

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

```
001      !Interface module for CAMB
002
003      module CAMB
004      use Precision
005      use ModelParams
006      use ModelData
007      use Transfer
008      use GaugeInterface
009      use InitialPower
010      use Reionization
011      use Recombination
012      use lensing
013
014
015      !#SimDataAdd
016      use CSGalCalc
017      use Clusters
018      !#SimDataAdd
019
020      implicit none
021
022      Type CAMBdata
023          Type (ClTransferData)
024          Type (MatterTransferDa
025          Type (CAMBparams) :: P
026      end Type CAMBdata
027
028      !          public CAMB_GetT
029      !          CAMB_Vali
030      contains
```

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

```
001      !Interface module for CAMB
002
003      module CAMB
004      use Precision
005      use ModelParams
006      use ModelData
007      use Transfer
008      use GaugeInterface
009      use InitialPower
010      use Reionization
011      use Recombination
012      use lensing
013
014
015
016
017
018
019
020      implicit none
021
022
023      Type CAMBdata
024          Type (ClTransferData)
025          Type (MatterTransferDa
026          Type (CAMBparams) :: P
027      end Type CAMBdata
028
029      !          public CAMB_GetT
030      !          CAMB_Vali
031      contains
```

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

```
031
032      subroutine CAMB_GetTransfe
033      use CAMBmain
034      use lensing
035      type(CAMBparams) :: Params
036      type (CAMBdata)  :: OutDat
037      integer :: error !Zero if

038
039      !Set internal types from 0
040
041      MT = OutData%MTrans
042

043      CTransScal = OutData%ClTra
044      CTransVec  = OutData%ClTra
045      CTransTens = OutData%ClTra
046
047
048      call CAMB_GetResults(Param

049      OutData%Params = Params
050      OutData%MTrans = MT

051      OutData%ClTransScal = CTra
052      OutData%ClTransVec  = CTra
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 24

```
024
025      subroutine CAMB_GetTransfe
026      use CAMBmain
027      use lensing
028      type(CAMBparams) :: Params
029      type (CAMBdata)  :: OutDat
030      integer :: error !Zero if
031      Type(MatterTransferData) :
032      Type(ClTransferData) :: em

033
034      !Set internal types from 0
035
036      call Transfer_Free(MT)
037      MT = OutData%MTrans

038
039      call Free_ClTransfer(CTran
040      call Free_ClTransfer(CTran
041      call Free_ClTransfer(CTran
042      CTransScal = OutData%ClTra
043      CTransVec  = OutData%ClTra
044      CTransTens = OutData%ClTra
045
046
047      call CAMB_GetResults(Param

048      OutData%Params = Params
049      OutData%MTrans = MT
050      MT = emptyMT
051      OutData%ClTransScal = CTra
052      OutData%ClTransVec  = CTra
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 53

```
053      OutData%ClTransTens = CTra
054
055      end subroutine CAMB_GetTra
056
057      subroutine CAMB_InitCAMBda
058      type (CAMBdata) :: Dat
059
060      !Comment these out to try
061      call Ranges_Nullify(Dat%Cl
062      call Ranges_Nullify(Dat%Cl
063      call Ranges_Nullify(Dat%Cl
064
065      nullify(Dat%ClTransScal%De
066      nullify(Dat%ClTransVec%Del
067      nullify(Dat%ClTransTens%De
068      nullify(Dat%MTrans%sigma_8
069
070      end subroutine CAMB_InitCA
071
072
073      subroutine CAMB_FreeCAMBda
074      type (CAMBdata) :: Dat
075
076      call Free_ClTransfer(Dat%C
077      call Free_ClTransfer(Dat%C
078      call Free_ClTransfer(Dat%C
079      call Transfer_Free(Dat%MTr
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 53

```
053      OutData%ClTransTens = CTra
054      CTransScal = emptyCl
055      CTransVec = emptyCl
056      CTransTens = emptyCl
057
058      end subroutine CAMB_GetTra
059
060      subroutine CAMB_InitCAMBda
061      type (CAMBdata) :: Dat
062
063      !Comment these out to try
064      call Ranges_Nullify(Dat%Cl
065      call Ranges_Nullify(Dat%Cl
066      call Ranges_Nullify(Dat%Cl
067
068      nullify(Dat%ClTransScal%De
069      nullify(Dat%ClTransVec%Del
070      nullify(Dat%ClTransTens%De
071      nullify(Dat%MTrans%sigma_8
072
073      end subroutine CAMB_InitCA
074
075
076      subroutine CAMB_FreeCAMBda
077      type (CAMBdata) :: Dat
078
079      call Free_ClTransfer(Dat%C
080      call Free_ClTransfer(Dat%C
081      call Free_ClTransfer(Dat%C
082      call Transfer_Free(Dat%MTr
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 80

```
080
081      end subroutine CAMB_FreeCA
082
083
084      subroutine CAMB_TransferST
085      use CAMBmain
086      use lensing
087      type (CAMBdata) :: CData
088
089      CP = CData%Params
090      call InitializePowers(CP%I
091      if (global_error_flag/=0)
092      if (CData%Params%WantCls)
093          call ClTransferToCl(CD
094          if (CP%DoLensing .and.
095          if (global_error_flag/
096      end if
097      if (CData%Params%WantTrans
098
099      end subroutine CAMB_Transf
100
101
102      !Call this routine with a
103      subroutine CAMB_GetResults
104      use CAMBmain
105      use lensing
106      use Bispectrum
107      use Errors
108      type(CAMBparams) :: Params
109      integer, optional :: error
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 83

```
083
084      end subroutine CAMB_FreeCA
085
086
087      subroutine CAMB_TransferST
088      use CAMBmain
089      use lensing
090      type (CAMBdata) :: CData
091
092      CP = CData%Params
093      call InitializePowers(CP%I
094      if (global_error_flag/=0)
095      if (CData%Params%WantCls)
096          call ClTransferToCl(CD
097          if (CP%DoLensing .and.
098          if (global_error_flag/
099      end if
100      if (CData%Params%WantTrans
101
102      end subroutine CAMB_Transf
103
104
105      !Call this routine with a
106      subroutine CAMB_GetResults
107      use CAMBmain
108      use lensing
109      use Bispectrum
110      use Errors
111      type(CAMBparams) :: Params
112      integer, optional :: error
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 110

```
110      type(CAMBparams) P
111      logical :: separate = .fal
112      logical :: InReionization
113      integer jj,jj_Pk,iin
114
115      !#SimData
116      type(OutputCls) :: dummy
117      type(OutputClusters) :: du
118      !#SimData
119      !JD no longer need to calc
120      !      if ((Params%DoLensing
121      !      .and. Params%NonLinea
122      InReionization = Params%Re
123      global_error_flag = 0
124      call_again = .false.
125
126      if (Params%WantCls .and. P
127      P = Params
128      if (HighAccuracyDefaul
129      P%Max_eta_k=max(mi
130      end if
131
132      if (separate) then
133      P%WantTransfer = .
134      P%Transfer%high_pr
135
136      end if
137      P%WantTensors = .false
138      P%WantVectors = .false
139      call CAMBParams_Set(P)
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 113

```
113      type(CAMBparams) P
114      logical :: separate = .fal
115      logical :: InReionization
116
117      !JD no longer need to calc
118      !      if ((Params%DoLensing
119      !      .and. Params%NonLinea
120      InReionization = Params%Re
121      global_error_flag = 0
122      call_again = .false.
123
124      if (Params%WantCls .and. P
125      P = Params
126      if (HighAccuracyDefaul
127      P%Max_eta_k=max(mi
128      end if
129
130      if (separate) then
131      P%WantTransfer = .
132      P%Transfer%high_pr
133      P%Transfer%accurat
134      end if
135      P%WantTensors = .false
136      P%WantVectors = .false
137      call CAMBParams_Set(P)
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 139

```
139      if (global_error_flag=  
140      if (global_error_flag/  
141          if (present(error)  
142              return  
143      end if  
144      call_again = .true.  
145      !Need to store CP%flat  
146      CP%Transfer%high_preci  
  
147      CP%WantTransfer = Para  
148      CP%WantTensors = Param  
149      CP%WantVectors = Param  
150      CP%Transfer%num_redshi  
151      !JD 08/13 for nonlinea  
152      CP%Transfer%PK_redshif  
153      CP%Transfer%PK_num_red  
154      Params = CP  
155  end if  
  
156  
157  if (Params%WantCls .and. P  
158      P=Params  
159      P%WantTransfer = .fals  
160      P%Transfer%high_precis  
161      P%WantScalars = .false  
162      P%WantVectors = .false  
163      call CAMBParams_Set(P)  
164      if (global_error_flag=  
165      if (global_error_flag/  
166          if (present(error)  
167          return
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 138

```
138      if (global_error_flag=  
139      if (global_error_flag/  
140          if (present(error)  
141              return  
142      end if  
143      call_again = .true.  
144      !Need to store CP%flat  
145      CP%Transfer%high_preci  
146      CP%Transfer%accurate_m  
  
147      CP%WantTransfer = Para  
148      CP%WantTensors = Param  
149      CP%WantVectors = Param  
150      CP%Transfer%num_redshi  
151      !JD 08/13 for nonlinea  
152      CP%Transfer%PK_redshif  
153      CP%Transfer%PK_num_red  
154      Params = CP  
155  end if  
  
156  
157  if (Params%WantCls .and. P  
158      P=Params  
159      P%WantTransfer = .fals  
160      P%Transfer%high_precis  
161      P%WantScalars = .false  
162      P%WantVectors = .false  
163      call CAMBParams_Set(P)  
164      if (global_error_flag=  
165      if (global_error_flag/  
166          if (present(error)  
167          return
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 168

```
168      end if
169      call_again = .true.
170      CP%Transfer%high_preci
171      CP%WantTransfer = Para
172      CP%WantScalars = Param
173      CP%WantVectors = Param
174      CP%Transfer%num_redshi
175      !JD 08/13 for nonlinear
176      CP%Transfer%PK_redshif
177      CP%Transfer%PK_num_red
178      Params = CP
179  end if
180
181  if (Params%WantCls .and. P
182      P=Params
183      P%WantTransfer = .fals
184      P%Transfer%high_precis
185      P%WantScalars = .false
186      P%WantTensors = .false
187      call CAMBParams_Set(P)
188      if (global_error_flag=
189      if (global_error_flag/
190          if (present(error)
191          return
192      end if
193      call_again = .true.
194      CP%Transfer%high_preci
195      CP%WantTransfer = Para
196      CP%WantTensors = Param
197      CP%WantScalars = Param
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 168

```
168      end if
169      call_again = .true.
170      CP%Transfer%high_preci
171      CP%WantTransfer = Para
172      CP%WantScalars = Param
173      CP%WantVectors = Param
174      CP%Transfer%num_redshi
175      !JD 08/13 for nonlinear
176      CP%Transfer%PK_redshif
177      CP%Transfer%PK_num_red
178      Params = CP
179  end if
180
181  if (Params%WantCls .and. P
182      P=Params
183      P%WantTransfer = .fals
184      P%Transfer%high_precis
185      P%WantScalars = .false
186      P%WantTensors = .false
187      call CAMBParams_Set(P)
188      if (global_error_flag=
189      if (global_error_flag/
190          if (present(error)
191          return
192      end if
193      call_again = .true.
194      CP%Transfer%high_preci
195      CP%WantTransfer = Para
196      CP%WantTensors = Param
197      CP%WantScalars = Param
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 198

```
198      CP%Transfer%num_redshi
199      !JD 08/13 for nonlinea
200      CP%Transfer%PK_redshif
201      CP%Transfer%PK_num_red
202      Params = CP
203  end if
204
205  if (Params%WantTransfer .a
206 .not. (Params%WantCls .and
207      P=Params
208      P%WantCls = .false.
209      P%WantScalars = .false
210      P%WantTensors = .false
211      P%WantVectors = .false
212      call CAMBParams_Set(P)
213      if (global_error_flag=
214      if (global_error_flag/
215          if (present(error)
216              return
217      end if
218      !Need to store num red
219      CP%WantScalars = Param
220      CP%WantCls = Params%W
221      CP%WantTensors = Param
222      CP%WantVectors = Param
223      CP%Reion%Reionization
224      Params = CP
225  end if
226
227  call_again = .false.
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 198

```
198      CP%Transfer%num_redshi
199      !JD 08/13 for nonlinea
200      CP%Transfer%PK_redshif
201      CP%Transfer%PK_num_red
202      Params = CP
203  end if
204
205  if (Params%WantTransfer .a
206 .not. (Params%WantCls
207      P=Params
208      P%WantCls = .false.
209      P%WantScalars = .false
210      P%WantTensors = .false
211      P%WantVectors = .false
212      call CAMBParams_Set(P)
213      if (global_error_flag=
214      if (global_error_flag/
215          if (present(error)
216              return
217      end if
218      !Need to store num red
219      CP%WantScalars = Param
220      CP%WantCls = Params%W
221      CP%WantTensors = Param
222      CP%WantVectors = Param
223      CP%Reion%Reionization
224      Params = CP
225  end if
226
227  call_again = .false.
```


/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 228

```
228
229      !#SimData
230      if(Params%DoShePowFoC) the
231          if ((.not. Params%CSGa
232              if ((.not. Params%CSGa
233          end if
234      !#SimData
235
236      !Luci
237      !call Transfer_output_Sig8(MT)
238
239      !      Do iin=1,CP%InitPower%nn
240      !      Do jj_Pk=1,CP%Transfe
241      !      jj=CP%Transfer%Pk_red
242      !      print*,CP%Transfer%re
243      !      MT%sigma2_vde
244      !      Enddo
245      !      Enddo
246
247
248
249      if (.not. CP%OnlyTransfers
250          if (CP%DoLensing .and.
251              call lens_Cls
252          end if
253
254          if (do_bispectrum .and
255      end if
256
257      end subroutine CAMB_GetRes
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 228

```
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249      if (.not. CP%OnlyTransfers
250          if (CP%DoLensing .and.
251              call lens_Cls
252          end if
253
254          if (do_bispectrum .and
255      end if
256
257      end subroutine CAMB_GetRes
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 258

```
258
259
260      !Return real (NOT double p
261      !Output is l(l+1)C_l/2pi
262      !If GC_Conventions = .fals
263      subroutine CAMB_GetCls(Cls
264      integer, intent(IN) :: lma
265      logical, intent(IN) :: GC_
266      real, intent(OUT) :: Cls(2
267      integer l
268
269      Cls = 0
270      do l=2, lmax
271          if (CP%WantScalars .an
272              if (CP%DoLensing)
273                  if (l<=lmax_le
274              else
275                  Cls(l,1:2) = C
276                  Cls(l,4) = Cl_
277              endif
278          end if
279          if (CP%WantTensors .an
280              Cls(l,1:4) = Cls(l
281          end if
282      end do
283      if (GC_conventions) then
284          Cls(:,2:3) = Cls(:,2:3
285          Cls(:,4)   = Cls(:,4)/
286      end if
287
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 238

```
238
239
240      !Return real (NOT double p
241      !Output is l(l+1)C_l/2pi
242      !If GC_Conventions = .fals
243      subroutine CAMB_GetCls(Cls
244      integer, intent(IN) :: lma
245      logical, intent(IN) :: GC_
246      real, intent(OUT) :: Cls(2
247      integer l
248
249      Cls = 0
250      do l=2, lmax
251          if (CP%WantScalars .an
252              if (CP%DoLensing)
253                  if (l<=lmax_le
254              else
255                  Cls(l,1:2) = C
256                  Cls(l,4) = Cl_
257              endif
258          end if
259          if (CP%WantTensors .an
260              Cls(l,1:4) = Cls(l
261          end if
262      end do
263      if (GC_conventions) then
264          Cls(:,2:3) = Cls(:,2:3
265          Cls(:,4)   = Cls(:,4)/
266      end if
267
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 288

```
288      end subroutine CAMB_GetCls
289
290      function CAMB_GetAge(P)
291      !Return age in gigayears,
292      type(CAMBparams), intent(i
293      real(dl) CAMB_GetAge
294      integer error
295
296      call CAMBParams_Set(P, er
297
298      if (error/=0) then
299          CAMB_GetAge = -1
300      else
301          CAMB_GetAge = DeltaPhy
302      end if
303
304      end function CAMB_GetAge
305
306
307      function CAMB_GetZreFromTa
308      type(CAMBparams) :: P
309      real(dl) tau
310      real(dl) CAMB_GetZreFromTa
311      integer error
312
313      P%Reion%use_optical_depth
314      P%Reion%optical_depth = ta
315      call CAMBParams_Set(P, erro
316
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 268

```
268      end subroutine CAMB_GetCls
269
270      function CAMB_GetAge(P)
271      !Return age in gigayears,
272      type(CAMBparams), intent(i
273      real(dl) CAMB_GetAge
274      integer error
275
276      call CAMBParams_Set(P, er
277
278      if (error/=0) then
279          CAMB_GetAge = -1
280      else
281          CAMB_GetAge = DeltaPhy
282      end if
283
284      end function CAMB_GetAge
285
286
287      function CAMB_GetZreFromTa
288      type(CAMBparams) :: P
289      real(dl) tau
290      real(dl) CAMB_GetZreFromTa
291      integer error
292
293      P%Reion%use_optical_depth
294      P%Reion%optical_depth = ta
295      call CAMBParams_Set(P, erro
296      if (error/=0) then
297          CAMB_GetZreFromTau = -
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 317

```
317      CAMB_GetZreFromTau = CP%Re
318
319      end function CAMB_GetZreFr
320
321
322      subroutine CAMB_SetDefPara
323      use Bispectrum
324      use constants
325      type(CAMBparams), intent(o
326
327      P%WantTransfer= .false.
328      P%WantCls = .true.
329
330      P%omegab = .045
331      P%omegac = 0.255
332      P%omegav = 0.7
333      P%omegan = 0
334      P%H0 = 65
335
336      P%TCMB = COBE_CMBTemp
337      P%YHe = 0.24
338      P%Num_Nu_massless = default
339      P%Num_Nu_massive = 0
340      P%share_delta_neff = .fals
341      P%Nu_mass_eigenstates = 0
342      P%Nu_mass_numbers=0
343
344      P%Scalar_initial_condition
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 298

```
298      else
299      CAMB_GetZreFromTau = C
300      end if
301
302      end function CAMB_GetZreFr
303
304
305      subroutine CAMB_SetDefPara
306      use Bispectrum
307      use constants
308      type(CAMBparams), intent(o
309
310      P%WantTransfer= .false.
311      P%WantCls = .true.
312
313      P%omegab = .045
314      P%omegac = 0.255
315      P%omegav = 0.7
316      P%omegan = 0
317      P%H0 = 65
318
319      P%TCMB = COBE_CMBTemp
320      P%YHe = 0.24
321      P%Num_Nu_massless = default
322      P%Num_Nu_massive = 0
323      P%share_delta_neff = .fals
324      P%Nu_mass_eigenstates = 0
325      P%Nu_mass_numbers=0
326
327      P%Scalar_initial_condition
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 345

```
345      P%NonLinear = NonLinear_no
346      P%Want_CMB = .true.
347
348      call SetDefPowerParams(P%I
349
350      call Recombination_SetDefP
351
352      call Reionization_SetDefPa
353
354      P%Transfer%high_precision=
355
356      P%OutputNormalization = ou
357
358      P%WantScalars = .true.
359      P%WantVectors = .false.
360      P%WantTensors = .false.
361      P%want_zstar = .false.    !!
362      P%want_zdrag = .false.    !!
363
364      P%Max_l=1500
365      P%Max_eta_k=3000
366      P%Max_l_tensor=400
367      P%Max_eta_k_tensor=800
368      !Set up transfer just enou
369      P%Transfer%kmax=0.9
370      P%Transfer%k_per_logint=0
371      P%Transfer%num_redshifts=1
372      P%Transfer%redshifts=0
373      !JD 08/13 CAMB Fix for for
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 328

```
328      P%NonLinear = NonLinear_no
329      P%Want_CMB = .true.
330
331      call SetDefPowerParams(P%I
332
333      call Recombination_SetDefP
334
335      call Reionization_SetDefPa
336
337      P%Transfer%high_precision=
338      P%Transfer%accurate_massiv
339
340      P%OutputNormalization = ou
341
342      P%WantScalars = .true.
343      P%WantVectors = .false.
344      P%WantTensors = .false.
345      P%want_zstar = .false.    !!
346      P%want_zdrag = .false.    !!
347
348      P%Max_l=2500
349      P%Max_eta_k=5000
350      P%Max_l_tensor=600
351      P%Max_eta_k_tensor=1200
352      !Set up transfer just enou
353      P%Transfer%kmax=0.9
354      P%Transfer%k_per_logint=0
355      P%Transfer%num_redshifts=1
356      P%Transfer%redshifts=0
357      !JD 08/13 CAMB Fix for for
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 374

```
374      P%Transfer%PK_num_redshift
375      P%Transfer%PK_redshifts=0
376      P%Transfer%NLL_num_redshif
377      P%Transfer%NLL_redshifts=0
378      !End JD
379
380      P%AccuratePolarization = .
381      P%AccurateReionization = .
382      P%AccurateBB = .false.
383
384      P%DoLensing = .false.
385
386      P%MassiveNuMethod = Nu_bes
387      P%OnlyTransfers = .false.
388
389      P%DerivedParameters = .tru
390
391      end subroutine CAMB_SetDef
392
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 358

```
358      P%Transfer%PK_num_redshift
359      P%Transfer%PK_redshifts=0
360      P%Transfer%NLL_num_redshif
361      P%Transfer%NLL_redshifts=0
362      !End JD
363
364      P%AccuratePolarization = .
365      P%AccurateReionization = .
366      P%AccurateBB = .false.
367
368      P%DoLensing = .true.
369
370      P%MassiveNuMethod = Nu_bes
371      P%OnlyTransfers = .false.
372
373      P%DerivedParameters = .tru
374
375      end subroutine CAMB_SetDef
376
377      subroutine CAMB_SetNeutrino
378      use constants
379      type(CAMBparams), intent(i
380      real(dl), intent(in) :: om
381      integer, intent(in) :: neu
382      integer, intent(in), optio
383      integer, parameter :: neut
384      real(dl) normal_frac, m3,
385      real(dl), external :: Newt
386
387      if (omnuh2==0) return
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 393

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 388

```
388      P%Nu_mass_eigenstates=0
389      if ( omnuh2 > omnuh2_steri
390          normal_frac = (omnuh2
391          if (neutrino_hierarchy
392              neff_massive_stand
393              P%Num_Nu_Massive =
394              P%Nu_mass_eigensta
395              if (nnu > neff_mas
396                  P%Num_Nu_Massl
397              else
398                  P%Num_Nu_Massl
399                  neff_massive_s
400              end if
401              P%Nu_mass_numbers(
402              P%Nu_mass_degenera
403              P%Nu_mass_fraction
404          else
405              !Use normal or inv
406              mnu = (omnuh2 - om
407              if (neutrino_hiera
408                  if (mnu > mnu_
409                      !Two eigen
410                      m1=Newton_
411                      P%Num_Nu_M
412                  else
413                      !One eigen
414                      P%Num_Nu_M
415                  end if
416              else if (neutrino_
417                  if (mnu > sqrt
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 393

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 418

```
418      !Valid cas
419      m1=Newton_
420      P%Num_Nu_M
421      else
422      !Unphysica
423      P%Num_Nu_M
424      end if
425  else
426      error stop 'Un
427  end if
428  neff_massive_stand
429  if (nnu > neff_mas
430      P%Num_Nu_Mass1
431  else
432      P%Num_Nu_Mass1
433      neff_massive_s
434  end if
435  if (P%Num_Nu_Massi
436      !two with mass
437      P%Nu_mass_eige
438      P%Nu_mass_dege
439      P%Nu_mass_dege
440      m3 = mnu - 2*m
441      P%Nu_mass_frac
442      P%Nu_mass_frac
443      P%Nu_mass_numb
444      P%Nu_mass_numb
445  else
446      P%Nu_mass_dege
447      P%Nu_mass_numb
```


/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 393

```
393
394      !Stop with error is not go
395      function CAMB_ValidatePara
396      type(CAMBparams), intent(i
397      logical OK
398
399      OK = .true.
400      if (.not. P%WantTransfer .
401          OK = .false.
402          write(*,*) 'There is n
403      end if
404
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 448

```
448      P%Nu_mass_eige
449      P%Nu_mass_frac
450      end if
451      end if
452      else
453      neff_massive_standard=
454      end if
455      if (omnuh2_sterile>0) then
456      if (nnu<default_nnu) c
457      P%Num_Nu_Massless = de
458      P%Num_Nu_Massive=P%Num
459      P%Nu_mass_eigenstates=
460      P%Nu_mass_numbers(P%Nu
461      P%Nu_mass_degeneracies
462      P%Nu_mass_fractions(P%
463      end if
464      end subroutine CAMB_SetNeu
465
466
467      !Stop with error is not go
468      function CAMB_ValidatePara
469      type(CAMBparams), intent(i
470      logical OK
471
472      OK = .true.
473      if (.not. P%WantTransfer .
474          OK = .false.
475          write(*,*) 'There is n
476      end if
477
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 405

```
405      if (P%h0 < 20._dl.or.P%h0
406          OK = .false.
407          write(*,*) ' Warning:
408      end if
409      if (P%tcmb < 2.7d0.or.P%tc
410          write(*,*) ' Warning:
411      end if
412
413      if (P%yhe < 0.2d0.or.P%yhe
414          OK = .false.
415          write(*,*) &
416          ' Warning: YHe is the
417          ' Your have:', P%yhe
418      end if
419      if (P%Num_Nu_massive < 0)
420          OK = .false.
421          write(*,*) &
422          'Warning: Num_Nu_massi
423      end if
424      if (P%Num_Nu_massless < 0)
425          OK = .false.
426          write(*,*) &
427          'Warning: Num_nu_massl
428      end if
429      if (P%Num_Nu_massive < 1 .
430          OK = .false.
431          write(*,*) &
432          'Warning: You have ome
433      end if
434
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 478

```
478      if (P%h0 < 20._dl.or.P%h0
479          OK = .false.
480          write(*,*) ' Warning:
481      end if
482      if (P%tcmb < 2.7d0.or.P%tc
483          write(*,*) ' Warning:
484      end if
485
486      if (P%yhe < 0.2d0.or.P%yhe
487          OK = .false.
488          write(*,*) &
489          ' Warning: YHe is
490          ' Your have:', P%
491      end if
492      if (P%Num_Nu_massive < 0)
493          OK = .false.
494          write(*,*) &
495          'Warning: Num_Nu_m
496      end if
497      if (P%Num_Nu_massless < 0)
498          OK = .false.
499          write(*,*) &
500          'Warning: Num_nu_m
501      end if
502      if (P%Num_Nu_massive < 1 .
503          OK = .false.
504          write(*,*) &
505          'Warning: You have
506      end if
507
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 435

```
435
436      if (P%omegab<0.001 .or. P%
437          OK = .false.
438          write(*,*) 'Your matte
439      end if
440
441      if (P%WantScalars .and. P%
442      P%WantTensors .and. P%Max_
443          OK = .false.
444          write(*,*) 'You need M
445      end if
446
447      call Reionization_Validate
448      call Recombination_Validat
449
450      if (P%WantTransfer) then
451          if (P%transfer%num_red
452              OK = .false.
453              write(*,*) 'Maximu
454              'redshifts. You ha
455          end if
456          if (P%transfer%kmax <
457              P%transfer%k_per_login
458              OK = .false.
459              write(*,*) 'Strang
460          end if
461          if (P%transfer%num_red
462              OK = .false.
463              write(*,*) 'Maximu
464              'redshifts. You ha
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 508

```
508
509      if (P%omegab<0.001 .or. P%
510          OK = .false.
511          write(*,*) 'Your matte
512      end if
513
514      if (P%WantScalars .and. P%
515      P%WantTensors .and. P%
516          OK = .false.
517          write(*,*) 'You need M
518      end if
519
520      call Reionization_Validate
521      call Recombination_Validat
522
523      if (P%WantTransfer) then
524          if (P%transfer%num_red
525              OK = .false.
526              write(*,*) 'Maximu
527              'redshifts. Yo
528          end if
529          if (P%transfer%kmax <
530              P%transfer%k_per_l
531              !
532              write(*,*) 'Strang
533          end if
534          if (P%transfer%num_red
535              OK = .false.
536              write(*,*) 'Maximu
537              'redshifts. Yo
```

/Users/lp1opa/Compare/camb_simdata/camb
.f90, Top line: 465

```
465         end if
466     end if
467
468     end function CAMB_Validate
469
470     subroutine CAMB_cleanup
471     use ThermoData
472     use SpherBessels
473     use ModelData
474     use Transfer
475
476     !Free memory
477     call ThermoData_Free
478     call Bessels_Free
479     call ModelData_Free
480     call Transfer_Free(MT)
481
482     end subroutine CAMB_cleanu
483
484     end module CAMB
485
```

/Users/lp1opa/Compare/camb_des/camb.f90
, Top line: 538

```
538         end if
539     end if
540
541     end function CAMB_Validate
542
543     subroutine CAMB_cleanup
544     use ThermoData
545     use SpherBessels
546     use ModelData
547     use Transfer
548
549     !Free memory
550     call ThermoData_Free
551     call Bessels_Free
552     call ModelData_Free
553     call Transfer_Free(MT)
554
555     end subroutine CAMB_cleanu
556
557     end module CAMB
558
```