToStr,*) i
1286              IntToStr = adjustl(IntToStr
1287          end if
1288
1289      end function IntToStr
1290
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1261

```
1261
1262      end function LogicalToInt
1263
1264      function IntToLogical(I)
1265          integer, intent(in) :: I
1266          logical IntToLogical
1267
1268          IntToLogical = I /= 0
1269
1270      end function IntToLogical
1271
1272      function IntToStr(I, minlen
1273          integer , intent(in) :: I
1274          character(LEN=30) IntToStr
1275          integer, intent(in), opti
1276          integer n
1277          character (LEN=20) :: for
1278
1279          if (present(minlen)) then
1280              n = minlen
1281              if (I<0) n=n+1
1282              form = concat('(',I,',n,
1283              write (IntToStr,form)
1284          else
1285              write (IntToStr,*) i
1286              IntToStr = adjustl(In
1287          end if
1288
1289      end function IntToStr
1290
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1291

```
1291
1292      function StrToInt(S)
1293          integer :: StrToInt
1294          character(LEN=30), intent(
1295
1296              read (S,*) StrToInt
1297      end function StrToInt
1298
1299      function DoubleToStr(R, fi
1300          double precision, inten
1301          integer, intent(in), op
1302          character(LEN=30) Doubl
1303
1304          DoubleToStr = SingleToStr(
1305
1306      end function DoubleToStr
1307
1308      function SingleToStr(R, fi
1309          real, intent(in) :: R
1310          integer, intent(in), optio
1311          character(LEN=30) Singlet
1312
1313          if (abs(R)>=0.001 .or. R=
1314              write (SingleToStr,'(f12
1315
1316          SingleToStr = adjustl(Sing
1317          if (present(figs)) then
1318              SingleToStr = SingleToStr
1319          else
1320              SingleToStr = SingleToStr
```

/Users/lp1lopa/Compare/camb\_des/utils.F9
0, Top line: 1291

```
1291
1292      function StrToInt(S)
1293          integer :: StrToInt
1294          character(LEN=30), intent
1295
1296              read (S,*) StrToInt
1297      end function StrToInt
1298
1299      function DoubleToStr(R, f
1300          double precision, intent(
1301          integer, intent(in), opti
1302          character(LEN=30) DoubleT
1303
1304          DoubleToStr = SingleToStr
1305
1306      end function DoubleToStr
1307
1308      function SingleToStr(R, f
1309          real, intent(in) :: R
1310          integer, intent(in), optio
1311          character(LEN=30) Singlet
1312
1313          if (abs(R)>=0.001 .or. R=
1314              write (SingleToStr,'(
1315
1316          SingleToStr = adjustl(Sing
1317          if (present(figs)) then
1318              SingleToStr = SingleToStr
1319          else
1320              SingleToStr = Sin
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1321

```
1321      end if
1322
1323      else
1324          if (present(figs)) then
1325              write (SingleToStr,trim
1326          else
1327              write (SingleToStr,'(G9
1328          end if
1329          SingleToStr = adjustl(Si
1330      end if
1331
1332      end function SingleToStr
1333
1334      subroutine WriteFormatInts
1335          integer, intent(in) :: u
1336          character(LEN=*), intent
1337          integer, intent(in) :: i
1338          integer, intent(in),opti
1339          character(LEN=1024*16) S
1340
1341          S = formatst
1342          call StringReplace('%u',
1343          if (present(i2)) call St
1344          if (present(i3)) call St
1345          if (present(i4)) call St
1346
1347          write(unit,'(a)') trim(S
1348      end subroutine WriteFormat
1349
1350      function IndexOf(aval,arr,
```

/Users/lp1lopa/Compare/camb\_des/utils.F9  
0, Top line: 1321

```
1321      end if
1322
1323      else
1324          if (present(figs)) th
1325              write (SingleToStr
1326          else
1327              write (SingleToStr
1328          end if
1329          SingleToStr = adjustl
1330      end if
1331
1332      end function SingleToStr
1333
1334      subroutine WriteFormatInt
1335          integer, intent(in) :: un
1336          character(LEN=*), intent(
1337          integer, intent(in) :: il
1338          integer, intent(in),optio
1339          character(LEN=1024*16) S
1340
1341          S = formatst
1342          call StringReplace('%u',
1343          if (present(i2)) call Str
1344          if (present(i3)) call Str
1345          if (present(i4)) call Str
1346
1347          write(unit,'(a)') trim(S
1348      end subroutine WriteFormat
1349
1350      function IndexOf(aval,arr,
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1351

```
1351      integer, intent(in) :: n
1352      integer IndexOf, i
1353
1354      do i=1,n
1355          if (arr(i)==aval) then
1356              IndexOf= i
1357              return
1358          end if
1359      end do
1360      IndexOf = 0
1361
1362      end function IndexOf
1363
1364      function MaxIndex(arr, n)
1365          integer, intent(in) :: n
1366          real, intent(in) :: arr(
1367          integer locs(1:1), MaxIn
1368
1369          locs = maxloc(arr(1:n))
1370          MaxIndex = locs(1)
1371
1372      end function MaxIndex
1373
1374
1375      function MinIndex(arr, n)
1376          integer, intent(in) :: n
1377          real, intent(in) :: arr(
1378          integer locs(1:1), MinIn
1379
1380          locs = minloc(arr(1:n))
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1351

```
1351      integer, intent(in) :: n,
1352      integer IndexOf, i
1353
1354      do i=1,n
1355          if (arr(i)==aval) then
1356              IndexOf= i
1357              return
1358          end if
1359      end do
1360      IndexOf = 0
1361
1362      end function IndexOf
1363
1364      function MaxIndex(arr, n)
1365          integer, intent(in) :: n
1366          real, intent(in) :: arr(n
1367          integer locs(1:1), MaxInd
1368
1369          locs = maxloc(arr(1:n))
1370          MaxIndex = locs(1)
1371
1372      end function MaxIndex
1373
1374
1375      function MinIndex(arr, n)
1376          integer, intent(in) :: n
1377          real, intent(in) :: arr(n
1378          integer locs(1:1), MinInd
1379
1380          locs = minloc(arr(1:n))
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1381

```
1381      MinIndex = locs(1)
1382
1383      end function MinIndex
1384
1385
1386      subroutine TList_RealArr_S
1387          character(LEN=*), intent(
1388              Type (TList_RealArr) :: L
1389          character(LEN=20) aform
1390          integer i
1391          integer :: Plen = -1
1392          integer :: file_id
1393
1394          file_id = new_file_unit()
1395          call CreateTxtFile(fname,
1396          do i=1, L%Count
1397              if (PLen /= size(L%Items)
1398                  PLen = size(L%Items(i)%
1399                  aform = '(//trim(IntTo
1400                  end if
1401                  write (file_id,aform) L%
1402          end do
1403          call CloseFile(file_id)
1404
1405      end subroutine TList_RealA
1406
1407
1408      function ExtractFilePath(an
1409          character(LEN=*), intent(
1410          character(LEN=1024) Extra
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1381

```
1381      MinIndex = locs(1)
1382
1383      end function MinIndex
1384
1385
1386      subroutine TList_RealArr_S
1387          character(LEN=*), intent(
1388              Type (TList_RealArr) :: L
1389          character(LEN=20) aform
1390          integer i
1391          integer :: Plen = -1
1392          integer :: file_id
1393
1394          file_id = new_file_unit()
1395          call CreateTxtFile(fname,
1396          do i=1, L%Count
1397              if (PLen /= size(L%Items)
1398                  PLen = size(L%Items(i)%
1399                  aform = '(//trim(IntTo
1400                  end if
1401                  write (file_id,aform)
1402          end do
1403          call CloseFile(file_id)
1404
1405      end subroutine TList_RealA
1406
1407
1408      function ExtractFilePath(an
1409          character(LEN=*), intent(
1410          character(LEN=1024) Extra
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1411

```
1411      integer len, i
1412
1413      len = len_trim(aname)
1414      do i = len, 1, -1
1415          if (aname(i:i)=='/') t
1416              ExtractFilePath = a
1417              return
1418          end if
1419      end do
1420      ExtractFilePath = ''
1421
1422      end function ExtractFilePath
1423
1424      function ExtractFileExt(ana
1425          character(LEN=*), intent(
1426          character(LEN=120) Extrac
1427          integer len, i
1428
1429          len = len_trim(aname)
1430          do i = len, 1, -1
1431              if (aname(i:i)=='/') t
1432                  ExtractFileExt = ''
1433                  return
1434              else if (aname(i:i)=='.')
1435                  ExtractFileExt= ana
1436                  return
1437              end if
1438          end do
1439          ExtractFileExt = ''
1440
```

/Users/lp1lopa/Compare/camb\_des/utils.F9
0, Top line: 1411

```
1411      integer len, i
1412
1413      len = len_trim(aname)
1414      do i = len, 1, -1
1415          if (aname(i:i)=='/')
1416              ExtractFilePath =
1417              return
1418          end if
1419      end do
1420      ExtractFilePath = ''
1421
1422      end function ExtractFilePath
1423
1424      function ExtractFileExt(a
1425          character(LEN=*), intent(
1426          character(LEN=120) Extrac
1427          integer len, i
1428
1429          len = len_trim(aname)
1430          do i = len, 1, -1
1431              if (aname(i:i)=='/')
1432                  ExtractFileExt = ''
1433                  return
1434              else if (aname(i:i)=='.')
1435                  ExtractFileExt= a
1436                  return
1437              end if
1438          end do
1439          ExtractFileExt = ''
1440
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1441

```
1441 end function ExtractFileExt
1442
1443
1444 function ExtractFileName(aname
1445     character(LEN=*), intent(
1446     character(LEN=1024) Extra
1447     integer len, i
1448
1449     len = len_trim(aname)
1450     do i = len, 1, -1
1451         if (aname(i:i)=='/') t
1452             ExtractFileName = a
1453             return
1454         end if
1455     end do
1456     ExtractFileName = aname
1457
1458 end function ExtractFileName
1459
1460 function ChangeFileExt(aname
1461     character(LEN=*), intent(
1462     character(LEN=1024) Chang
1463     integer len, i
1464
1465     len = len_trim(aname)
1466     do i = len, 1, -1
1467         if (aname(i:i)=='.') t
1468             ChangeFileExt = ana
1469             return
1470         end if
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1441

```
1441 end function ExtractFileExt
1442
1443
1444 function ExtractFileName(aname
1445     character(LEN=*), intent(
1446     character(LEN=1024) Extra
1447     integer len, i
1448
1449     len = len_trim(aname)
1450     do i = len, 1, -1
1451         if (aname(i:i)=='/') t
1452             ExtractFileName = a
1453             return
1454         end if
1455     end do
1456     ExtractFileName = aname
1457
1458 end function ExtractFileName
1459
1460 function ChangeFileExt(aname
1461     character(LEN=*), intent(
1462     character(LEN=1024) Chang
1463     integer len, i
1464
1465     len = len_trim(aname)
1466     do i = len, 1, -1
1467         if (aname(i:i)=='.') t
1468             ChangeFileExt = ana
1469             return
1470         end if
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1471

```
1471      end do
1472      ChangeFileExt = trim(anam
1473
1474      end function ChangeFileExt
1475
1476
1477      function CheckTrailingSlash
1478          character(LEN=*) , intent
1479          character(LEN=1024) Chec
1480          integer len
1481
1482          len = len_trim(aname)
1483 #ifdef IBMXL
1484          if (aname(len:len) /= '\
1485 #else
1486 #ifdef ESCAPEBACKSLASH
1487          if (aname(len:len) /= '\
1488 #else
1489          if (aname(len:len) /= '\
1490 #endif
1491 #endif
1492          CheckTrailingSlash = tr
1493          else
1494          CheckTrailingSlash = an
1495          end if
1496
1497
1498      end function CheckTrailing
1499
1500
```

/Users/lp1lopa/Compare/camb\_des/utils.F9
0, Top line: 1471

```
1471      end do
1472      ChangeFileExt = trim(anam
1473
1474      end function ChangeFileEx
1475
1476
1477      function CheckTrailingSla
1478          character(LEN=*) , intent(
1479          character(LEN=1024) Check
1480          integer len
1481
1482          len = len_trim(aname)
1483 #ifdef IBMXL
1484          if (aname(len:len) /= '\\\
1485 #else
1486 #ifdef ESCAPEBACKSLASH
1487          if (aname(len:len) /= \
1488 #else
1489          if (aname(len:len) /=
1490 #endif
1491 #endif
1492          CheckTrailing
1493          else
1494          CheckTrailing
1495          end if
1496
1497
1498      end function CheckTr
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1501

```
1501 subroutine DeleteFile(aname
1502     character(LEN=*), intent(
1503         integer file_id
1504
1505             if (FileExists(aname)) t
1506                 file_id = new_file_unit
1507                 open(unit = file_id, fi
1508                     close(unit = file_id, s
1509                         file_units(file_id) = .
1510                     end if
1511
1512     end subroutine DeleteFile
1513
1514
1515 subroutine FlushFile(aunit)
1516 #ifdef __INTEL_COMPILER_BUILD
1517     USE IFPORT
1518 #endif
1519     integer, intent(IN) :: au
1520
1521
1522 #ifdef IBMXL
1523     call flush_(aunit)
1524 #else
1525     call flush(aunit)
1526 #endif
1527
1528     end subroutine FlushFile
1529
1530
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1501

```
1501 subroutine DeleteFile
1502     character(LEN=*), int
1503         integer file_id
1504
1505             if (FileExists(aname)
1506                 file_id = new fil
1507                 open(unit = file_
1508                     close(unit = file_
1509                         file_units(file_i
1510                     end if
1511
1512     end subroutine Delete
1513
1514
1515 subroutine FlushFile(
1516     __INTEL_COMPILER_BUILD
1517     USE IFPORT
1518 #endif
1519     integer, intent(IN) :
1520
1521
1522 #ifdef IBMXL
1523     call flush_(aunit)
1524 #else
1525     call flush(aunit)
1526 #endif
1527
1528     end subroutine FlushF
1529
1530
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1531

```
1531      function FileExists(aname)
1532          character(LEN=*), intent(
1533              logical FileExists
1534
1535                  inquire(file=aname, e
1536
1537      end function FileExists
1538
1539      subroutine OpenFile(aname, a
1540          character(LEN=*), intent(I
1541          integer, intent(in) :: aun
1542
1543          open(unit=aunit,file=aname
1544          return
1545
1546
1547 500 call MPIStop('File not fo
1548
1549
1550      end subroutine OpenFile
1551
1552
1553      subroutine OpenTxtFile(aname
1554          character(LEN=*), intent(I
1555          integer, intent(in) :: aun
1556
1557          call OpenFile(aname,aunit,
1558
1559      end subroutine OpenTxtFile
1560
```

/Users/lp1lopa/Compare/camb\_des/utils.F9
0, Top line: 1531

```
1531      function FileExists(a
1532          character(LEN=*), int
1533              logical FileExists
1534
1535                  inquire(file=aname, e
1536
1537      end function FileExis
1538
1539      subroutine OpenFile(a
1540          character(LEN=*), int
1541          integer, intent(in) :
1542
1543
1544          open(unit=aunit,file=
1545          return
1546
1547
1548 500 call MPIStop('File no
1549
1550
1551
1552
1553      subroutine OpenTxtFil
1554          character(LEN=*), int
1555          integer, intent(in) :
1556
1557          call OpenFile(aname,a
1558
1559      end subroutine OpenTx
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1561

```
1561 subroutine CreateOpenTxtFile(  
1562     character(LEN=*), intent(I  
1563     integer, intent(in) :: aun  
1564     logical, optional, intent(  
1565     logical A  
1566  
1567     if (present	append)) then  
1568         A=append  
1569     else  
1570         A = .false.  
1571     endif  
1572  
1573     call CreateOpenFile(aname,  
1574  
1575 end subroutine CreateOpenTxt  
1576  
1577  
1578 subroutine CreateTxtFile(aname  
1579     character(LEN=*), intent(I  
1580     integer, intent(in) :: aun  
1581  
1582     call CreateFile(aname,auni  
1583  
1584 end subroutine CreateTxtFile  
1585  
1586  
1587 subroutine CreateFile(aname,  
1588     character(LEN=*), intent(I  
1589     integer, intent(in) :: aun  
1590
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1560

```
1560 subroutine CreateOpen  
1561 character(LEN=*), int  
1562 integer, intent(in) :  
1563 logical, optional, in  
1564 logical A  
1565  
1566 if (present-append))  
1567 A=append  
1568 else  
1569 A = .false.  
1570 endif  
1571  
1572 call CreateOpenFile(a  
1573  
1574 end subroutine Create  
1575  
1576  
1577 subroutine CreateTxtF  
1578 character(LEN=*), int  
1579 integer, intent(in) :  
1580  
1581 call CreateFile(aname  
1582  
1583 end subroutine Create  
1584  
1585  
1586 subroutine CreateFile  
1587 character(LEN=*), int  
1588 integer, intent(in) :  
1589
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1591

```
1591      open(unit=aunit,file=ana
1592      return
1593
1594
1595 500 call MPIStop('Error creat
1596
1597
1598 end subroutine CreateFile
1599
1600 subroutine CreateOpenFile(an
1601     character(LEN=*), intent(
1602     integer, intent(in) :: aun
1603     logical, optional, intent(
1604     logical A
1605
1606     if (present	append)) then
1607         A=append
1608     else
1609         A = .false.
1610     endif
1611
1612     if (A) then
1613         open(unit=aunit,file=ana
1614     else
1615         open(unit=aunit,file=ana
1616     end if
1617
1618     return
1619
1620 500 call MPIStop('Error creat
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1590

```
1590      open(unit=aunit,file=
1591
1592      return
1593
1594
1595
1596
1597
1598
1599
1600 subroutine CreateOpen
1601     character(LEN=*), int
1602     integer, intent(in) :
1603     logical, optional, in
1604     logical A
1605
1606     if (present	append)) then
1607         A=append
1608     else
1609         A = .false.
1610     endif
1611
1612     if (A) then
1613         open(unit=aunit,f
1614     else
1615         open(unit=aunit,f
1616     end if
1617
1618
1619
1620 500 call MPIStop('Error c
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1621

```
1621
1622
1623      end subroutine CreateOpenFil
1624
1625      function TxtNumberColumns(In
1626          character(LEN=*) :: InLine
1627          integer n,i
1628          logical isNum
1629
1630          n=0
1631          isNum=.false.
1632          do i=1, len_trim(InLine)
1633              if (verify(InLine(i:i),'-'
1634                  if (.not. IsNum) n=n+1
1635                  IsNum=.true.
1636              else
1637                  IsNum=.false.
1638              end if
1639          end do
1640
1641      end function TxtNumberColumn
1642
1643      function TxtColumns(InLine)
1644          character(LEN=*) :: InLine
1645          integer n,i
1646          logical isNum
1647
1648          n=0
1649          isNum=.false.
1650          do i=1, len_trim(InLine)
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1620

```
1620
1621
1622      end subroutine Create
1623
1624      function TxtNumberCol
1625          character(LEN=*) :: I
1626          integer n,i
1627          logical isNum
1628
1629          n=0
1630          isNum=.false.
1631          do i=1, len_trim(InLI
1632              if (verify(InLine
1633                  if (.not. ISN
1634                  IsNum=.true.
1635              else
1636                  IsNum=.false.
1637              end if
1638          end do
1639
1640      end function TxtNumbe
1641
1642      function TxtColumns(I
1643          character(LEN=*) :: I
1644          integer n,i
1645          logical isNum
1646
1647          n=0
1648          isNum=.false.
1649          do i=1, len_trim(InLi
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1651

```
1651      if (InLine(i:i) > char(32)
1652          if (.not. IsNum) n=n+1
1653              IsNum=.true.
1654          else
1655              IsNum=.false.
1656          end if
1657      end do
1658
1659  end function TxtColumns
1660
1661  function FileColumns(aunit)
1662      integer, intent(in) :: aun
1663      integer n
1664      character(LEN=4096*32) :: 
1665
1666      n=0
1667      read(aunit,'(a)', end = 10
1668      n = TxtNumberColumns(InLin
1669 10 rewind aunit
1670
1671  end function FileColumns
1672
1673  function FileLines(aunit) re
1674      integer, intent(in) :: aun
1675      integer n
1676      character(LEN=4096) :: InL
1677
1678      n=0
1679      do
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1650

```
1650      if (InLine(i:i) >
1651          if (.not. IsN
1652              IsNum=.true.
1653          else
1654              IsNum=.false.
1655          end if
1656      end do
1657
1658  end function TxtColumn
1659
1660  function FileColumns(
1661      integer, intent(in) :
1662      integer n
1663      character(LEN=4096*32
1664
1665      n=0
1666      read(aunit,'(a)', end
1667      n = TxtNumberColumns(
1668      rewind aunit
1669
1670  end function FileColu
1671
1672  function FileLines(aunit)
1673      integer, intent(in) :
1674      integer n
1675      character(LEN=4096) :
1676
1677      n=0
1678      do
1679
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1681

```
1681      read(aunit,'(a)', end = 20
1682      n = n+1
1683      end do
1684
1685 200 rewind aunit
1686
1687
1688 end function FileLines
1689
1690
1691 function TopCommentLine(anam
1692     character(LEN=*), intent(
1693     integer file_id
1694     character(LEN=1024) :: In
1695
1696     res = ''
1697     file_id = new_file_unit()
1698     call OpenTxtFile(aname, f
1699     InLine=''
1700     do while (InLine /= '')
1701         read(file_id,'(a)', end
1702         end do
1703         If (InLine(1:1)=='#') the
1704             res = InLine
1705         end if
1706
1707 10 call CloseFile(file_id)
1708
1709 end function TopCommentLine
1710
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1680

```
1680      read(aunit,'(a)', n = n+1
1681      end do
1682
1683
1684 200 rewind aunit
1685
1686
1687
1688
1689
1690
1691 function TopCommentLi
1692     character(LEN=*), int
1693     integer file_id
1694     character(LEN=1024) :
1695
1696     res = ''
1697     file_id = new_file_un
1698     call OpenTxtFile(anam
1699     InLine=''
1700     do while (InLine /= '
1701         read(file_id,'(a)
1702         end do
1703         If (InLine(1:1)=='#') the
1704             res = InLine
1705         end if
1706
1707 10 call CloseFile(file_i
1708
1709 end function TopComme
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1711

```
1711
1712      function TxtFileColumns(anam
1713          character(LEN=*), intent(
1714              integer n, file_id
1715
1716
1717          file_id = new_file_unit()
1718
1719          call OpenTxtFile(aname, f
1720          n = FileColumns(file_id)
1721          call CloseFile(file_id)
1722
1723      end function TxtFileColumns
1724
1725
1726      function LastFileLine(aname)
1727          character(LEN=*), intent(I
1728          character(LEN = 5000) Last
1729          integer file_id
1730
1731          file_id = new_file_unit()
1732
1733          InLine = ''
1734          call OpenTxtFile(aname,fil
1735          do
1736              read(file_id,'(a)', end =
1737          end do
1738
1739          200 call CloseFile(file_id)
1740
```

/Users/lp1lopa/Compare/camb\_des/utils.F9
0, Top line: 1710

```
1710
1711      function TxtFileColumn
1712          character(LEN=*), int
1713          integer n, file_id
1714
1715
1716          file_id = new_file_un
1717
1718          call OpenTxtFile(anam
1719          n = FileColumns(file_
1720          call CloseFile(file_i
1721
1722      end function TxtFileC
1723
1724
1725      function LastFileLine
1726          character(LEN=*), int
1727          character(LEN = 5000)
1728          integer file_id
1729
1730          file_id = new_file_un
1731
1732          InLine = ''
1733          call OpenTxtFile(anam
1734          do
1735              read(file_id,'(a)
1736          end do
1737
1738          200 call CloseFile(file_i
1739
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1741

```
1741      LastFileLine = InLine
1742
1743      end function LastFileLine
1744
1745      subroutine writeArrayLine(un
1746        real, intent(in) :: arr(:)
1747        integer, intent(in) :: unit
1748
1749        write(unit,concat('(',size(
1750
1751      end subroutine writeArrayLin
1752
1753
1754      subroutine spline_real(x
1755        integer, intent(in) :: d
1756        integer, parameter :: d
1757        real(dp), intent(in) :: d
1758        real(dp), intent(out) :: d
1759        real(dp), dimension(:),
1760        integer i
1761        real(dp) xp,sig,xxdiv,d
1762
1763      allocate(u(1:n-1))
1764
1765      d2(1)=0._dp
1766      u(1)=0._dp
1767
1768      d1r= (y(2)-y(1))/(x(2)-
1769      do i=2,n-1
1770        d1l=d1r
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1740

```
1740      LastFileLine = InLine
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1771

```
1771      d1r=(y(i+1)-y(i))/(x( 1770
1772      xxd1v=1._dp/(x(i+1)-x 1771
1773      sig=(x(i)-x(i-1))*xxd 1772
1774      xp=1._dp/(sig*d2(i-1) 1773
1775      d2(i)=(sig-1._dp)*xp 1774
1776      u(i)=(6._dp*(d1r-d1l) 1775
1777      end do 1776
1778
1779      d2(n)=0._dp 1777
1780      do i=n-1,1,-1 1778
1781      d2(i)=d2(i)*d2(i+1)+u 1779
1782      end do 1780
1783
1784      deallocate(u) 1781
1785      end subroutine spline_rea 1782
1786
1787      subroutine spline_double 1783
1788      integer, intent(in) :: 1787
1789      integer, parameter :: d 1788
1790      real(dp), intent(in) :: 1789
1791      real(dp), intent(out) : 1790
1792      real(dp), dimension(:), 1791
1793      integer i 1792
1794      real(dp) xp,sig,xxdiv,d 1793
1795
1796      allocate(u(1:n-1)) 1794
1797
1798      d2(1)=0._dp 1795
1799      u(1)=0._dp 1796
1800
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1770

```
1770      d1r=(y(i+1)-y(i)) 1771
1771      xxd1v=1._dp/(x(i+1)-x 1772
1772      sig=(x(i)-x(i-1)) 1773
1773      xp=1._dp/(sig*d2(i)) 1774
1774      d2(i)=(sig-1._dp)*xp 1775
1775      u(i)=(6._dp*(d1r-d1l) 1776
1776      end do 1777
1777
1778      d2(n)=0._dp 1778
1779      do i=n-1,1,-1 1779
1780      d2(i)=d2(i)*d2(i+1)+u 1780
1781      end do 1781
1782
1783      deallocate(u) 1782
1784      end subroutine spline 1783
1785
1786      subroutine spline_double 1784
1787      integer, intent(in) :: 1787
1788      integer, parameter :: d 1788
1789      real(dp), intent(in) :: 1789
1790      real(dp), intent(out) : 1790
1791      real(dp), dimension(:), 1791
1792      integer i 1792
1793      real(dp) xp,sig,xxdiv 1793
1794
1795      allocate(u(1:n-1)) 1794
1796
1797      d2(1)=0._dp 1795
1798      u(1)=0._dp 1796
1799
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1801

```
1801      d1r= (y(2)-y(1))/(x(2)-  
1802      do i=2,n-1  
1803          d1l=d1r  
1804          d1r=(y(i+1)-y(i))/(x(  
1805              xxd=1._dp/(x(i+1)-x  
1806              sig=(x(i)-x(i-1))*xxd  
1807              xp=1._dp/(sig*d2(i-1))  
1808              d2(i)=(sig-1._dp)*xp  
1809              u(i)=(6._dp*(d1r-d1l))  
1810      end do  
1811  
1812      d2(n)=0._dp  
1813      do i=n-1,1,-1  
1814          d2(i)=d2(i)*d2(i+1)+u  
1815      end do  
1816  
1817      deallocate(u)  
1818  end subroutine spline_dou  
1819  
1820  
1821      function DLGAMMA(x)  
1822          !Use Stirling generali  
1823          !See e.g. http://en.wi  
1824          !Is accurate to at lea  
1825          double precision :: x  
1826          double precision:: DLG  
1827          double precision, para  
1828  
1829          if (x<32.d0) then  
1830              DLGAMMA = log(GAMMA(x
```

/Users/lp1lopa/Compare/camb\_des/utils.F9  
0, Top line: 1800

```
1800      d1r= (y(2)-y(1))/(x(2)-  
1801  
1802      do i=2,n-1  
1803          d1l=d1r  
1804          d1r=(y(i+1)-y(i))  
1805          xxd=1._dp/(x(i+1)-x  
1806          sig=(x(i)-x(i-1))  
1807          xp=1._dp/(sig*d2(i-1))  
1808          d2(i)=(sig-1._dp)*xp  
1809          u(i)=(6._dp*(d1r-d1l))  
1810  
1811      d2(n)=0._dp  
1812      do i=n-1,1,-1  
1813          d2(i)=d2(i)*d2(i+1)+u  
1814      end do  
1815  
1816      deallocate(u)  
1817  end subroutine spline  
1818  
1819  
1820  
1821      function DLGAMMA(x)  
1822          !Use Stirling general  
1823          !See e.g. http://en.w  
1824          !Is accurate to at le  
1825          double precision :: x  
1826          double precision:: DLG  
1827          double precision, par  
1828  
1829          if (x<32.d0) then  
1830              DLGAMMA = log(GAM
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1831

```
1831      else
1832          DLGAMMA = (x-0.5d0)*1
1833              1/12.d0/(1+x)*(1+1/
1834      end if
1835      end function DLGAMMA
1836
1837
1838      function LogGamma(x)
1839          real LogGamma
1840          real, intent(in) :: x
1841          integer i, j
1842          real r
1843
1844          i = nint(x*2)
1845          if (abs(i-x*2) > 1e-4
1846          if (mod(i,2) == 0) th
1847              r=0
1848              do j = 2, i/2-1
1849                  r = r + log(re
1850              end do
1851              LogGamma = r
1852          else
1853              r = log(pi)/2
1854              do j = 1, i-2 , 2
1855                  r = r+ log(j/2.0
1856              end do
1857              LogGamma = r
1858          end if
1859
1860      end function LogGamma
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1830

```
1830      else
1831          DLGAMMA = (x-0.5d
1832              1/12.d0/(1+x)
1833      end if
1834      end function DLGAMMA
1835
1836
1837      function LogGamma(x)
1838          real LogGamma
1839          real, intent(in) :: x
1840          integer i, j
1841          real r
1842
1843          i = nint(x*2)
1844          if (abs(i-x*2) > 1e-4
1845          if (mod(i,2) == 0) th
1846              r=0
1847              do j = 2, i/2-1
1848                  r = r + log(r
1849              end do
1850              LogGamma = r
1851
1852          else
1853              r = log(pi)/2
1854              do j = 1, i-2 , 2
1855                  r = r+ log(j/
1856              end do
1857              LogGamma = r
1858          end if
1859
1860      end function LogGamma
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1861

```
1861
1862      DOUBLE PRECISION FUNCTION
1863      !-----
1864      !
1865      ! This routine calculates the
1866      ! Computation is based on a
1867      ! The program uses rational
1868      ! function to at least 20 s
1869      ! for the approximation ove
1870      ! Those for the approximati
1871      ! The accuracy achieved dep
1872      ! compiler, the intrinsic f
1873      ! machine-dependent constan
1874      ! ****
1875      !
1876      ! Explanation of machine-depe
1877      !
1878      ! beta   - radix for the floa
1879      ! maxexp - the smallest posit
1880      ! XBIG   - the largest argume
1881      !           in the machine, i.
1882      !           GAMMA(XBIG)
1883      ! XINF   - the largest machin
1884      !           approximately beta
1885      ! EPS    - the smallest posit
1886      !           1.0+EPS .GT. 1.0
1887      ! XMININ - the smallest posit
1888      !           1/XMININ is machin
1889      !
1890      ! Approximate values for
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1860

```
1860
1861
1862      DOUBLE PRECISION FUNC
1863      !-----
1864      !
1865      ! This routine calcul
1866      ! Computation is ba
1867      ! The program uses
1868      ! function to at le
1869      ! for the approxima
1870      ! Those for the app
1871      ! The accuracy achi
1872      ! compiler, the int
1873      ! machine-dependent
1874      !
1875      !
1876      !
1877      ! beta   - radix for
1878      ! maxexp - the smalle
1879      ! XBIG   - the larges
1880      !           in the mac
1881      !           GA
1882      ! XINF   - the larges
1883      !           approximat
1884      ! EPS    - the smalle
1885      !           1.0+EPS .G
1886      ! XMININ - the smalle
1887      !           1/XMININ i
1888      !
1889      !
1890      ! Approximate val
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1891

1891 !  
1892 !  
1893 !  
1894 ! CRAY-1 (S.P.)  
1895 ! Cyber 180/855  
1896 ! under NOS (S.P.)  
1897 ! IEEE (IBM/XT,  
1898 ! SUN, etc.) (S.P.)  
1899 ! IEEE (IBM/XT,  
1900 ! SUN, etc.) (D.P.)  
1901 ! IBM 3033 (D.P.)  
1902 ! VAX D-Format (D.P.)  
1903 ! VAX G-Format (D.P.)  
1904 !  
1905 !  
1906 !  
1907 ! CRAY-1 (S.P.) 5.4  
1908 ! Cyber 180/855  
1909 ! under NOS (S.P.) 1.2  
1910 ! IEEE (IBM/XT,  
1911 ! SUN, etc.) (S.P.) 3.4  
1912 ! IEEE (IBM/XT,  
1913 ! SUN, etc.) (D.P.) 1.7  
1914 ! IBM 3033 (D.P.) 7.2  
1915 ! VAX D-Format (D.P.) 1.7  
1916 ! VAX G-Format (D.P.) 8.9  
1917 !  
1918 !\*\*\*\*\*  
1919 !\*\*\*\*\*  
1920 !

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1890

1890 !  
1891 !  
1892 !  
1893 ! CRAY-1 (S.P)  
1894 ! Cyber 180/855  
1895 ! under NOS (S.P)  
1896 ! IEEE (IBM/XT,  
1897 ! SUN, etc.) (S.P)  
1898 ! IEEE (IBM/XT,  
1899 ! SUN, etc.) (D.P)  
1900 ! IBM 3033 (D.P)  
1901 ! VAX D-Format (D.P)  
1902 ! VAX G-Format (D.P)  
1903 !  
1904 !  
1905 !  
1906 ! CRAY-1 (S.P)  
1907 ! Cyber 180/855  
1908 ! under NOS (S.P)  
1909 ! IEEE (IBM/XT,  
1910 ! SUN, etc.) (S.P)  
1911 ! IEEE (IBM/XT,  
1912 ! SUN, etc.) (D.P)  
1913 ! IBM 3033 (D.P)  
1914 ! VAX D-Format (D.P)  
1915 ! VAX G-Format (D.P)  
1916 !  
1917 !\*\*\*\*\*  
1918 !\*\*\*\*\*  
1919 !

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1921

```
1921 ! Error returns
1922 !
1923 ! The program returns the va
1924 ! when overflow would occ
1925 ! to be free of underflow
1926 !
1927 !
1928 ! Intrinsic functions requir
1929 !
1930 ! INT, DBLE, EXP, LOG, RE
1931 !
1932 !
1933 ! References: "An Overview of
1934 ! Functions", W.
1935 ! 506, Numerical
1936 ! (ed.), Springer
1937 !
1938 ! Computer Appro
1939 ! sons, New York
1940 !
1941 ! Latest modification: Octob
1942 !
1943 ! Authors: W. J. Cody and L.
1944 ! Applied Mathemati
1945 ! Argonne National
1946 ! Argonne, IL 60439
1947 !
1948 !-----
1949 !----- INTEGER I,N
1950 !----- LOGICAL PARITY
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1920

```
1920 !
1921 !
1922 !
1923 ! The program return
1924 ! when overflow w
1925 !
1926 !
1927 ! Intrinsic function
1928 !
1929 !
1930 !
1931 !
1932 ! References: "An Ove
1933 ! Functi
1934 ! 506, N
1935 ! (ed.),
1936 !
1937 !
1938 !
1939 !
1940 !
1941 !
1942 ! Latest modificatio
1943 !
1944 !
1945 !
1946 !
1947 !
1948 !
1949 !----- INTEGER I,N
1950 !----- LOGICAL PARITY
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1951

```
1951      DOUBLE PRECISION C,EPS,FA
1952          TWO,X,XBIG,XDEN,XINF
1953          DIMENSION C(7),P(8),Q(8
1954 !-----
1955 !  Mathematical constants
1956 !-----
1957     DATA ONE,HALF,TWELVE,TWO,
1958          SQRTPI/0.918938533204
1959          PI/3.1415926535897932
1960 !-----
1961 !  Machine dependent parameter
1962 !-----
1963     DATA XBIG,XMININ,EPS/35.0
1964          XINF/3.4E38/
1965 !-----
1966 !  Numerator and denominator
1967 !  approximation over (1,2
1968 !-----
1969     DATA P/-1.716185138865494
1970          -3.7980425647094563
1971          8.66966202790413211
1972          -3.6144413418691172
1973     DATA Q/-3.084023001197389
1974          -1.01515636749021914
1975          2.2538118420980151
1976          -1.34659959864969306
1977 !-----
1978 !  Coefficients for minimax ap
1979 !-----
1980     DATA C/-1.910444077728D-0
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1950

```
1950      DOUBLE PRECISION C,EP
1951          TWO,X,XBIG,XDEN,X
1952          DIMENSION C(7),P(8),Q
1953 !-----
1954 !  Mathematical const
1955 !-----
1956     DATA ONE,HALF,TWELVE,
1957          SQRTPI/0.91893853
1958          PI/3.141592653589
1959 !-----
1960 !  Machine dependent
1961 !-----
1962     DATA XBIG,XMININ,EPS/
1963          XINF/3.4E38/
1964 !-----
1965 !  Numerator and deno
1966 !  approximation o
1967 !-----
1968     DATA P/-1.71618513886
1969          -3.79804256470945
1970          8.669662027904132
1971          -3.61444134186911
1972     DATA Q/-3.08402300119
1973          -1.01515636749021
1974          2.253811842098015
1975          -1.34659959864969
1976 !-----
1977 !  Coefficients for mi
1978 !-----
1979     DATA C/-1.91044407772
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 1981

```
1981          -5.952379913043012D-0
1982          -2.7777777777777681622
1983          5.7083835261D-03/
1984 !-----
1985 ! Statement functions for co
1986 !-----
1987     PARITY = .FALSE.
1988     FACT = ONE
1989     N = 0
1990     Y = X
1991     IF (Y .LE. ZERO) THEN
1992 !-----
1993 ! Argument is negative
1994 !-----
1995     Y = -X
1996     Y1 = AINT(Y)
1997     RES = Y - Y1
1998     IF (RES .NE. ZERO
1999         IF (Y1 .NE.
2000             FACT = -PI
2001             Y = Y + ONE
2002         ELSE
2003             RES = XINF
2004             GO TO 900
2005         END IF
2006     END IF
2007 !-----
2008 ! Argument is positive
2009 !-----
2010    IF (Y .LT. EPS) THEN
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 1980

```
1980          -5.95237991304301
1981          -2.777777777777768
1982          5.7083835261D-03/
1983 !-----
1984 ! Statement function
1985 !
1986     PARITY = .FALSE.
1987     FACT = ONE
1988     N = 0
1989     Y = X
1990     IF (Y .LE. ZERO) THEN
1991 !-----
1992 ! Argument is ne
1993 !
1994     Y = -X
1995     Y1 = AINT(Y)
1996     RES = Y - Y1
1997     IF (RES .NE. ZERO
1998         IF (Y1 .NE. A
1999             FACT = -PI /
2000             Y = Y + ONE
2001         ELSE
2002             RES = XINF
2003             GO TO 900
2004         END IF
2005     END IF
2006 !-----
2007 ! Argument is positi
2008 !
2009     IF (Y .LT. EPS) THEN
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2011

```
2011 !-----  
2012 ! Argument .LT. EPS  
2013 !-----  
2014      IF (Y .GE. XMININ  
2015          RES = ONE /  
2016      ELSE  
2017          RES = XINF  
2018          GO TO 900  
2019      END IF  
2020      ELSE IF (Y .LT. TWEL  
2021          Y1 = Y  
2022          IF (Y .LT. ONE) T  
2023      !-----  
2024      ! 0.0 .LT. argument .LT. 1.0  
2025      !-----  
2026          Z = Y  
2027          Y = Y + ONE  
2028      ELSE  
2029      !-----  
2030      ! 1.0 .LT. argument .LT. 12.  
2031      !-----  
2032          N = INT(Y)  
2033          Y = Y - REA  
2034          Z = Y - ONE  
2035      END IF  
2036      !-----  
2037      ! Evaluate approximation for  
2038      !-----  
2039          XNUM = ZERO  
2040          XDEN = ONE
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2010

```
2010 !-----  
2011 ! Argument .LT.  
2012 !-----  
2013      IF (Y .GE. XMININ  
2014          RES = ONE / Y  
2015      ELSE  
2016          RES = XINF  
2017          GO TO 900  
2018      END IF  
2019      ELSE IF (Y .LT. TWELV  
2020          Y1 = Y  
2021          IF (Y .LT. ONE) T  
2022      !-----  
2023      ! 0.0 .LT. argument .LT. 1.0  
2024      !-----  
2025          Z = Y  
2026          Y = Y + ONE  
2027      ELSE  
2028      !-----  
2029      ! 1.0 .LT. argument .LT. 12.  
2030      !-----  
2031          N = INT(Y) -  
2032          Y = Y - REAL(  
2033          Z = Y - ONE  
2034      END IF  
2035      !-----  
2036      ! Evaluate approx  
2037      !-----  
2038          XNUM = ZERO  
2039          XDEN = ONE
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2041

```
2041          DO 260 I = 1, 8
2042          XNUM = (XNUM +
2043          XDEN = XDEN *
2044 260      CONTINUE
2045          RES = XNUM / XDEN
2046          IF (Y1 .LT. Y) TH
2047 !-----
2048 !   Adjust result for case 0.
2049 !
2050          RES = RES /
2051          ELSE IF (Y1 .G
2052 !
2053 !   Adjust result for case 2.
2054 !
2055          DO 290 I =
2056          RES = RE
2057          Y = Y +
2058 290      CONTINUE
2059      END IF
2060      ELSE
2061 !
2062 !   Evaluate for argument .GE.
2063 !
2064          IF (Y .LE. XBIG)
2065          YSQ = Y * Y
2066          SUM = C(7)
2067          DO 350 I =
2068          SUM = SU
2069 350      CONTINUE
2070          SUM = SUM/Y
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2040

```
2040          DO 260 I = 1, 8
2041          XNUM = (XNUM +
2042          XDEN = XDEN *
2043 260      CONTINUE
2044          RES = XNUM / XDEN
2045          IF (Y1 .LT. Y) TH
2046 !-----
2047 !   Adjust res
2048 !
2049          RES = RES / Y
2050          ELSE IF (Y1 .GT.
2051 !
2052 !   Adjust res
2053 !
2054          DO 290 I = 1,
2055          RES = RES
2056          Y = Y + O
2057 290      CONTINUE
2058      END IF
2059      ELSE
2060 !
2061 !   Evaluate for a
2062 !
2063          IF (Y .LE. XBIG)
2064          YSQ = Y * Y
2065          SUM = C(7)
2066          DO 350 I =
2067          SUM = SU
2068 350      CONTINUE
2069          SUM = SUM/Y -
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2071

```
2071          SUM = SUM +  
2072          RES = EXP(S  
2073      ELSE  
2074          RES = XINF  
2075          GO TO 900  
2076      END IF  
2077      END IF  
2078 !-----  
2079 ! Final adjustments and retu  
2080 !-----  
2081      IF (PARITY) RES = -RES  
2082      IF (FACT .NE. ONE) RES  
2083 900 GAMMA = RES  
2084  
2085      END FUNCTION GAMMA  
2086  
2087 subroutine SetIdlePriority  
2088 #ifdef RUNIDLE  
2089     USE DFWIN  
2090     Integer dwPriority  
2091     Integer CheckPriority  
2092  
2093     dwPriority = 64 ! idle pr  
2094     CheckPriority = SetPriori  
2095 #endif  
2096 end subroutine SetIdlePrior  
2097  
2098 subroutine GetThreeJs(thr  
2099 !Recursive evaluation o
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2070

```
2070          SUM = SUM + (S  
2071          RES = EXP(SUM  
2072      ELSE  
2073          RES = XINF  
2074          GO TO 900  
2075      END IF  
2076      END IF  
2077 !-----  
2078 ! Final adjustments  
2079 !-----  
2080      IF (PARITY) RES = -RE  
2081      IF (FACT .NE. ONE) RE  
2082 900 GAMMA = RES  
2083  
2084      END FUNCTION GAMMA  
2085  
2086 subroutine SetIdlePri  
2087 #ifdef RUNIDLE  
2088     USE DFWIN  
2089     Integer dwPriority  
2090     Integer CheckPriority  
2091  
2092     dwPriority = 64 ! idl  
2093     CheckPriority = SetPr  
2094 #endif  
2095 end subroutine SetIdl  
2096  
2097  
2098 subroutine GetThreeJs(thr  
2099 !Recursive evaluation o
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2101

```
2101      implicit none
2102      integer, parameter :: d
2103      integer, intent(in) :::
2104      real(dl), dimension(*) :::
2105      INTEGER, PARAMETER :: i
2106      integer(i8) :: 12,13,m2
2107      integer(i8) :: 11, m1,
2108
2109      real(dl) :: newfac, old
2110      real(dl) :: x1,x2,x3, y
2111      integer i,ier, index, n
2112      integer nfinp1,nfinp2,n
2113      real(dl), parameter :::
2114      real(dl), parameter :::
2115
2116      ! routine to generate set
2117
2118      ! by recursion from l1min
2119      !           to l1max
2120      ! the resulting 3j-coeffs
2121
2122      ! to achieve the numerica
2123      ! simultaneously forwards
2124      ! respectively.
2125
2126      ! lmatch is the l1-value
2127
2128      ! ndim is the length of t
2129
2130      ! ier = -1 for all 3j van
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2100

```
2100      implicit none
2101      integer, parameter :::
2102      integer, intent(in) :::
2103      real(dl), dimension(*) :::
2104      INTEGER, PARAMETER :: i
2105      integer(i8) :: 12,13,
2106      integer(i8) :: 11, m1
2107
2108      real(dl) :: newfac, o
2109      real(dl) :: x1,x2,x3,
2110      integer i,ier, index,
2111      integer nfinp1,nfinp2
2112      real(dl), parameter :::
2113      real(dl), parameter :::
2114
2115      ! routine to generate
2116
2117      ! by recursion from l1
2118      !           to l1
2119      ! the resulting 3j-co
2120
2121      ! to achieve the nume
2122      ! simultaneously forw
2123      ! respectively.
2124
2125      ! lmatch is the l1-va
2126
2127      ! ndim is the length
2128
2129      ! ier = -1 for all 3j
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2131

```
2131      ! ier = -2 if possible 3j
2132      ! ier >= 0 otherwise
2133
2134      l2=l2in
2135      l3=l3in
2136      m2=m2in
2137      m3=m3in
2138      newfac = 0
2139      lmatch = 0
2140      m1 = -(m2+m3)
2141
2142      ! check relative magnitud
2143      ier = 0
2144
2145      if (l2 < abs(m2) .or. l
2146      ier = -1
2147      call MPIStop('error ier'
2148      return
2149      end if
2150
2151      ! limits for ll
2152      llmin = max(abs(l2-l3),
2153      llmax = l2+l3
2154
2155      if (llmin >= llmax) the
2156      if (llmin/=llmax) then
2157      ier = -1
2158      call MPIStop('error i
2159      return
2160      end if
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2130

```
2130      ! ier = -2 if possibl
2131      ! ier >= 0 otherwise
2132
2133      l2=l2in
2134      l3=l3in
2135      m2=m2in
2136      m3=m3in
2137      newfac = 0
2138      lmatch = 0
2139      m1 = -(m2+m3)
2140
2141      ! check relative magn
2142      ier = 0
2143
2144      if (l2 < abs(m2) .or.
2145      ier = -1
2146      call MPIStop('err
2147      return
2148      end if
2149
2150      ! limits for ll
2151      llmin = max(abs(l2-l3
2152      llmax = l2+l3
2153
2154      if (llmin >= llmax) t
2155      if (llmin/=llmax)
2156      ier = -1
2157      call MPIStop(
2158      return
2159      end if
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2161

```
2161
2162      ! reached if l1 can take
2163      thrcof(1) = (-1)**abs(l
2164      return
2165
2166      end if
2167
2168      nfin = l1max-l1min+1
2169
2170      ! starting forward recurs
2171      l1 = l1min
2172      thrcof(1) = srtiny
2173      sum1 = (2*l1 + 1)*tiny
2174
2175      lstep = 1
2176
2177      30      lstep = lstep+1
2178      l1 = l1+1
2179
2180      oldfac = newfac
2181      a1 = (l1+l2+l3+1)*(l1-1
2182      a2 = (l1+m1)*(l1-m1)*(-
2183      newfac = sqrt(a2*real(a
2184      if (l1 == 1) then
2185          !IF L1 = 1 (L1-1) H
2186          c1 = -(2*l1-1)*l1*(m
2187      else
2188
2189          dv = -12*(l2+1)*m1 + 1
2190          denom = (l1-1)*newfac
```

/Users/lp1lopa/Compare/camb\_des/utils.F9
0, Top line: 2160

```
2160
2161      ! reached if l1 c
2162      thrcof(1) = (-1)*
2163      return
2164
2165      end if
2166
2167      nfin = l1max-l1min+1
2168
2169      ! starting forward re
2170      l1 = l1min
2171      thrcof(1) = srtiny
2172      sum1 = (2*l1 + 1)*tin
2173
2174      lstep = 1
2175
2176      30      lstep = lstep+1
2177      l1 = l1+1
2178
2179      oldfac = newfac
2180      a1 = (l1+l2+l3+1)*(l1
2181      a2 = (l1+m1)*(l1-m1)*
2182      newfac = sqrt(a2*real(a
2183      if (l1 == 1) then
2184          !IF L1 = 1 (L1-1)
2185          c1 = -(2*l1-1)*l1
2186      else
2187
2188          dv = -12*(l2+1)*m
2189          denom = (l1-1)*ne
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2191

```
2191
2192      if (lstep > 2) c1old =
2193      c1 = -(2*11-1)*dv/deno
2194
2195      end if
2196
2197      if (lstep<= 2) then
2198
2199      ! if l1=l1min+1 the third
2200      x = srtiny*c1
2201      thrcof(2) = x
2202      sum1 = sum1+tiny*(2*11
2203      if(lstep==nfin) then
2204          sumuni=sum1
2205          go to 230
2206      end if
2207      goto 30
2208
2209      end if
2210
2211      c2 = -11*oldfac/denom
2212
2213      ! recursion to the next 3
2214      x = c1*thrcof(lstep-1)
2215      thrcof(lstep) = x
2216      sumfor = sum1
2217      sum1 = sum1 + (2*11+1)*
2218      if (lstep/=nfin) then
2219
2220      ! see if last unnormalise
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2190

```
2190
2191      if (lstep > 2) c1
2192      c1 = -(2*11-1)*dv
2193
2194      end if
2195
2196      if (lstep<= 2) then
2197
2198      ! if l1=l1min+1 t
2199      x = srtiny*c1
2200      thrcof(2) = x
2201      sum1 = sum1+tiny*
2202      if(lstep==nfin) t
2203          sumuni=sum1
2204          go to 230
2205      end if
2206      goto 30
2207
2208      end if
2209
2210      c2 = -11*oldfac/denom
2211
2212      ! recursion to the ne
2213      x = c1*thrcof(lstep-1)
2214      thrcof(lstep) = x
2215      sumfor = sum1
2216      sum1 = sum1 + (2*11+1)*
2217      if (lstep/=nfin) then
2218
2219      ! see if last unn
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2221

```
2221      if (abs(x) >= srhuge) t
2222      ! REACHED IF LAST 3J
2223      ! SO THAT THE RECURS
2224      ! HAS TO BE RESCALED
2225
2226      ier = ier+1
2227      do i = 1, lstep
2228          if (abs(thrcof(i))
2229              thrcof(i) = thrco
2230      end do
2231
2232      sum1 = sum1/huge
2233      sumfor = sumfor/huge
2234      x = x/srhuge
2235
2236
2237      end if
2238
2239      ! as long as abs(c1) is d
2240      ! 3j-valuse and so is num
2241      ! detected, the recursion
2242
2243      if (c1old > abs(c1)) got
2244
2245      end if !lstep/=nfin
2246
2247      ! keep three 3j-coeffs ar
2248
2249      lmatch = 11-1
2250      x1 = x
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2220

```
2220      if (abs(x) >= srh
2221      ! REACHED IF
2222      ! SO THAT THE
2223      ! HAS TO BE R
2224
2225
2226      ier = ier+1
2227      do i = 1, lst
2228          if (abs(t
2229              thrcof(i)
2230      end do
2231
2232      sum1 = sum1/h
2233      sumfor = sumf
2234      x = x/srhuge
2235
2236
2237      end if
2238
2239      ! as long as abs(
2240      ! 3j-valuse and s
2241      ! detected, the r
2242
2243
2244      if (c1old > abs(c
2245
2246      end if !lstep/=nfin
2247
2248      ! keep three 3j-coeff
2249      lmatch = 11-1
2250      x1 = x
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2251

```
2251      x2 = thrcof(lstep-1)
2252      x3 = thrcof(lstep-2)
2253      nstep2 = nfin-lstep+3
2254
2255      ! -----
2256      !
2257      ! starting backward recur
2258      ! forward and backward re
2259      ! l1 = lmatch-1, lmatch,
2260
2261      nfinp1 = nfin+1
2262      nfinp2 = nfin+2
2263      nfinp3 = nfin+3
2264      l1 = l1max
2265      thrcof(nfin) = srtiny
2266      sum2 = tiny*(2*l1+1)
2267
2268      l1 = l1+2
2269      lstep=1
2270
2271      do
2272      lstep = lstep + 1
2273      l1= l1-1
2274
2275      oldfac = newfac
2276      a1 = (l1+l2+l3)*(l1-l2+
2277      a2 = (l1+m1-1)*(l1-m1-1
2278      newfac = sqrt(a1*real(a
2279
2280      dv = -12*(l2+1)*m1 + 13
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2250

```
2250      x2 = thrcof(lstep-1)
2251      x3 = thrcof(lstep-2)
2252      nstep2 = nfin-lstep+3
2253
2254      ! -----
2255      !
2256      ! starting backward r
2257      ! forward and backwar
2258      ! l1 = lmatch-1, lmat
2259
2260      nfinp1 = nfin+1
2261      nfinp2 = nfin+2
2262      nfinp3 = nfin+3
2263      l1 = l1max
2264      thrcof(nfin) = srtiny
2265      sum2 = tiny*(2*l1+1)
2266
2267      l1 = l1+2
2268      lstep=1
2269
2270      do
2271      lstep = lstep + 1
2272      l1= l1-1
2273
2274      oldfac = newfac
2275      a1 = (l1+l2+l3)*(l1-l2+
2276      a2 = (l1+m1-1)*(l1-m1-1
2277      newfac = sqrt(a1*real(a
2278
2279      dv = -12*(l2+1)*m1
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2281

```
2281
2282     denom = 11*newfac
2283     c1 = -(2*11-1)*dv/denom
2284     if (lstep <= 2) then
2285
2286         ! if l2=l2max+1, the
2287
2288         y = srtiny*c1
2289         thrcof(nfin-1) = y
2290         sumbac = sum2
2291         sum2 = sum2 + tiny*(  

2292
2293         cycle
2294
2295     end if
2296
2297     c2 = -(11-1)*oldfac/den
2298
2299     ! recursion to the next 3
2300     y = c1*thrcof(nfinp2-ls
2301
2302     if (lstep==nstep2) exit
2303
2304     thrcof(nfinp1-lstep) =
2305     sumbac = sum2
2306     sum2 = sum2+(2*11-3)*y*
2307
2308     ! see if last unnormalise
2309     if (abs(y) >= srhuge) t
2310
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2280

```
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2311

```
2311          ! reached if 3j-coef
2312          ! thrcof(nfin),...
2313
2314          ier=ier+1
2315          do i = 1, lstep
2316              index=nfin-i+1
2317              if (abs(thrcof(in
2318                  thrcof(index) = t
2319          end do
2320
2321          sum2=sum2/huge
2322          sumbac=sumbac/huge
2323
2324      end if
2325
2326      end do
2327
2328      ! the forward recursion 3
2329      ! corresponding backward
2330
2331          y3 = y
2332          y2 = thrcof(nfinp2-lste
2333          y1 = thrcof(nfinp3-lste
2334
2335          ! determine now ratio suc
2336
2337          ratio = (x1*y1+x2*y2+x3
2338          nlim = nfin-nstep2+1
2339
2340          if (abs(ratio) >= 1) th
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2310

```
2310          ! reached if
2311          ! thrcof(nfin
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340          if (abs(ratio) >= 1)
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2341

```
2341
2342         thrcof(1:nlim) = ratio
2343         sumuni = ratio*ratio*s
2344
2345     else
2346
2347         nlim = nlim+1
2348         ratio = 1/ratio
2349         do n = nlim, nfin
2350             thrcof(n) = ratio*th
2351         end do
2352         sumuni = sumfor + ratio
2353
2354     end if
2355     ! normalise 3j-coeffs
2356
2357 cnorm = 1/sqrt(sumuni)
2358
2359     ! sign convention for las
2360
2361         sign1 = sign(one,thrcof
2362         sign2 = (-1)**(abs(l2+m
2363         if (sign1*sign2 <= 0) t
2364             cnorm = -cnorm
2365         end if
2366         if (abs(cnorm) >= one)
2367             thrcof(1:nfin) = cno
2368             return
2369         end if
2370
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2340

```
2340
2341         thrcof(1:nlim) =
2342         sumuni = ratio*ra
2343
2344     else
2345
2346         nlim = nlim+1
2347         ratio = 1/ratio
2348         do n = nlim, nfin
2349             thrcof(n) = r
2350         end do
2351         sumuni = sumfor +
2352
2353     end if
2354     ! normalise 3j-coeffs
2355
2356 cnorm = 1/sqrt(sumuni)
2357
2358     ! sign convention for
2359
2360         sign1 = sign(one,thrcof
2361         sign2 = (-1)**(abs(l2+m
2362         if (sign1*sign2 <= 0)
2363             cnorm = -cnorm
2364         end if
2365         if (abs(cnorm) >= one)
2366             thrcof(1:nfin) =
2367             return
2368         end if
2369
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2371

```
2371      thresh = tiny/abs(cnorm
2372
2373      do n = 1, nfin
2374          if (abs(thrcof(n)) <
2375              thrcof(n) = cnorm*th
2376      end do
2377      return
2378
2379  end subroutine GetThreeJs
2380
2381
2382
2383  end module AMLutils
2384
2385
2386 #ifdef ZIGGURAT
2387 MODULE Ziggurat
2388 ! Marsaglia & Tsang generator
2389 ! Translated from C by Alan M
2390
2391 ! Marsaglia, G. & Tsang, W.W.
2392 ! random variables', J. Stati
2393
2394 ! This is an electronic journ
2395 ! http://www.jstatsoft.org/v0
2396
2397 ! N.B. It is assumed that all
2398 ! N.B. The value of M2 has be
2399 ! unsigned integers in F
2400
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2370

```
2370      thresh = tiny/abs(cno
2371
2372      do n = 1, nfin
2373          if (abs(thrcof(n))
2374              thrcof(n) = cnorm
2375      end do
2376      return
2377
2378  end subroutine GetThr
2379
2380
2381
2382
2383
2384
2385
2386 #ifdef ZIGGURAT
2387 MODULE Ziggurat
2388 ! Marsaglia & Tsang gener
2389 ! Translated from C by Al
2390
2391 ! Marsaglia, G. & Tsang,
2392 ! random variables', J. S
2393
2394 ! This is an electronic j
2395 ! http://www.jstatsoft.or
2396
2397 ! N.B. It is assumed that
2398 ! N.B. The value of M2 ha
2399 ! unsigned integers
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2401

```
2401 ! Latest version - 1 January
2402 !
2403 ! AL: useful material at http
2404     IMPLICIT NONE
2405
2406 PRIVATE
2407
2408 INTEGER,  PARAMETER :: D
2409 REAL(DP), PARAMETER :: m
2410             h
2411 REAL(DP)           :: d
2412             v
2413             q
2414             t
2415             v
2416 INTEGER,  SAVE    :: i
2417             k
2418 REAL(DP), SAVE    :: w
2419 LOGICAL,  SAVE    :: i
2420
2421 PUBLIC :: zigset, shr3, u
2422
2423
2424 CONTAINS
2425
2426
2427 SUBROUTINE zigset( jsrseed )
2428
2429     INTEGER, INTENT(IN) :: js
2430
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2400

```
2400 ! Latest version - 1 Janu
2401 !
2402 !
2403 ! AL: useful material at
2404     IMPLICIT NONE
2405
2406 PRIVATE
2407
2408 INTEGER,  PARAMETER :: :
2409 REAL(DP), PARAMETER :: :
2410             half=0.5_DP
2411 REAL(DP)           :: :
2412             vn=0.0099125630352621
2413             q,
2414             te=7.697117470131487_
2415             ve=0.0039496598225815
2416 INTEGER,  SAVE    :: :
2417             ke(0:255), hz
2418 REAL(DP), SAVE    :: :
2419 LOGICAL,  SAVE    :: :
2420
2421
2422
2423
2424 CONTAINS
2425
2426
2427 SUBROUTINE zigset( jsrsee
2428
2429     INTEGER, INTENT(IN) :: j
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2431

```
2431      INTEGER :: i
2432
2433      ! Set the seed
2434      jsr = jsrseed
2435
2436      ! Tables for RNOR
2437      q = vn*EXP(half*dn*dn)
2438      kn(0) = (dn/q)*m1
2439      kn(1) = 0
2440      wn(0) = q/m1
2441      wn(127) = dn/m1
2442      fn(0) = 1.0_DP
2443      fn(127) = EXP( -half*dn*dn)
2444      DO i = 126, 1, -1
2445          dn = SQRT( -2.0_DP * LO
2446          kn(i+1) = (dn/tn)*m1
2447          tn = dn
2448          fn(i) = EXP(-half*dn*dn)
2449          wn(i) = dn/m1
2450      END DO
2451
2452      ! Tables for REXP
2453      q = ve*EXP( de )
2454      ke(0) = (de/q)*m2
2455      ke(1) = 0
2456      we(0) = q/m2
2457      we(255) = de/m2
2458      fe(0) = 1.0_DP
2459      fe(255) = EXP( -de )
2460      DO i = 254, 1, -1
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2430

```
2430      INTEGER :: i
2431
2432      ! Set the seed
2433      jsr = jsrseed
2434
2435      ! Tables for RNOR
2436      q = vn*EXP(half*dn*dn)
2437      kn(0) = (dn/q)*m1
2438      kn(1) = 0
2439      wn(0) = q/m1
2440      wn(127) = dn/m1
2441      fn(0) = 1.0_DP
2442      fn(127) = EXP( -half*dn*d
2443      DO i = 126, 1, -1
2444          dn = SQRT( -2.0_DP *
2445          kn(i+1) = (dn/tn)*m1
2446          tn = dn
2447          fn(i) = EXP(-half*dn*
2448          wn(i) = dn/m1
2449      END DO
2450
2451      ! Tables for REXP
2452      q = ve*EXP( de )
2453      ke(0) = (de/q)*m2
2454      ke(1) = 0
2455      we(0) = q/m2
2456      we(255) = de/m2
2457      fe(0) = 1.0_DP
2458      fe(255) = EXP( -de )
2459      DO i = 254, 1, -1
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2461

```
2461      de = -LOG( ve/de + EXP(   
2462      ke(i+1) = m2 * (de/te)  
2463      te = de  
2464      fe(i) = EXP( -de )  
2465      we(i) = de/m2  
2466      END DO  
2467      initialized = .TRUE.  
2468      RETURN  
2469      END SUBROUTINE zigset  
2470  
2471  
2472  
2473      ! Generate random 32-bit int  
FUNCTION shr3( ) RESULT( ival  
2475      INTEGER :: ival  
2476  
2477      jz = jsr  
2478      jsr = IEOR( jsr, ISHFT( js  
2479      jsr = IEOR( jsr, ISHFT( js  
2480      jsr = IEOR( jsr, ISHFT( js  
2481      ival = jz + jsr  
2482      RETURN  
2483      END FUNCTION shr3  
2484  
2485  
2486  
2487      ! Generate uniformly distrib  
FUNCTION uni( ) RESULT( fn_va  
2489      REAL(DP) :: fn_val  
2490
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2460

```
2460      de = -LOG( ve/de + EX  
2461      ke(i+1) = m2 * (de/te)  
2462      te = de  
2463      fe(i) = EXP( -de )  
2464      we(i) = de/m2  
2465      END DO  
2466      initialized = .TRUE.  
2467      RETURN  
2468      END SUBROUTINE zigset  
2469  
2470  
2471  
2472      ! Generate random 32-bit  
FUNCTION shr3( ) RESULT( ival  
2474      INTEGER :: ival  
2475  
2476      jz = jsr  
2477      jsr = IEOR( jsr, ISHFT( j  
2478      jsr = IEOR( jsr, ISHFT( j  
2479      jsr = IEOR( jsr, ISHFT( j  
2480      ival = jz + jsr  
2481      RETURN  
2482      END FUNCTION shr3  
2483  
2484  
2485  
2486  
2487      ! Generate uniformly dis  
FUNCTION uni( ) RESULT( f  
2489      REAL(DP) :: fn_val  
2490
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2491

```
2491 fn_val = half + 0.2328306e
2492 RETURN
2493 END FUNCTION uni
2494
2495
2496
2497 ! Generate random normals
2498 FUNCTION rnor( ) RESULT( fn_v
2499     REAL(DP) :: f
2500
2501     REAL(DP), PARAMETER :: r
2502     REAL(DP) :: x
2503
2504     IF( .NOT. initialized ) CA
2505         hz = shr3( )
2506         iz = IAND( hz, 127 )
2507         IF( ABS( hz ) < kn(iz) ) T
2508             fn_val = hz * wn(iz)
2509         ELSE
2510             DO
2511                 IF( iz == 0 ) THEN
2512                     DO
2513                         x = -0.2904764
2514                         y = -LOG( uni(
2515                             IF( y+y >= x*x
2516                                 END DO
2517                                 fn_val = r+x
2518                                 IF( hz <= 0 ) fn_
2519                                     RETURN
2520                                     END IF
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2490

```
2490 fn_val = half + 0.2328306
2491 RETURN
2492 END FUNCTION uni
2493
2494
2495
2496
2497 ! Generate random normal
2498 FUNCTION rnor( ) RESULT(
2499     REAL(DP) :: :
2500
2501     REAL(DP), PARAMETER :: :
2502     REAL(DP) :: :
2503
2504     IF( .NOT. initialized ) C
2505         hz = shr3( )
2506         iz = IAND( hz, 127 )
2507         IF( ABS( hz ) < kn(iz) )
2508             fn_val = hz * wn(iz)
2509         ELSE
2510             DO
2511                 IF( iz == 0 ) THE
2512                     DO
2513                         x = -0.29
2514                         y = -LOG(
2515                             IF( y+y >
2516                                 END DO
2517                                 fn_val = r+x
2518                                 IF( hz <= 0 )
2519                                     RETURN
2520                                     END IF
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2521

```
2521      x = hz * wn(iz)
2522      IF( fn(iz) + uni( )*
2523          fn_val = x
2524          RETURN
2525      END IF
2526      hz = shr3( )
2527      iz = IAND( hz, 127 )
2528      IF( ABS( hz ) < kn(i
2529          fn_val = hz * wn(
2530          RETURN
2531      END IF
2532      END DO
2533  END IF
2534  RETURN
2535 END FUNCTION rnor
2536
2537
2538
2539 ! Generate random exponentia
2540 FUNCTION rexp( ) RESULT( fn_v
2541     REAL(DP) :: fn_val
2542
2543     REAL(DP) :: x
2544
2545     IF( .NOT. initialized ) CA
2546         jz = shr3( )
2547         iz = IAND( jz, 255 )
2548         IF( ABS( jz ) < ke(iz) ) T
2549             fn_val = ABS(jz) * we(i
2550             RETURN
```

/Users/lp1lopa/Compare/camb\_des/utils.F9
0, Top line: 2520

```
2520      x = hz * wn(iz)
2521      IF( fn(iz) + uni( )
2522          fn_val = x
2523          RETURN
2524      END IF
2525      hz = shr3( )
2526      iz = IAND( hz, 12
2527      IF( ABS( hz ) < k
2528          fn_val = hz *
2529          RETURN
2530      END IF
2531      END DO
2532  END IF
2533  RETURN
2534 END FUNCTION rnor
2535
2536
2537
2538
2539 ! Generate random expone
2540 FUNCTION rexp( ) RESULT(
2541     REAL(DP) :: fn_val
2542
2543     REAL(DP) :: x
2544
2545     IF( .NOT. initialized ) C
2546         jz = shr3( )
2547         iz = IAND( jz, 255 )
2548         IF( ABS( jz ) < ke(iz) )
2549             fn_val = ABS(jz) * we
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2551

```
2551      END IF
2552      DO
2553          IF( iz == 0 ) THEN
2554              fn_val = 7.69711 - L
2555              RETURN
2556          END IF
2557          x = ABS( jz ) * we(iz)
2558          IF( fe(iz) + uni( )*(fe
2559              fn_val = x
2560              RETURN
2561          END IF
2562          jz = shr3( )
2563          iz = IAND( jz, 255 )
2564          IF( ABS( jz ) < ke(iz)
2565              fn_val = ABS( jz ) *
2566              RETURN
2567          END IF
2568      END DO
2569      RETURN
2570  END FUNCTION rexp
2571
2572  END MODULE ziggurat
2573 #endif
2574
2575
2576
2577  module Random
2578      integer :: rand_inst = 0
2579      logical, parameter :: use_zi
2580      integer, parameter :: krand
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2550

```
2550      END IF
2551      DO
2552          IF( iz == 0 ) THEN
2553              fn_val = 7.69711
2554              RETURN
2555          END IF
2556          x = ABS( jz ) * we(iz)
2557          IF( fe(iz) + uni( )*(fe
2558              fn_val = x
2559              RETURN
2560          END IF
2561          jz = shr3( )
2562          iz = IAND( jz, 255 )
2563          IF( ABS( jz ) < ke(iz)
2564              fn_val = ABS( jz )
2565              RETURN
2566          END IF
2567      END DO
2568      RETURN
2569  END FUNCTION rexp
2570
2571
2572
2573
2574
2575
2576
2577  module Random
2578      integer :: rand_inst = 0
2579      logical, parameter :: use_zi
2580      integer, parameter :: krand
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2581

```
2581 !Ziggurat is significantly
2582 !Have seem some suspicious
2583
2584 INTERFACE RandRotation
2585 MODULE PROCEDURE RandRotati
2586 END INTERFACE
2587
2588 contains
2589
2590 subroutine initRandom(i, i2
2591 use AMLUtils
2592 #ifdef ZIGGURAT
2593 use Ziggurat
2594 #endif
2595 implicit none
2596 integer, optional, intent(I
2597 integer, optional, intent(I
2598 integer seed_in,kl,ij
2599 character(len=10) :: fred
2600 real(krand) :: klr
2601
2602 if (present(i)) then
2603   seed_in = i
2604 else
2605   seed_in = -1
2606 end if
2607 if (seed_in /= -1) then
2608   if (present(i2)) then
2609     kl=i2
2610     if (i2 > 30081) call :
```

/Users/lp1lopa/Compare/camb\_des/utils.F9
0, Top line: 2580

```
2580 !Ziggurat is significatl
2581 !Have seem some suspiciou
2582
2583 INTERFACE RandRotation
2584 MODULE PROCEDURE RandRota
2585 END INTERFACE
2586
2587 contains
2588
2589 subroutine initRandom(i,
2590 use AMLUtils
2591 #ifdef ZIGGURAT
2592 use Ziggurat
2593 #endif
2594 implicit none
2595 integer, optional, intent
2596 integer, optional, intent
2597 integer seed_in,kl,ij
2598 character(len=10) :: fred
2599 real(krand) :: klr
2600
2601 if (present(i)) then
2602   seed_in = i
2603 else
2604   seed_in = -1
2605 end if
2606 if (seed_in /= -1) then
2607   if (present(i2)) then
2608     kl=i2
2609     if (i2 > 30081) c:
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2611

```
2611      else
2612          kl = 9373
2613      end if
2614      ij = i
2615      else
2616          call system_clock(coun
2617          ij = mod(ij + rand_ins
2618          call date_and_time(ti
2619          read (fred,'(e10.3)')
2620          kl = mod(int(klr*1000)
2621      end if
2622
2623      if (Feedback > 0 ) writ
2624          call rmarin(ij,kl)
2625 #ifdef ZIGGURAT
2626          if (use_ziggurat) call
2627 #endif
2628      end subroutine initRandom
2629
2630      subroutine RandIndices(indi
2631          use AMLUtils
2632          integer, intent(in) :: nm
2633          integer indices(n),i, ix
2634          integer tmp(nmax)
2635
2636          if (n> nmax) call MPIStop
2637          do i=1, nmax
2638              tmp(i)=i
2639          end do
2640          do i=1, n
```

/Users/lp1lopa/Compare/camb\_des/utils.F9  
0, Top line: 2610

```
2610      else
2611          kl = 9373
2612      end if
2613      ij = i
2614      else
2615          call system_clock(cou
2616          ij = mod(ij + rand_in
2617          call date_and_time(ti
2618          read (fred,'(e10.3)')
2619          kl = mod(int(klr*1000
2620      end if
2621
2622      if (Feedback > 0 ) write(
2623          call rmarin(ij,kl)
2624 #ifdef ZIGGURAT
2625          if (use_ziggurat) call zi
2626 #endif
2627      end subroutine initRandom
2628
2629      subroutine RandIndices(indi
2630          use AMLUtils
2631          integer, intent(in) :: nm
2632          integer indices(n),i, ix
2633          integer tmp(nmax)
2634
2635          if (n> nmax) call MPIStop
2636          do i=1, nmax
2637              tmp(i)=i
2638          end do
2639          do i=1, n
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2641

```
2641      ix = int(ranmar())*(nma
2642      indices(i) = tmp(ix)
2643      tmp(ix) = tmp(nmax+1-i
2644      end do
2645
2646      end subroutine RandIndices
2647
2648      subroutine RandRotationS(R,
2649      !this is most certainly no
2650      integer, intent(in) :: N
2651      real R(N,N), vec(N), norm
2652      integer i,j
2653
2654      do j = 1, N
2655          do
2656              do i = 1, N
2657                  vec(i) = Gaussian1(
2658              end do
2659              do i = 1, j-1
2660                  vec = vec - sum(ve
2661              end do
2662              norm = sum(vec**2)
2663              if (norm > 1e-3) exi
2664          end do
2665          R(j,:) = vec / sqrt(norm
2666      end do
2667
2668      end subroutine RandRotation
2669
2670
```

/Users/lp1lopa/Compare/camb\_des/utils.F9  
0, Top line: 2640

```
2640      ix = int(ranmar())*(nm
2641      indices(i) = tmp(ix)
2642      tmp(ix) = tmp(nmax+1-
2643      end do
2644
2645      end subroutine RandIndice
2646
2647      subroutine RandRotationS(
2648      !this is most certainly n
2649      integer, intent(in) :: N
2650      real R(N,N), vec(N), norm
2651      integer i,j
2652
2653      do j = 1, N
2654          do
2655              do i = 1, N
2656                  vec(i) = Gaus
2657              end do
2658              do i = 1, j-1
2659                  vec = vec - s
2660              end do
2661              norm = sum(vec**2)
2662              if (norm > 1e-3)
2663          end do
2664          R(j,:) = vec / sqrt(n
2665      end do
2666
2667      end subroutine RandRotati
2668
2669
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2671

```
2671 subroutine RandRotationD(R,  
2672 !this is most certainly no  
2673 integer, intent(in) :: N  
2674 double precision R(N,N),  
2675 integer i,j  
2676  
2677 do j = 1, N  
2678   do  
2679     do i = 1, N  
2680       vec(i) = Gaussian1(  
2681     end do  
2682     do i = 1, j-1  
2683       vec = vec - sum(ve  
2684     end do  
2685     norm = sum(vec**2)  
2686     if (norm > 1e-3) exi  
2687   end do  
2688   R(j,:) = vec / sqrt(norm)  
2689 end do  
2690  
2691 end subroutine RandRotation
```

```
2694 double precision function G  
2695 #ifdef ZIGGURAT  
2696   use Ziggurat  
2697 #endif  
2698   implicit none  
2699   double precision R, V1, V  
2700   integer, save :: iset = 0
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2670

```
2670 subroutine RandRotationD(  
2671 !this is most certainly n  
2672 integer, intent(in) :: N  
2673 double precision R(N,N),  
2674 integer i,j  
2675  
2676 do j = 1, N  
2677   do  
2678     do i = 1, N  
2679       vec(i) = Gaus  
2680     end do  
2681     do i = 1, j-1  
2682       vec = vec - s  
2683     end do  
2684     norm = sum(vec**2)  
2685     if (norm > 1e-3)  
2686   end do  
2687   R(j,:) = vec / sqrt(n  
2688 end do  
2689  
2690 end subroutine RandRotati
```

```
2693 double precision function  
2694 #ifdef ZIGGURAT  
2695   use Ziggurat  
2696 #endif  
2697   implicit none  
2698   double precision R, V1, V  
2699   integer, save :: iset = 0
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2701

```
2701      double precision, save ::  
2702  
2703          if (use_ziggurat) then  
2704 #ifdef ZIGGURAT  
2705             Gaussian1 = rnor( )  
2706 #endif  
2707          else  
2708              !Box muller  
2709              if (ISET==0) then  
2710                  R=2  
2711                  do while (R >= 1.d0)  
2712                      V1=2.d0*ranmar()-1.d0  
2713                      V2=2.d0*ranmar()-1.d0  
2714                      R=V1**2+V2**2  
2715                  end do  
2716                  FAC=sqrt(-2.d0*log(R))  
2717                  GSET=V1*FAC  
2718                  GAUSSIAN1=V2*FAC  
2719                  ISET=1  
2720              else  
2721                  GAUSSIAN1=GSET  
2722                  ISET=0  
2723              endif  
2724          end if  
2725      end function GAUSSIAN1  
2726  
2727  
2728      double precision function  
2729          implicit none  
2730
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2700

```
2700      double precision, save ::  
2701  
2702          if (use_ziggurat) then  
2703 #ifdef ZIGGURAT  
2704             Gaussian1 = rnor( )  
2705 #endif  
2706          else  
2707              !Box muller  
2708              if (ISET==0) then  
2709                  R=2  
2710                  do while (R >= 1.  
2711                      V1=2.d0*ranma  
2712                      V2=2.d0*ranma  
2713                      R=V1**2+V2**2  
2714                  end do  
2715                  FAC=sqrt(-2.d0*lo  
2716                  GSET=V1*FAC  
2717                  GAUSSIAN1=V2*FAC  
2718                  ISET=1  
2719              else  
2720                  GAUSSIAN1=GSET  
2721                  ISET=0  
2722              endif  
2723          end if  
2724      end function GAUSSIAN1  
2725  
2726  
2727  
2728      double precision function  
2729          implicit none  
2730
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2731

```
2731      Cauchy1 = Gaussian1()/m
2732
2733      end function CAUCHY1
2734
2735
2736      real FUNCTION RANDEXP1()
2737      !
2738      ! Random-number generator
2739      ! Algorithm EA from J. H.
2740      ! Communications of the A
2741      ! Coded by K. G. Hamilton
2742      !
2743      real u, up, g, y
2744
2745      real, parameter :: al
2746      real, parameter :: a
2747      real, parameter :: b
2748      real, parameter :: c
2749      real, parameter :: p
2750      real, parameter :: aa
2751      real, parameter :: bb
2752      real, parameter :: hh
2753      real, parameter :: dd
2754
2755      u = ranmar()
2756      do while (u.le.0)
2757          u = ranmar()
2758      enddo
2759      g = c
2760      u = u+u
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2730

```
2730      Cauchy1 = Gaussian1()/max
2731
2732
2733
2734
2735
2736      real FUNCTION RANDEXP1()
2737      !
2738      ! Random-number gener
2739      ! Algorithm EA from J
2740      ! Communications of t
2741      ! Coded by K. G. Hami
2742      !
2743
2744      real, parameter :: alog
2745      real, parameter :: a
2746      real, parameter :: b
2747      real, parameter :: c
2748      real, parameter :: p
2749      real, parameter :: aa
2750      real, parameter :: bb
2751      real, parameter :: hh
2752      real, parameter :: dd
2753
2754      u = ranmar()
2755      do while (u.le.0)
2756          u = ranmar()
2757      enddo
2758      g = c
2759      u = u+u
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2761

```
2761      do while (u.lt.1.0)
2762          g = g + alog2
2763          u = u+u
2764      enddo
2765      u = u-1.0
2766      if (u.le.p) then
2767          randexpl = g + aa/(bb
2768          return
2769      endif
2770      do
2771          u = ranmar()
2772          y = a/(b-u)
2773          up = ranmar()
2774          if ((up*hh+dd)*(b-u) *
2775              randexpl = g+y
2776              return
2777          endif
2778      enddo
2779
2780      end function randexpl
2781
2782
2783 ! This random number generato
2784 ! Random Number Generator' b
2785 ! Florida State University Re
2786 !
2787 ! It was later modified by F.
2788 ! random Number Generators'
2789 !
2790 ! THIS IS THE BEST KNOWN RAND
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2760

```
2760      do while (u.lt.1.0)
2761          g = g + alog2
2762          u = u+u
2763      enddo
2764      u = u-1.0
2765      if (u.le.p) then
2766          randexpl = g + aa/(bb
2767          return
2768      endif
2769      do
2770          u = ranmar()
2771          y = a/(b-u)
2772          up = ranmar()
2773          if ((up*hh+dd)*(b-u) *
2774              randexpl = g+y
2775              return
2776          endif
2777      enddo
2778
2779      end function randexpl
2780
2781
2782 ! This random number gene
2783 ! Random Number Generator
2784 ! Florida State Universit
2785 !
2786 ! It was later modified b
2787 ! random Number Generator
2788 !
2789 ! THIS IS THE BEST KNOWN
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2791

```
2791 !      (However, a newly discov
2792 !          a period of 10^600.
2793 !
2794 ! It passes ALL of the tests
2795 ! of 2^144, is completely p
2796 ! machines with at least 24
2797 ! representation).
2798 !
2799 ! The algorithm is a combinat
2800 ! and 33, and operation "su
2801 ! "arithmetic sequence" (us
2802 !
2803 ! On a Vax 11/780, this rando
2804 ! 13 microseconds.
2805 =====
2806 !
2807 !      PROGRAM TstrAN
2808 !      INTEGER IJ, KL, I
2809 ! Thee are the seeds needed t
2810 !      IJ = 1802
2811 !      KL = 9373
2812 !
2813 !
2814 ! Do the initialization
2815 !      call rmarin(ij,kl)
2816 !
2817 ! Generate 20000 random numbe
2818 !      do 10 I = 1, 20000
2819 !          x = RANMAR()
2820 !10      continue
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2790

```
2790 !      (However, a newly di
2791 !          a period of 10^6
2792 !
2793 ! It passes ALL of the te
2794 ! of 2^144, is complete
2795 ! machines with at leas
2796 ! representation).
2797 !
2798 ! The algorithm is a comb
2799 ! and 33, and operation
2800 ! "arithmetic sequence"
2801 !
2802 ! On a Vax 11/780, this r
2803 ! 13 microseconds.
2804 =====
2805 !
2806 !      PROGRAM TstrAN
2807 !      INTEGER IJ, KL, I
2808 ! Thee are the seeds need
2809 !      IJ = 1802
2810 !      KL = 9373
2811 !
2812 !
2813 ! Do the initialization
2814 !      call rmarin(ij,kl)
2815 !
2816 ! Generate 20000 random n
2817 !      do 10 I = 1, 20000
2818 !          x = RANMAR()
2819 !10      continue
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2821

```
2821 !  
2822 ! If the random number genera  
2823 ! numbers should be:  
2824 ! 6533892.0 1422022  
2825 ! 6172232.0 8354498.0 1  
2826 !  
2827 !  
2828 !  
2829 ! write(6,20) (4096.0*40  
2830 !20 format (3f12.1)  
2831 ! end  
2832 !  
2833 subroutine RMARIN(IJ,KL)  
2834 ! This is the initialization  
2835 ! NOTE: The seed variables ca  
2836 !  
2837 !The random number sequences  
2838 ! length to complete an entir  
2839 ! different groups are workin  
2840 ! each group could be assigne  
2841 ! with 30000 choices for the  
2842 ! number generator can create  
2843 ! each subsequence having a l  
2844 !  
2845 ! Use IJ = 1802 & KL = 9373 t  
2846 ! subroutine RANMAR should be  
2847 ! Then display the next six r  
2848 ! If the random number genera  
2849 ! should be:  
2850 ! 6533892.0 142202
```

/Users/lp1lopa/Compare/camb\_des/utils.F9  
0, Top line: 2820

```
2820 !  
2821 ! If the random number ge  
2822 ! numbers should be:  
2823 ! 6533892.0 142  
2824 ! 6172232.0 8354498.0  
2825 !  
2826 !  
2827 !  
2828 ! write(6,20) (4096.  
2829 !20 format (3f12.1)  
2830 !  
2831 !  
2832 subroutine RMARIN(IJ,KL)  
2833 ! This is the initializat  
2834 ! NOTE: The seed variable  
2835 !  
2836 !The random number sequen  
2837 ! length to complete an e  
2838 ! different groups are wo  
2839 ! each group could be ass  
2840 ! with 30000 choices for  
2841 ! number generator can cr  
2842 ! each subsequence having  
2843 !  
2844 ! Use IJ = 1802 & KL = 93  
2845 ! subroutine RANMAR shoul  
2846 ! Then display the next s  
2847 ! If the random number ge  
2848 ! should be:  
2849 ! 6533892.0 14
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2851

```
2851 !          6172232.0  835449
2852      double precision U(97),
2853      integer I97, J97,i,j,k,
2854      integer ij,kl
2855      integer ii,jj
2856
2857
2858 !      INTEGER IRM(103)
2859
2860      common /RASET1/ U, C, C
2861      if( IJ .lt. 0 .or. IJ
2862          KL .lt. 0 .or. KL
2863          print '(A)', ' The
2864          print '(A)', ' The s
2865          stop
2866      endif
2867      I = mod(IJ/177, 177) +
2868      J = mod(IJ , 177) +
2869      K = mod(KL/169, 178) +
2870      L = mod(KL, 169)
2871      do 2 II = 1, 97
2872          S = 0.0
2873          T = 0.5
2874          do 3 JJ = 1, 24
2875              M = mod(mod(I*j,
2876                  I = J
2877                  J = K
2878                  K = M
2879                  L = mod(53*L+1, 1
2880                  if (mod(L*M, 64)
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2850

```
2850 !
2851      6172232.0  83
2852      double precision U(97), C
2853      integer I97, J97,i,j,k,l,
2854      integer ij,kl
2855      integer ii,jj
2856
2857
2858 !      INTEGER IRM(103)
2859
2860      common /RASET1/ U, C, CD,
2861      if( IJ .lt. 0 .or. IJ .
2862          KL .lt. 0 .or. KL .
2863          print '(A)', ' The fi
2864          print '(A)', ' The sec
2865          stop
2866      endif
2867      I = mod(IJ/177, 177) + 2
2868      J = mod(IJ , 177) + 2
2869      K = mod(KL/169, 178) + 1
2870      L = mod(KL,
2871      do 2 II = 1, 97
2872          S = 0.0
2873          T = 0.5
2874          do 3 JJ = 1, 24
2875              M = mod(mod(I*j,
2876                  I = J
2877                  J = K
2878                  K = M
2879                  L = mod(53*L+1, 1
2880                  if (mod(L*M, 64)
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2881

```
2881      S = S + T
2882      endif
2883      T = 0.5 * T
2884      continue
2885      U(II) = S
2886      continue
2887      C = 362436.0 / 16777216
2888      CD = 7654321.0 / 167772
2889      CM = 16777213.0 /167772
2890      I97 = 97
2891      J97 = 33
2892
2893      end subroutine RMARIN
2894
2895      double precision functi
2896      ! This is the random number g
2897      ! Florida State University Re
2898      ! It was slightly modified by
2899      ! numbers.
2900      double precision U(97),
2901      integer I97, J97
2902      double precision uni
2903
2904      common /RASET1/ U, C, C
2905      !      INTEGER IVEC
2906      UNI = U(I97) - U(J97)
2907      if( UNI .lt. 0.0 ) U
2908      U(I97) = UNI
2909      I97 = I97 - 1
2910      if(I97 .eq. 0) I97 =
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2880

```
2880      S = S + T
2881      endif
2882      T = 0.5 * T
2883      continue
2884      U(II) = S
2885      continue
2886      C = 362436.0 / 16777216.0
2887      CD = 7654321.0 / 16777216
2888      CM = 16777213.0 /16777216
2889      I97 = 97
2890      J97 = 33
2891
2892      end subroutine RMARIN
2893
2894      double precision function
2895      ! This is the random numb
2896      ! Florida State Universit
2897      ! It was slightly modifie
2898      ! numbers.
2899      double precision U(97), C
2900      integer I97, J97
2901      double precision uni
2902
2903      common /RASET1/ U, C, CD,
2904      !      INTEGER IVEC
2905      UNI = U(I97) - U(J97)
2906      if( UNI .lt. 0.0 ) UNI =
2907      U(I97) = UNI
2908      I97 = I97 - 1
2909      if(I97 .eq. 0) I97 = 97
```

/Users/lp1lopa/Compare/camb\_simdata/util  
s.F90, Top line: 2911

```
2911      J97 = J97 - 1
2912      if(J97 .eq. 0) J97 =
2913      C = C - CD
2914      if( C .lt. 0.d0 ) C
2915      UNI = UNI - C
2916      if( UNI .lt. 0.d0 )
2917      RANMAR = UNI
2918
2919      end function RANMAR
2920
2921
2922      end module Random
2923
2924
2925
```

/Users/lp1lopa/Compare/camb\_des/utils.F90, Top line: 2910

```
2910      J97 = J97 - 1
2911      if(J97 .eq. 0) J97 = 97
2912      C = C - CD
2913      if( C .lt. 0.d0 ) C = C +
2914      UNI = UNI - C
2915      if( UNI .lt. 0.d0 ) UNI =
2916      RANMAR = UNI
2917
2918      end function RANMAR
2919
2920
2921      end module Random
2922
2923
2924
```