

/Users/lplopa/Compare/camb_simdata/cmbmain.f90, Top line: 1

0001	!	This this is the ma
0002	!	
0003	!	Code for Anisotropi
0004	!	by Antony lewis (ht
0005	!	See readme.html for
0006		
0007	!	Note that though th
0008	!	so you cannot gener
0009	!	
0010	!	Based on CMBFAST b
0011	!	on Boltzmann code w
0012	!	Original CMBFAST co
0013	!	
0014	!	Copyright 1996 by H
0015	!	the Massachusetts I
0016	!	
0017	!	THIS SOFTWARE IS PR
0018	!	REPRESENTATIONS OR
0019	!	By way of example,
0020	!	M.I.T. AND C.f.A MA
0021	!	MERCHANTABILITY OR
0022	!	THE USE OF THE LICE
0023	!	ANY THIRD PARTY PAT
0024	!	
0025	!	portions of this so
0026	!	E. Bertschinger. S
0027	!	for restrictions on
0028		
0029		module CAMBmain
0030		

/Users/lplopa/Compare/camb_des/cmbmain.f90, Top line: 1

0001	!	This this is the ma
0002	!	
0003	!	Code for Anisotropi
0004	!	by Antony lewis (ht
0005	!	See readme.html for
0006		
0007	!	Note that though th
0008	!	so you cannot gener
0009	!	
0010	!	Based on CMBFAST b
0011	!	on Boltzmann code w
0012	!	Original CMBFAST co
0013	!	
0014	!	Copyright 1996 by H
0015	!	the Massachusetts I
0016	!	
0017	!	THIS SOFTWARE IS PR
0018	!	REPRESENTATIONS OR
0019	!	By way of example,
0020	!	M.I.T. AND C.f.A MA
0021	!	MERCHANTABILITY OR
0022	!	THE USE OF THE LICE
0023	!	ANY THIRD PARTY PAT
0024	!	
0025	!	portions of this so
0026	!	E. Bertschinger. S
0027	!	for restrictions on
0028		
0029		module CAMBmain
0030		

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

```
0031      !      This code evolves t
0032      !      the Boltzmann equat
0033      !      of a Friedmann-Robe
0034      !      in a modules called
0035      !      computed at sampled
0036      !      are then interpolat
0037      !      sight integrals of
0038      !      For CP%flat models
0039      !      non-CP%flat models
0040      !      differential equati
0041      !      wavenumbers in para
0042
0043      !      The time variable i
0044      !      with q=sqrt(k**2 +
0045      !      The units of both l
0046
0047      !      Many elements are pa
0048      !      CP = CAMB parameters
0049      !      EV = Time evolution
0050      !      IV = Source integrat
0051      !      CT = Cl transfer dat
0052      !      MT = matter transfer
0053
0054      ! Modules are defined in
0055      ! perturbation equations,
0056
0057      use precision
0058      use ModelParams
0059      use ModelData
0060      use GaugeInterface
```

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

```
0031      !      This code evolves t
0032      !      the Boltzmann equat
0033      !      of a Friedmann-Robe
0034      !      in a modules called
0035      !      computed at sampled
0036      !      are then interpolat
0037      !      sight integrals of
0038      !      For CP%flat models
0039      !      non-CP%flat models
0040      !      differential equati
0041      !      wavenumbers in para
0042
0043      !      The time variable i
0044      !      with q=sqrt(k**2 +
0045      !      The units of both l
0046
0047      !      Many elements are pa
0048      !      CP = CAMB parameters
0049      !      EV = Time evolution
0050      !      IV = Source integrat
0051      !      CT = Cl transfer dat
0052      !      MT = matter transfer
0053
0054      ! Modules are defined in
0055      ! perturbation equations,
0056
0057      use precision
0058      use ModelParams
0059      use ModelData
0060      use GaugeInterface
```

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

```
0061      use Transfer
0062      use SpherBessels
0063      use lvalues
0064      use MassiveNu
0065      use InitialPower
0066      use Errors
0067      implicit none
0068      private
0069
0070      logical :: WantLateTime =
0071
0072      logical ExactClosedSum !
0073
0074      !Variables for integratin
0075      type IntegrationVars
0076          integer q_ix
0077          real(dl) q, dq      !q
0078          !      real(dl), di
0079          !Contribution to C_l
0080          real(dl), dimension(:
0081          !Interpolated sources
0082          integer SourceSteps !
0083      end type IntegrationVars
0084
0085      integer SourceNum
0086      !SourceNum is total numbe
0087
0088      real(dl) tautf(0:max_tran
0089
0090      real(dl), dimension(:, :, :
```

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

```
0061      use Transfer
0062      use SpherBessels
0063      use lvalues
0064      use MassiveNu
0065      use InitialPower
0066      use Errors
0067      implicit none
0068      private
0069
0070      logical :: WantLateTime =
0071
0072      logical ExactClosedSum !
0073
0074      !Variables for integratin
0075      type IntegrationVars
0076          integer q_ix
0077          real(dl) q, dq      !q
0078          !      real(dl), di
0079          !Contribution to C_l
0080          real(dl), dimension(:
0081          !Interpolated sources
0082          integer SourceSteps !
0083      end type IntegrationVars
0084
0085      integer SourceNum
0086      !SourceNum is total numbe
0087
0088      real(dl) tautf(0:max_tran
0089
0090      real(dl), dimension(:, :, :
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 91

```
0091      ! indices Src( k_index,  
0092  
0093      real(dl), dimension(:,:,:) :  
0094      ! Cls at the l values we  
0095  
0096      real(dl), dimension(:,:,:) :  
0097      !Full covariance at each  
0098  
0099      ! values of q to evolve t  
0100      Type(Regions) :: Evolve_q  
0101  
0102      real(dl),parameter :: qmi  
0103  
0104      real(dl) :: dtaurec_q  
0105  
0106      !      qmax - CP%Max_eta_k  
0107      real(dl) qmin, qmax  
0108  
0109      real(dl) max_etak_tensor  
0110      !      Will only be calcul  
0111  
0112      integer maximum_l !Max va  
0113      real(dl) :: maximum_qeta  
0114  
0115      integer :: l_smooth_sampl  
0116  
0117      real(dl) :: fixq = 0._dl  
0118  
0119      real(dl) :: ALens = 1._dl  
0120
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 91

```
0091      ! indices Src( k_index,  
0092  
0093      real(dl), dimension(:,:,:) :  
0094      ! Cls at the l values we  
0095  
0096      real(dl), dimension(:,:,:) :  
0097      !Full covariance at each  
0098  
0099      ! values of q to evolve t  
0100      Type(Regions) :: Evolve_q  
0101  
0102      real(dl),parameter :: qmi  
0103  
0104      real(dl) :: dtaurec_q  
0105  
0106      !      qmax - CP%Max_eta_k  
0107      real(dl) qmin, qmax  
0108  
0109      real(dl) max_etak_tensor  
0110      !      Will only be calcul  
0111  
0112      integer maximum_l !Max va  
0113      real(dl) :: maximum_qeta  
0114  
0115      integer :: l_smooth_sampl  
0116  
0117      real(dl) :: fixq = 0._dl  
0118  
0119      real(dl) :: ALens = 1._dl  
0120
```

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

```
0121      Type(ClTransferData), poi
0122
0123 | public cmbmain, ALens, Cl
0124
0125      contains
0126
0127
0128      subroutine cmbmain
0129      integer q_ix
0130      type(EvolutionVars) EV
0131      !      Timing variables fo
0132      real(sp) actual,timeprev,
0133
0134      WantLateTime = CP%DoLens
0135
0136      if (CP%WantCls) then
0137 |         if (CP%WantTensors .a
0138             !Use CAMB_GetResults
0139
0140             if (CP%WantTensors) t
0141                 maximum_l = CP%Ma
0142                 maximum_qeta = CP
0143             else
0144                 maximum_l = CP%Ma
0145                 maximum_qeta = CP
0146             end if
0147
0148             call initlval(lSamp,
0149         end if
0150
```

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

```
0121      Type(ClTransferData), poi
0122
0123 | public cmbmain, ALens, Cl
0124
0125      contains
0126
0127
0128      subroutine cmbmain
0129      integer q_ix
0130      type(EvolutionVars) EV
0131      !      Timing variables fo
0132      real(sp) actual,timeprev,
0133
0134      WantLateTime = CP%DoLens
0135
0136      if (CP%WantCls) then
0137 |         if (CP%WantTensors .a
0138             !Use CAMB_GetResults
0139
0140             if (CP%WantTensors) t
0141                 maximum_l = CP%Ma
0142                 maximum_qeta = CP
0143             else
0144                 maximum_l = CP%Ma
0145                 maximum_qeta = CP
0146             end if
0147
0148             call initlval(lSamp,
0149         end if
0150
```

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

```
0151
0152      if (DebugMsgs .and. Feedb
0153          actual=GetTestTime()
0154          starttime=actual !tim
0155      end if
0156
0157      call InitVars !Most of si
0158      if (global_error_flag/=0)
0159
0160          if (DebugMsgs .and. Feedb
0161              timeprev=actual
0162              actual=GetTestTime()
0163              write(*,*) actual-tim
0164              write (*,*) 'r = ',re
0165          end if
0166
0167          !JD 08/13 for nonlinear l
0168          !if (.not. CP%OnlyTransfe
0169          if (.not. CP%OnlyTransfer
0170      call InitializePowers(CP%
0171      if (global_error_flag/=0)
0172
0173          !Calculation of the CMB s
0174
0175          if (CP%WantCls) call Setk
0176
0177          if (CP%WantTransfer) call
0178
0179          !***note that !$ is the p
0180          !$ if (ThreadNum /=0) cal
```

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

```
0151
0152      if (DebugMsgs .and. Feedb
0153          actual=GetTestTime()
0154          starttime=actual !tim
0155      end if
0156
0157      call InitVars !Most of si
0158      if (global_error_flag/=0)
0159
0160          if (DebugMsgs .and. Feedb
0161              timeprev=actual
0162              actual=GetTestTime()
0163              write(*,*) actual-tim
0164              write (*,*) 'r = ',re
0165          end if
0166
0167          !JD 08/13 for nonlinear l
0168          !if (.not. CP%OnlyTransfe
0169          if (.not. CP%OnlyTransfer
0170      call InitializePowers
0171      if (global_error_flag/=0)
0172
0173          !Calculation of the CMB s
0174
0175          if (CP%WantCls) call Setk
0176
0177          if (CP%WantTransfer) call
0178
0179          !***note that !$ is the p
0180          !$ if (ThreadNum /=0) cal
```

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

```
0181
0182         if (CP%WantCls) then
0183             if (DebugMsgs .and. F
0184
0185             call GetSourceMem
0186
0187             if (CP%WantScalars) t
0188                 ThisCT => CTransS
0189             else if (CP%WantVecto
0190                 ThisCT => CTransV
0191             else
0192                 ThisCT => CTransT
0193             end if
0194
0195             ThisCT%NumSources = S
0196             ThisCT%ls = lSamp
0197
0198             !$OMP PARALLEL DO DEF
0199             !$OMP & PRIVATE(EV, q
0200             do q_ix= 1,Evolve_q%n
0201                 if (global_error_
0202             end do
0203             !$OMP END PARALLEL DO
0204
0205             if (DebugMsgs .and. F
0206                 timeprev=actual
0207                 actual=GetTestTim
0208                 write(*,*) actual
0209             end if
0210
```

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

```
0181
0182         if (CP%WantCls) then
0183             if (DebugMsgs .and. F
0184
0185             call GetSourceMem
0186
0187             if (CP%WantScalars) t
0188                 ThisCT => CTransS
0189             else if (CP%WantVecto
0190                 ThisCT => CTransV
0191             else
0192                 ThisCT => CTransT
0193             end if
0194
0195             ThisCT%NumSources = S
0196             ThisCT%ls = lSamp
0197
0198             !$OMP PARALLEL DO DEF
0199             do q_ix= Evolve_q%np
0200
0201                 if (global_error_
0202             end do
0203             !$OMP END PARALLEL DO
0204
0205             if (DebugMsgs .and. F
0206                 timeprev=actual
0207                 actual=GetTestTim
0208                 write(*,*) actual
0209             end if
0209
```

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

```
0211      endif !WantCls
0212
0213      ! If transfer functions a
0214      if (CP%WantTransfer .and.
0215          call TransferOut
0216          if (DebugMsgs .and. F
0217              timeprev=actual
0218              actual=GetTestTim
0219              write(*,*) actual
0220          end if
0221      end if
0222
0223      if (CP%WantTransfer .and.
0224          .and. (CP%NonLinear==NonL
0225          call MakeNonlinearSou
0226          if (DebugMsgs .and. F
0227              timeprev=actual
0228              actual=GetTestTim
0229              write(*,*) actual
0230          end if
0231      end if
0232
0233      if (CP%WantTransfer .and.
0234          call Transfer_Get_sigmas(
0235
0236      !      if CMB calculations
0237      !      integrating the sou
0238
0239      if (CP%WantCls) then
0240          if (global_error_flag
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 210

```
0210      endif !WantCls
0211
0212      ! If transfer functions a
0213      if (CP%WantTransfer .and.
0214          call TransferOut
0215          if (DebugMsgs .and. F
0216              timeprev=actual
0217              actual=GetTestTim
0218              write(*,*) actual
0219          end if
0220      end if
0221
0222      if (CP%WantTransfer .and.
0223          .and. (CP%NonLinear==
0224          call MakeNonlinearSou
0225          if (DebugMsgs .and. F
0226              timeprev=actual
0227              actual=GetTestTim
0228              write(*,*) actual
0229          end if
0230      end if
0231
0232      if (CP%WantTransfer .and.
0233          call Transfer_Get_sig
0234
0235      !      if CMB calculations
0236      !      integrating the sou
0237
0238      if (CP%WantCls) then
0239          if (global_error_flag
```


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

```
0241      call InitSourceIn
0242
0243      ExactClosedSum =
0244
0245      max_bessels_l_ind
0246      max_bessels_etak
0247
0248      if (CP%WantScalar
0249      ThisCT%max_index_
0250
0251      if (CP%flat) call
0252      !This is only slo
0253      !Preferably stick
0254
0255      call SetkValuesFo
0256
0257      if (DebugMsgs .an
0258
0259      !Begin k-loop and
0260      !$OMP PARALLEL DO
0261      do q_ix=1,ThisCT%
0262          call SourceTo
0263      end do !q loop
0264      !$OMP END PARALLE
0265
0266      if (DebugMsgs .an
0267          timeprev=actu
0268          actual=GetTes
0269          write(*,*)act
0270      end if
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 240

```
0240      call InitSourceIn
0241
0242      ExactClosedSum =
0243
0244      max_bessels_l_ind
0245      max_bessels_etak
0246
0247      if (CP%WantScalar
0248      ThisCT%max_index_
0249
0250      if (CP%flat) call
0251      !This is only slo
0252      !Preferably stick
0253
0254      call SetkValuesFo
0255
0256      if (DebugMsgs .an
0257
0258      !Begin k-loop and
0259      !$OMP PARALLEL DO
0260      do q_ix=1,ThisCT%
0261          call SourceTo
0262      end do !q loop
0263      !$OMP END PARALLE
0264
0265      if (DebugMsgs .an
0266          timeprev=actu
0267          actual=GetTes
0268          write(*,*)act
0269      end if
```

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

```
0271         end if
0272
0273         call FreeSourceMem
0274
0275         !Final calculations f
0276
0277         if (.not. CP%OnlyTran
0278 call ClTransferToCl(C
0279     end if
0280
0281     if (DebugMsgs .and. Feedb
0282         timeprev=actual
0283         actual = GetTestTime(
0284         write(*,*) actual - t
0285         write(*,*) actual -st
0286     end if
0287
0288     end subroutine cmbmain
0289
0290     subroutine ClTransferToCl
0291     Type(ClTransferData) :: C
0292
0293     if (CP%WantScalars .and.
0294         lSamp = CTransS%ls
0295         allocate(iCl_Scalar(C
0296         iCl_scalar = 0
0297         if (has_cl_2D_array)
0298             allocate(iCl_Arra
0299             iCl_Array = 0
0300     end if
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 270

```
0270         end if
0271
0272         call FreeSourceMem
0273
0274         !Final calculations f
0275
0276         if (.not. CP%OnlyTran
0277 call ClTransferTo
0278     end if
0279
0280     if (DebugMsgs .and. Feedb
0281         timeprev=actual
0282         actual = GetTestTime(
0283         write(*,*) actual - t
0284         write(*,*) actual -st
0285     end if
0286
0287     end subroutine cmbmain
0288
0289     subroutine ClTransferToCl
0290     Type(ClTransferData) :: C
0291
0292     if (CP%WantScalars .and.
0293         lSamp = CTransS%ls
0294         allocate(iCl_Scalar(C
0295         iCl_scalar = 0
0296         if (has_cl_2D_array)
0297             allocate(iCl_Arra
0298             iCl_Array = 0
0299     end if
```

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

```
0301
0302      call CalcLimberScalCl
0303      call CalcScalCls(CTra
0304      if (DebugMsgs .and. F
0305  end if
0306
0307      if (CP%WantVectors .and.
0308      allocate(iCl_vector(C
0309      iCl_vector = 0
0310      call CalcVecCls(CTran
0311      if (DebugMsgs .and. F
0312  end if
0313
0314
0315      if (CP%WantTensors .and.
0316      allocate(iCl_Tensor(C
0317      iCl_tensor = 0
0318      call CalcTensCls(CTra
0319      if (DebugMsgs .and. F
0320  end if
0321
0322      if (global_error_flag==0)
0323      call Init_Cls
0324      !      Calculating Cls
0325      call InterpolateCls(C
0326      if (DebugMsgs .and. F
0327  end if
0328
0329      if (CP%WantScalars .and.
0330      if (CP%WantScalars .and.
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 300

```
0300
0301      call CalcLimberScalCl
0302      call CalcScalCls(CTra
0303      if (DebugMsgs .and. F
0304  end if
0305
0306      if (CP%WantVectors .and.
0307      allocate(iCl_vector(C
0308      iCl_vector = 0
0309      call CalcVecCls(CTran
0310      if (DebugMsgs .and. F
0311  end if
0312
0313
0314      if (CP%WantTensors .and.
0315      allocate(iCl_Tensor(C
0316      iCl_tensor = 0
0317      call CalcTensCls(CTra
0318      if (DebugMsgs .and. F
0319  end if
0320
0321      if (global_error_flag==0)
0322      call Init_Cls
0323      !      Calculating Cls
0324      call InterpolateCls(C
0325      if (DebugMsgs .and. F
0326  end if
0327
0328      if (CP%WantScalars .and.
0329      if (CP%WantScalars .and.
```

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

```
0331      if (CP%WantVectors .and.  
0332      if (CP%WantTensors .and.  
0333  
0334      if (global_error_flag/=0)  
0335  
0336      if (CP%OutputNormalizatio  
0337      !Normalize to C_l=1 at l=  
0338  
0339      end subroutine ClTransfer  
0340  
0341      subroutine CalcLimberScal  
0342      Type(ClTransferData) :: C  
0343      integer ell, i, s_ix, pix  
0344      real(dl) CL, reall,fac, d  
0345      integer s_ix2,j,n  
0346      Type(LimberRec), pointer  
0347      integer winmin  
0348  
0349      if (limber_phiphi>0) then  
0350          winmin = 0  
0351      else  
0352          winmin=1  
0353      end if  
0354  
0355      do pix=1,CP%InitPower%nn  
0356          do i =winmin, num_red  
0357              s_ix = 3+i  
0358              if (CTrans%limber  
0359                  do j= i, num_  
0360                      s_ix2 = 3
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 330

```
0330      if (CP%WantVectors .and.  
0331      if (CP%WantTensors .and.  
0332  
0333      if (global_error_flag/=0)  
0334  
0335      if (CP%OutputNormalizatio  
0336      !Normalize to C_l=1 at l=  
0337  
0338      end subroutine ClTransfer  
0339  
0340      subroutine CalcLimberScal  
0341      Type(ClTransferData) :: C  
0342      integer ell, i, s_ix, pix  
0343      real(dl) CL, reall,fac, d  
0344      integer s_ix2,j,n  
0345      Type(LimberRec), pointer  
0346      integer winmin  
0347  
0348      if (limber_phiphi>0) then  
0349          winmin = 0  
0350      else  
0351          winmin=1  
0352      end if  
0353  
0354      do pix=1,CP%InitPower%nn  
0355          do i =winmin, num_red  
0356              s_ix = 3+i  
0357              if (CTrans%limber  
0358                  do j= i, num_  
0359                      s_ix2 = 3
```

```
0361      if (CTran
0362      !$OMP
0363      do el
0364          C
0365          L
0366          L
0367
0368          d
0369
0370
0371
0372          e
0373
0374          r
0375          f
0376          C
0377
0378          i
0379          i
0380
0381
0382
0383          e
0384      end d
0385      !$OMP
0386  end if
0387  end do
0388  end if
0389  end do
0390  end do
```

```
0360      if (CTran
0361      !$OMP
0362      do el
0363          C
0364          L
0365          L
0366
0367          d
0368
0369
0370
0371          e
0372
0373          r
0374          f
0375          C
0376
0377          i
0378          i
0379
0380
0381          e
0382      end d
0383      !$OMP
0384  end if
0385  end do
0386  end if
0387  end do
0388  end do
```

/Users/lplopa/Compare/camb_simdata/cmbmain.f90, Top line: 391

```
0391
0392     end subroutine CalcLimber
0393
0394     subroutine GetLimberTrans
0395     integer ell, ell_needed
0396     integer i,s_ix
0397     integer n1,n2,n, ell_limb
0398     real(dl) int,k, chi, chim
0399     integer klo, khi
0400     real(dl) a0,b0,ho2o6,a03,
0401     Type(LimberRec), pointer
0402
0403     call Init_Limber(ThisCT)
0404
0405     if (num_redshiftwindows==
0406
0407     if (ThisCT%ls%l(ThisCT%ls
0408         max_bessels_l_index =
0409     else
0410         max_bessels_l_index =
0411     end if
0412
0413     if (CP%Want_CMB) then
0414         max_bessels_etak= min
0415     else
0416         max_bessels_etak = 50
0417     end if
0418
0419     do i =0, num_redshiftwind
0420         s_ix = 3+i
```

/Users/lplopa/Compare/camb_des/cmbmain.f90, Top line: 389

```
0389
0390     end subroutine CalcLimber
0391
0392     subroutine GetLimberTrans
0393     integer ell, ell_needed
0394     integer i,s_ix
0395     integer n1,n2,n, ell_limb
0396     real(dl) int,k, chi, chim
0397     integer klo, khi
0398     real(dl) a0,b0,ho2o6,a03,
0399     Type(LimberRec), pointer
0400
0401     call Init_Limber(ThisCT)
0402
0403     if (num_redshiftwindows==
0404
0405     if (ThisCT%ls%l(ThisCT%ls
0406         max_bessels_l_index =
0407     else
0408         max_bessels_l_index =
0409     end if
0410
0411     if (CP%Want_CMB) then
0412         max_bessels_etak= min
0413     else
0414         max_bessels_etak = 50
0415     end if
0416
0417     do i =0, num_redshiftwind
0418         s_ix = 3+i
```

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

```
0421
0422         if (i==0) then
0423             ell_limb = limber
0424         end if
0425
0426         ell_needed = ThisCT%l
0427         do ell = 1, ThisCT%ls
0428             if (ThisCT%ls%l(e
0429                 ThisCT%limber
0430                 ell_needed =
0431                 max_bessels_l
0432                 if (FeedbackL
0433                     exit
0434             end if
0435         end do
0436
0437         if (ThisCT%limber_l_m
0438             if (i==0) then
0439                 n1 = Ranges_I
0440                 n2 = TimeStep
0441             end if
0442
0443             do ell = ThisCT%l
0444                 LimbRec => Th
0445                 LimbRec%n1 =
0446                 LimbRec%n2 =
0447
0448                 allocate(Limb
0449                 allocate(Limb
0450
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 419

```
0419
0420         if (i==0) then
0421             ell_limb = limber
0422         end if
0423
0424         ell_needed = ThisCT%l
0425         do ell = 1, ThisCT%ls
0426             if (ThisCT%ls%l(e
0427                 ThisCT%limber
0428                 ell_needed =
0429                 max_bessels_l
0430                 if (FeedbackL
0431                     exit
0432             end if
0433         end do
0434
0435         if (ThisCT%limber_l_m
0436             if (i==0) then
0437                 n1 = Ranges_I
0438                 n2 = TimeStep
0439             end if
0440
0441             do ell = ThisCT%l
0442                 LimbRec => Th
0443                 LimbRec%n1 =
0444                 LimbRec%n2 =
0445
0446                 allocate(Limb
0447                 allocate(Limb
0448
```

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

```
0451      int = 0
0452      do n = n1,n2
0453          chi = (CP
0454              k = (This
0455              LimbRec%k
0456              if (k<=qm
0457                  klo =
0458                  khi=k
0459                  ho=Ev
0460                  a0=(E
0461                  b0=(k
0462                  ho2o6
0463                  a03=(
0464                  b03=(
0465
0466                  LimbR
0467                  b0*Sr
0468              else
0469                  LimbR
0470              end if
0471          end do
0472      end do
0473      else
0474          max_bessels_l_ind
0475      end if
0476  end do
0477
0478  end subroutine GetLimberT
0479
0480  subroutine SourceToTransf
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 449

```
0449      int = 0
0450      do n = n1,n2
0451          chi = (CP
0452              k = (This
0453              LimbRec%k
0454              if (k<=qm
0455                  klo =
0456                  khi=k
0457                  ho=Ev
0458                  a0=(E
0459                  b0=(k
0460                  ho2o6
0461                  a03=(
0462                  b03=(
0463
0464                  LimbR
0465                  b
0466              else
0467                  LimbR
0468              end if
0469          end do
0470      end do
0471      else
0472          max_bessels_l_ind
0473      end if
0474  end do
0475
0476  end subroutine GetLimberT
0477
0478  subroutine SourceToTransf
```


/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 481

```
0481      integer q_ix
0482      type(IntegrationVars) ::
0483
0484      allocate(IV%Source_q(Time
0485      if (.not.CP%flat) allocat
0486
0487      call IntegrationVars_init
0488
0489      IV%q_ix = q_ix
0490      IV%q = ThisCT%q%points(q_i
0491      IV%dq= ThisCT%q%dpoints(q
0492
0493      call InterpolateSources(I
0494
0495      call DoSourceIntegration(
0496
0497      if (.not.CP%flat) dealloc
0498      deallocate(IV%Source_q)
0499
0500      end subroutine SourceToTr
0501
0502
0503      subroutine InitTransfer
0504      integer nu,lastnu, ntodo,
0505      real(dl) dlog_lowk1,dlog_
0506      real(dl) amin,q_switch_lo
0507      real(dl), dimension(:), a
0508
0509      if (CP%Transfer%k_per_log
0510          !Optimized spacing
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 479

```
0479      integer q_ix
0480      type(IntegrationVars) ::
0481
0482      allocate(IV%Source_q(Time
0483      if (.not.CP%flat) allocat
0484
0485      call IntegrationVars_init
0486
0487      IV%q_ix = q_ix
0488      IV%q = ThisCT%q%points(q_i
0489      IV%dq= ThisCT%q%dpoints(q
0490
0491      call InterpolateSources(I
0492
0493      call DoSourceIntegration(
0494
0495      if (.not.CP%flat) dealloc
0496      deallocate(IV%Source_q)
0497
0498      end subroutine SourceToTr
0499
0500
0501      subroutine InitTransfer
0502      integer nu,lastnu, ntodo,
0503      real(dl) dlog_lowk1,dlog_
0504      real(dl) amin,q_switch_lo
0505      real(dl), dimension(:), a
0506
0507      if (CP%Transfer%k_per_log
0508          !Optimized spacing
```

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

```
0511      !Large log spacing on
0512      !Linear spacing for h
0513      !Log spacing for last
0514      !large log spacing fo
0515
0516      boost = AccuracyBoost
0517      !#SimData
0518      if (CP%Transfer%high_
0519      !#SimData
0520
0521      q_switch_lowk1 = 0.7/
0522      dlog_lowk1=2*boost
0523
0524      q_switch_lowk = 8/tau
0525      dlog_lowk=8*boost
0526      if (HighAccuracyDefau
0527
0528      q_switch_osc = min(CP
0529      d_osc= 200*boost
0530      if (HighAccuracyDefau
0531
0532      q_switch_highk = min(
0533      dlog_osc = 17*boost
0534      if (HighAccuracyDefau
0535
0536      !Then up to kmax
0537      dlog_highk = 3*boost
0538
0539      amin = 5e-5_dl
0540
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 509

```
0509      !Large log spacing on
0510      !Linear spacing for h
0511      !Log spacing for last
0512      !large log spacing fo
0513
0514      boost = AccuracyBoost
0515      if (CP%Transfer%high_
0516
0517      q_switch_lowk1 = 0.7/
0518      dlog_lowk1=2*boost
0519
0520      q_switch_lowk = 8/tau
0521      dlog_lowk=8*boost
0522      if (HighAccuracyDefau
0523
0524      q_switch_osc = min(CP
0525      d_osc= 200*boost
0526      if (HighAccuracyDefau
0527
0528      q_switch_highk = min(
0529      dlog_osc = 17*boost
0530      if (HighAccuracyDefau
0531
0532      !Then up to kmax
0533      dlog_highk = 3*boost
0534
0535      amin = 5e-5_dl
0536
```

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

```
0541      nq=int((log(CP%Transf
0542      allocate(q_transfer(n
0543
0544      nq=int((log(q_switch_
0545      do q_ix=1, nq
0546          q_transfer(q_ix)
0547      end do
0548      MT%num_q_trans = nq
0549
0550      nq=int(log( q_switch_
0551      do q_ix=1, nq
0552          q_transfer(MT%num
0553      end do
0554      MT%num_q_trans = MT%n
0555
0556      nq=int((q_switch_osc-
0557      do q_ix=1, nq
0558          q_transfer(MT%num
0559      end do
0560      MT%num_q_trans = MT%n
0561
0562      if (CP%Transfer%kmax
0563          nq=int(log( q_swi
0564          do q_ix=1, nq
0565              q_transfer(MT
0566          end do
0567          MT%num_q_trans =
0568      end if
0569
0570      if (CP%Transfer%kmax
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 537

```
0537      nq=int((log(CP%Transf
0538      allocate(q_transfer(n
0539
0540      nq=int((log(q_switch_
0541      do q_ix=1, nq
0542          q_transfer(q_ix)
0543      end do
0544      MT%num_q_trans = nq
0545
0546      nq=int(log( q_switch_
0547      do q_ix=1, nq
0548          q_transfer(MT%num
0549      end do
0550      MT%num_q_trans = MT%n
0551
0552      nq=int((q_switch_osc-
0553      do q_ix=1, nq
0554          q_transfer(MT%num
0555      end do
0556      MT%num_q_trans = MT%n
0557
0558      if (CP%Transfer%kmax
0559          nq=int(log( q_swi
0560          do q_ix=1, nq
0561              q_transfer(MT
0562          end do
0563          MT%num_q_trans =
0564      end if
0565
0566      if (CP%Transfer%kmax
```

/Users/lplopa/Compare/camb_simdata/cmbmain.f90, Top line: 571

```
0571      nq=int(log(CP%Tra
0572      do q_ix=1, nq
0573          q_transfer(MT
0574      end do
0575      MT%num_q_trans =
0576      end if
0577  else
0578      !Fixed spacing
0579      MT%num_q_trans=int((1
0580      allocate(q_transfer(M
0581      do q_ix=1, MT%num_q_t
0582          q_transfer(q_ix)
0583      end do
0584  end if
0585
0586  if (CP%closed) then
0587      lastnu=0
0588      ntodo = 0
0589      do q_ix=1,MT%num_q_tr
0590          nu =nint(CP%r*q_t
0591          if (.not. ((nu<3)
0592              ntodo=ntodo+1
0593              q_transfer(nt
0594              lastnu=nu
0595          end if
0596      end do
0597      MT%num_q_trans = ntod
0598  end if
0599
0600  if (CP%WantCls) then
```

/Users/lplopa/Compare/camb_des/cmbmain.f90, Top line: 567

```
0567      nq=int(log(CP%Tra
0568      do q_ix=1, nq
0569          q_transfer(MT
0570      end do
0571      MT%num_q_trans =
0572      end if
0573  else
0574      !Fixed spacing
0575      MT%num_q_trans=int((1
0576      allocate(q_transfer(M
0577      do q_ix=1, MT%num_q_t
0578          q_transfer(q_ix)
0579      end do
0580  end if
0581
0582  if (CP%closed) then
0583      lastnu=0
0584      ntodo = 0
0585      do q_ix=1,MT%num_q_tr
0586          nu =nint(CP%r*q_t
0587          if (.not. ((nu<3)
0588              ntodo=ntodo+1
0589              q_transfer(nt
0590              lastnu=nu
0591          end if
0592      end do
0593      MT%num_q_trans = ntod
0594  end if
0595
0596  if (CP%WantCls) then
```

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

```
0601      ntodo = MT%num_q_tran
0602      first_i = ntodo+1
0603      do q_ix = 1,ntodo
0604          if (q_transfer(q
0605              !Feb13 fix fo
0606              ! if (q_tra
0607              first_i=q_ix
0608              exit
0609          end if
0610      end do
0611
0612      if (first_i > ntodo)
0613          MT%num_q_trans =
0614      else
0615          MT%num_q_trans =
0616      end if
0617      call Transfer_Allocat
0618
0619      MT%q_trans(1:Evolve_q
0620      if (MT%num_q_trans >
0621          MT%q_trans(Evolve
0622      end if
0623  else
0624      Evolve_q%npoints = 0
0625      call Transfer_Allocat
0626      MT%q_trans = q_transf
0627  end if
0628
0629      deallocate(q_transfer)
0630
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 597

```
0597      ntodo = MT%num_q_tran
0598      first_i = ntodo+1
0599      do q_ix = 1,ntodo
0600          if (q_transfer(q
0601              !Feb13 fix fo
0602              ! if (q_tra
0603              first_i=q_ix
0604              exit
0605          end if
0606      end do
0607
0608      if (first_i > ntodo)
0609          MT%num_q_trans =
0610      else
0611          MT%num_q_trans =
0612      end if
0613      call Transfer_Allocat
0614
0615      MT%q_trans(1:Evolve_q
0616      if (MT%num_q_trans >
0617          MT%q_trans(Evolve
0618      end if
0619  else
0620      Evolve_q%npoints = 0
0621      call Transfer_Allocat
0622      MT%q_trans = q_transf
0623  end if
0624
0625      deallocate(q_transfer)
0626
```

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

```
0631      end subroutine InitTransf
0632
0633      function GetTauStart(q)
0634      real(dl), intent(IN) :: q
0635      real(dl) taustart, GetTau
0636
0637      !      Begin when wave is
0638      !      Conformal time (in
0639      !      of relativistic neu
0640      if (CP%flat) then
0641          taustart=0.001_dl/q
0642      else
0643          taustart=0.001_dl/sqr
0644      end if
0645
0646      if (fixq/=0._dl) then
0647          taustart = 0.001_dl/f
0648      end if
0649
0650      !      Make sure to start
0651      taustart=min(taustart,0.1
0652
0653      !      Start when massive
0654      if (CP%Num_nu_massive>0)
0655          taustart=min(taustart
0656      end if
0657
0658      GetTauStart=taustart
0659      end function GetTauStart
0660
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 627

```
0627      end subroutine InitTransf
0628
0629      function GetTauStart(q)
0630      real(dl), intent(IN) :: q
0631      real(dl) taustart, GetTau
0632
0633      !      Begin when wave is
0634      !      Conformal time (in
0635      !      of relativistic neu
0636      if (CP%flat) then
0637          taustart=0.001_dl/q
0638      else
0639          taustart=0.001_dl/sqr
0640      end if
0641
0642      if (fixq/=0._dl) then
0643          taustart = 0.001_dl/f
0644      end if
0645
0646      !      Make sure to start
0647      taustart=min(taustart,0.1
0648
0649      !      Start when massive
0650      if (CP%Num_nu_massive>0)
0651          taustart=min(taustart
0652      end if
0653
0654      GetTauStart=taustart
0655      end function GetTauStart
0656
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 661

```
0661      subroutine DoSourcek(EV,q
0662      integer q_ix
0663      real(dl) taustart
0664      type(EvolutionVars) EV
0665
0666      EV%q=Evolve_q%points(q_ix
0667
0668      if (fixq/=0._dl) then
0669          EV%q= min(500._dl,fix
0670      end if
0671      EV%q2=EV%q**2
0672
0673      EV%q_ix = q_ix
0674      EV%TransferOnly=.false.
0675
0676      taustart = GetTauStart(EV
0677
0678      if (fixq/=0._dl) then
0679          EV%q= fixq
0680          EV%q2=EV%q**2
0681      end if
0682
0683      call GetNumEqns(EV)
0684
0685      if (CP%WantScalars .and.
0686      if (CP%WantVectors .and.
0687      if (CP%WantTensors .and.
0688
0689      end subroutine DoSourcek
0690
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 657

```
0657      subroutine DoSourcek(EV,q
0658      integer q_ix
0659      real(dl) taustart
0660      type(EvolutionVars) EV
0661
0662      EV%q=Evolve_q%points(q_ix
0663
0664      if (fixq/=0._dl) then
0665          EV%q= min(500._dl,fix
0666      end if
0667      EV%q2=EV%q**2
0668
0669      EV%q_ix = q_ix
0670      EV%TransferOnly=.false.
0671
0672      taustart = GetTauStart(EV
0673
0674      if (fixq/=0._dl) then
0675          EV%q= fixq
0676          EV%q2=EV%q**2
0677      end if
0678
0679      call GetNumEqns(EV)
0680
0681      if (CP%WantScalars .and.
0682      if (CP%WantVectors .and.
0683      if (CP%WantTensors .and.
0684
0685      end subroutine DoSourcek
0686
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 691

```
0691      subroutine GetSourceMem
0692
0693      if (CP%WantScalars) then
0694          if (WantLateTime) the
0695              SourceNum=3
0696              C_last = C_PhiE
0697              SourceNum=SourceN
0698
0699          else
0700              SourceNum=2
0701              C_last = C_Cross
0702          end if
0703      else
0704          SourceNum=3
0705      end if
0706
0707      allocate(Src(Evolve_q%npo
0708      Src=0
0709      allocate(ddSrc(Evolve_q%n
0710
0711      end subroutine GetSourceM
0712
0713      subroutine FreeSourceMem
0714
0715      if (allocated(Src))deallo
0716      call Ranges_Free(Evolve_q
0717
0718      end subroutine FreeSource
0719
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 687

```
0687      subroutine GetSourceMem
0688
0689      if (CP%WantScalars) then
0690          if (WantLateTime) the
0691              SourceNum=3
0692              C_last = C_PhiE
0693              SourceNum=SourceN
0694              num_extra_red
0695
0696          else
0697              SourceNum=2
0698              C_last = C_Cross
0699          end if
0700      else
0701          SourceNum=3
0702      end if
0703
0704      allocate(Src(Evolve_q%npo
0705      Src=0
0706      allocate(ddSrc(Evolve_q%n
0707
0708      end subroutine GetSourceM
0709
0710      subroutine FreeSourceMem
0711
0712      if (allocated(Src))deallo
0713      call Ranges_Free(Evolve_q
0714
0715      end subroutine FreeSource
0716
```


/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 720

```
0720
0721      ! initial variables, num
0722      subroutine InitVars
0723      use ThermoData
0724      use precision
0725      use ModelParams
0726
0727      implicit none
0728      real(dl) taumin, maxq, in
0729      integer itf
0730
0731      initAccuracyBoost = Accur
0732
0733      ! Maximum and minimum k-v
0734      if (CP%flat) then
0735          qmax=maximum_qeta/CP%
0736          qmin=qmin0/CP%tau0/in
0737      else
0738          qmax=maximum_qeta/CP%
0739          qmin=qmin0/CP%r/CP%ch
0740      end if
0741      !      Timesteps during re
0742      !      timestep is the min
0743      !      where taurst is the
0744
0745      dtaurec_q=4/qmax/initAccu
0746      if (.not. CP%flat) dtaure
0747      !AL:Changed Dec 2003, dta
0748      dtaurec = dtaurec_q
0749      !dtau rec may be changed
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 717

```
0717
0718      ! initial variables, num
0719      subroutine InitVars
0720      use ThermoData
0721      use precision
0722      use ModelParams
0723
0724      implicit none
0725      real(dl) taumin, maxq, in
0726      integer itf
0727
0728      initAccuracyBoost = Accur
0729
0730      ! Maximum and minimum k-v
0731      if (CP%flat) then
0732          qmax=maximum_qeta/CP%
0733          qmin=qmin0/CP%tau0/in
0734      else
0735          qmax=maximum_qeta/CP%
0736          qmin=qmin0/CP%r/CP%ch
0737      end if
0738      !      Timesteps during re
0739      !      timestep is the min
0740      !      where taurst is the
0741
0742      dtaurec_q=4/qmax/initAccu
0743      if (.not. CP%flat) dtaure
0744      !AL:Changed Dec 2003, dta
0745      dtaurec = dtaurec_q
0746      !dtau rec may be changed
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 750

```
0750
0751     max_etak_tensor = initAcc
0752     max_etak_scalar = initAcc
0753     if (maximum_qeta < 3500 .a
0754     !tweak to get large scale
0755     max_etak_vector = max_eta
0756
0757     if (CP%WantCls) then
0758         maxq = qmax
0759         if (CP%WantTransfer)
0760     else
0761         maxq = CP%Transfer%kmax
0762     end if
0763
0764
0765     taumin = GetTauStart(maxq)
0766
0767     !      Initialize baryon t
0768     !      This subroutine als
0769     !      saved in order to d
0770     !These routines in Thermo
0771     call inithermo(taumin, CP%
0772     if (global_error_flag /= 0)
0773
0774     if (DebugMsgs .and. Feedb
0775
0776     !Do any array initializat
0777     call GaugeInterface_Init
0778
0779     if (Feedbacklevel > 0) &
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 747

```
0747
0748     max_etak_tensor = initAcc
0749     max_etak_scalar = initAcc
0750     if (maximum_qeta < 3500 .a
0751     !tweak to get large scale
0752     max_etak_vector = max_eta
0753
0754     if (CP%WantCls) then
0755         maxq = qmax
0756         if (CP%WantTransfer)
0757     else
0758         maxq = CP%Transfer%kmax
0759     end if
0760
0761
0762     taumin = GetTauStart(maxq)
0763
0764     !      Initialize baryon t
0765     !      This subroutine als
0766     !      saved in order to d
0767     !These routines in Thermo
0768     call inithermo(taumin, CP%
0769     if (global_error_flag /= 0)
0770
0771     if (DebugMsgs .and. Feedb
0772
0773     !Do any array initializat
0774     call GaugeInterface_Init
0775
0776     if (Feedbacklevel > 0) &
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 780

```
0780 write(*, ' ("tau_recomb/Mpc
0781
0782 !      Calculating the tim
0783 !
0784 if (CP%WantTransfer) then
0785     do itf=1,CP%Transfer%
0786         tautf(itf)=min(Ti
0787         if (itf>1) then
0788             if (tautf(itf
0789             stop 'Tra
0790             end if
0791         end if
0792     end do
0793 endif
0794
0795 end subroutine InitVars
0796
0797 subroutine SetkValuesForS
0798 implicit none
0799 real(dl) dlnk0, dkn1, dkn
0800 real(dl) qmax_log
0801 real(dl) SourceAccuracyBo
0802 !      set k values for wh
0803 !      polarization will b
0804 !      use a logarithmic s
0805
0806 SourceAccuracyBoost = Acc
0807 if (CP%WantScalars .and.
0808     dlnk0=2._dl/10/Source
0809     !Need this to get acc
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 777

```
0777 write(*, ' ("tau_recomb
0778
0779 !      Calculating the tim
0780 !
0781 if (CP%WantTransfer) then
0782     do itf=1,CP%Transfer%
0783         tautf(itf)=min(Ti
0784         if (itf>1) then
0785             if (tautf(itf
0786             call MpiS
0787             end if
0788         end if
0789     end do
0790 endif
0791
0792 end subroutine InitVars
0793
0794 subroutine SetkValuesForS
0795 implicit none
0796 real(dl) dlnk0, dkn1, dkn
0797 real(dl) qmax_log
0798 real(dl) SourceAccuracyBo
0799 !      set k values for wh
0800 !      polarization will b
0801 !      use a logarithmic s
0802
0803 SourceAccuracyBoost = Acc
0804 if (CP%WantScalars .and.
0805     dlnk0=2._dl/10/Source
0806     !Need this to get acc
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 810

```
0810      else
0811          dlnk0=5._dl/10/Source
0812          if (CP%closed) dlnk0=
0813      end if
0814
0815      if (CP%AccurateReionizati
0816
0817      dkn1=0.6_dl/taurst/Source
0818      dkn2=0.9_dl/taurst/Source
0819      if (HighAccuracyDefault)
0820      if (CP%WantTensors .or. C
0821          dkn1=dkn1 *0.8_dl
0822          dlnk0=dlnk0/2 !*0.3_d
0823          dkn2=dkn2*0.85_dl
0824      end if
0825
0826      qmax_log = dkn1/dlnk0
0827      q_switch = 2*6.3/taurst
0828      !Want linear spacing for
0829      !Could use sound horizon,
0830
0831      q_cmb = 2*1_smooth_sample
0832      if (CP%Want_CMB .and. max
0833      !prevent EE going wild in
0834      dksmooth = q_cmb/2/(Accur
0835      if (CP%Want_CMB) dksmooth
0836
0837      call Ranges_Init(Evolve_q
0838      call Ranges_Add_delta(Evo
0839      call Ranges_Add_delta(Evo
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 807

```
0807      else
0808          dlnk0=5._dl/10/Source
0809          if (CP%closed) dlnk0=
0810      end if
0811
0812      if (CP%AccurateReionizati
0813
0814      dkn1=0.6_dl/taurst/Source
0815      dkn2=0.9_dl/taurst/Source
0816      if (HighAccuracyDefault)
0817      if (CP%WantTensors .or. C
0818          dkn1=dkn1 *0.8_dl
0819          dlnk0=dlnk0/2 !*0.3_d
0820          dkn2=dkn2*0.85_dl
0821      end if
0822
0823      qmax_log = dkn1/dlnk0
0824      q_switch = 2*6.3/taurst
0825      !Want linear spacing for
0826      !Could use sound horizon,
0827
0828      q_cmb = 2*1_smooth_sample
0829      if (CP%Want_CMB .and. max
0830      !prevent EE going wild in
0831      dksmooth = q_cmb/2/(Accur
0832      if (CP%Want_CMB) dksmooth
0833
0834      call Ranges_Init(Evolve_q
0835      call Ranges_Add_delta(Evo
0836      call Ranges_Add_delta(Evo
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 840

```
0840      if (qmax > q_switch) then
0841          call Ranges_Add_delta
0842          if (qmax > q_cmb) then
0843              dksmooth = log(1
0844              call Ranges_Add_d
0845          end if
0846      end if
0847
0848      call Ranges_GetArray(Evol
0849
0850      if (CP%closed) call SetCl
0851
0852      end subroutine SetkValues
0853
0854
0855      subroutine SetClosedkValu
0856      Type(Regions) :: R
0857      integer i,nu,lastnu,nmax
0858      !nu = 3,4,5... in CP%clos
0859      logical, intent(in) :: fo
0860      integer ix
0861      real(dl) dnu
0862      integer, allocatable :: n
0863
0864      if (forInt .and. nint(R%p
0865          !quantization is impo
0866          call Ranges_Getdpoint
0867          R%dpoints = max(1,int
0868          lastnu=2
0869          ix=1
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 837

```
0837      if (qmax > q_switch) then
0838          call Ranges_Add_delta
0839          if (qmax > q_cmb) then
0840              dksmooth = log(1
0841              call Ranges_Add_d
0842          end if
0843      end if
0844
0845      call Ranges_GetArray(Evol
0846
0847      if (CP%closed) call SetCl
0848
0849      end subroutine SetkValues
0850
0851
0852      subroutine SetClosedkValu
0853      Type(Regions) :: R
0854      integer i,nu,lastnu,nmax
0855      !nu = 3,4,5... in CP%clos
0856      logical, intent(in) :: fo
0857      integer ix
0858      real(dl) dnu
0859      integer, allocatable :: n
0860
0861      if (forInt .and. nint(R%p
0862          !quantization is impo
0863          call Ranges_Getdpoint
0864          R%dpoints = max(1,int
0865          lastnu=2
0866          ix=1
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 870

```
0870      dnu =R%dpoints(ix)
0871      nmax=0
0872      lastnu=2
0873      allocate(nu_array(R%n
0874      do
0875          do while (R%dpoin
0876              ix=ix+1
0877          end do
0878          do nu=lastnu+1,ni
0879              nmax=nmax+1
0880              nu_array(nmax
0881          end do
0882          lastnu=nu_array(n
0883          if (ix==R%npoints
0884              dnu = R%dpoints(i
0885      end do
0886      if (nint(R%points(R%n
0887          nmax=nmax+1
0888          nu_array(nmax) =
0889      end if
0890      deallocate(R%points)
0891      allocate(R%points(nma
0892      R%points = nu_array(1
0893      deallocate(nu_array)
0894  else
0895      lastnu=3
0896      nmax=1
0897
0898      do i=2,R%npoints
0899          nu=nint(R%points(
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 867

```
0867      dnu =R%dpoints(ix)
0868      nmax=0
0869      lastnu=2
0870      allocate(nu_array(R%n
0871      do
0872          do while (R%dpoin
0873              ix=ix+1
0874          end do
0875          do nu=lastnu+1,ni
0876              nmax=nmax+1
0877              nu_array(nmax
0878          end do
0879          lastnu=nu_array(n
0880          if (ix==R%npoints
0881              dnu = R%dpoints(i
0882      end do
0883      if (nint(R%points(R%n
0884          nmax=nmax+1
0885          nu_array(nmax) =
0886      end if
0887      deallocate(R%points)
0888      allocate(R%points(nma
0889      R%points = nu_array(1
0890      deallocate(nu_array)
0891  else
0892      lastnu=3
0893      nmax=1
0894
0895      do i=2,R%npoints
0896          nu=nint(R%points(
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 900

```
0900          if (nu > lastnu)
0901              nmax=nmax+1
0902              lastnu=nu
0903              R%points(nmax
0904          end if
0905      end do
0906      R%points(1)=3/CP%r
0907  end if
0908
0909      R%Lowest = R%points(1)
0910      R%Highest = R%points(nmax
0911      R%npoints=nmax
0912
0913  end subroutine SetClosedk
0914
0915
0916
0917  subroutine CalcScalarSour
0918  use Transfer
0919  implicit none
0920  type(EvolutionVars) EV
0921  real(dl) tau,tol1,tauend,
0922  integer j,ind,itf
0923  real(dl) c(24),w(EV%nvar,
0924
0925      real(dl) yprime(EV%nvar),
0926      external dtauda
0927
0928  if (fixq/=0._dl) then
0929      !evolution output
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 897

```
0897          if (nu > lastnu)
0898              nmax=nmax+1
0899              lastnu=nu
0900              R%points(nmax
0901          end if
0902      end do
0903      R%points(1)=3/CP%r
0904  end if
0905
0906      R%Lowest = R%points(1)
0907      R%Highest = R%points(nmax
0908      R%npoints=nmax
0909
0910  end subroutine SetClosedk
0911
0912
0913
0914  subroutine CalcScalarSour
0915  use Transfer
0916  implicit none
0917  type(EvolutionVars) EV
0918  real(dl) tau,tol1,tauend,
0919  integer j,ind,itf
0920  real(dl) c(24),w(EV%nvar,
0921
0922      real(dl) yprime(EV%nvar),
0923      external dtauda
0924
0925  if (fixq/=0._dl) then
0926      !evolution output
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 930

```
0930      EV%q=fixq
0931      EV%q2=EV%q**2
0932      endif
0933
0934      w=0
0935      y=0
0936      call initial(EV,y, tausta
0937      if (global_error_flag/=0)
0938
0939      tau=taustart
0940      ind=1
0941
0942      !!Example code for plotti
0943      if (fixq/=0._dl) then
0944          toll=tol/exp(Accuracy
0945          call CreateTxtFile('e
0946          do j=1,1000
0947              tauend = taustart
0948              call GaugeInterfa
0949              yprime = 0
0950              call derivs(EV,EV
0951              adotoa = 1/(y(1)*
0952              ddelta= (yprime(3
0953              delta=(grhoc*y(3)
0954              growth= ddelta/de
0955              write (1,' (7E15.5
0956          end do
0957          close(1)
0958          stop
0959      end if
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 927

```
0927      EV%q=fixq
0928      EV%q2=EV%q**2
0929      endif
0930
0931      w=0
0932      y=0
0933      call initial(EV,y, tausta
0934      if (global_error_flag/=0)
0935
0936      tau=taustart
0937      ind=1
0938
0939      !!Example code for plotti
0940      if (fixq/=0._dl) then
0941          toll=tol/exp(Accuracy
0942          call CreateTxtFile('e
0943          do j=1,1000
0944              tauend = taustart
0945              call GaugeInterfa
0946              yprime = 0
0947              call derivs(EV,EV
0948              adotoa = 1/(y(1)*
0949              ddelta= (yprime(3
0950              delta=(grhoc*y(3)
0951              growth= ddelta/de
0952              write (1,' (7E15.5
0953          end do
0954          close(1)
0955          stop
0956      end if
```


/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 960

```
0960
0961      !      Begin timestep loop
0962
0963      itf=1
0964      toll=tol/exp(AccuracyBoos
0965      if (CP%WantTransfer .and.
0966
0967      do j=2,TimeSteps%npoints
0968          tauend=TimeSteps%poin
0969
0970          if (.not. DebugEvolut
0971              .and. .not. WantLateT
0972              Src(EV%q_ix,1:Sou
0973          else
0974              !Integrate over t
0975              call GaugeInterfa
0976              if (global_error_
0977
0978              call output(EV,y,
0979              Src(EV%q_ix,1:Sou
0980
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 957

```
0957
0958      !      Begin timestep loop
0959
0960      itf=1
0961      toll=tol/exp(AccuracyBoos
0962      if (CP%WantTransfer) then
0963          if (CP%Transfer%high
0964              do while (itf <= CP%T
0965                  !Just in case som
0966                  call GaugeInterfa
0967                  if (global_error_
0968                  call outtransf(EV
0969                  itf = itf+1
0970              end do
0971          end if
0972
0973      do j=2,TimeSteps%npoints
0974          tauend=TimeSteps%poin
0975
0976          if (.not. DebugEvolut
0977              .and. .not. WantL
0978              Src(EV%q_ix,1:Sou
0979          else
0980              !Integrate over t
0981              call GaugeInterfa
0982              if (global_error_
0983
0984              call output(EV,y,
0985              Src(EV%q_ix,1:Sou
0986
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 981

```
0981      !      Calculation
0982      101      if (CP%WantTransf
0983              if (j < Times
0984                  if (tauen
0985                      call
0986                      if (g
0987                      endif
0988                  end if
0989              !      output
0990
0991              if (abs(tau-t
0992                  call outt
0993
0994                  itf=itf+1
0995                  if (j < T
0996                      if (i
0997                          Times
0998                      end if
0999                  endif
1000              end if
1001          end if
1002
1003      end do !time step loop
1004
1005      end subroutine
1006
1007
1008      subroutine CalcTensorSour
1009      implicit none
1010      type(EvolutionVars) EV
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 987

```
0987      !      Calculation
0988      101      if (CP%WantTransf
0989              if (j < Times
0990                  if (tauen
0991                      call
0992                      if (g
0993                      endif
0994                  end if
0995              !      output
0996
0997              if (abs(tau-t
0998                  call outt
0999
1000                  itf=itf+1
1001                  if (j < T
1002                      if (i
1003                          T
1004                      end if
1005                  endif
1006              end if
1007          end if
1008
1009      end do !time step loop
1010
1011      end subroutine
1012
1013
1014      subroutine CalcTensorSour
1015      implicit none
1016      type(EvolutionVars) EV
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1011

```
1011      real(dl) tau,tol1,tauend,
1012      integer j,ind
1013      real(dl) c(24),wt(EV%nvar
1014
1015      call initialt(EV,yt, taus
1016
1017      tau=taustart
1018      ind=1
1019      tol1=tol/exp(AccuracyBoos
1020
1021      !      Begin timestep loop
1022      do j=2,TimeSteps%npoints
1023          tauend=TimeSteps%poin
1024          if (EV%q*tauend > max
1025              Src(EV%q_ix,1:Sou
1026          else
1027              call GaugeInterfa
1028
1029              call outputt(EV,y
1030              Src(EV%q_ix,CT_E,
1031          end if
1032      end do
1033
1034      end subroutine CalcTensor
1035
1036
1037      subroutine CalcVectorSour
1038
1039      implicit none
1040      type(EvolutionVars) EV
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1017

```
1017      real(dl) tau,tol1,tauend,
1018      integer j,ind
1019      real(dl) c(24),wt(EV%nvar
1020
1021      call initialt(EV,yt, taus
1022
1023      tau=taustart
1024      ind=1
1025      tol1=tol/exp(AccuracyBoos
1026
1027      !      Begin timestep loop
1028      do j=2,TimeSteps%npoints
1029          tauend=TimeSteps%poin
1030          if (EV%q*tauend > max
1031              Src(EV%q_ix,1:Sou
1032          else
1033              call GaugeInterfa
1034
1035              call outputt(EV,y
1036              Src(EV%q_ix,C
1037          end if
1038      end do
1039
1040      end subroutine CalcTensor
1041
1042
1043      subroutine CalcVectorSour
1044
1045      implicit none
1046      type(EvolutionVars) EV
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1041

```
1041      real(dl) tau,tol1,tauend,
1042      integer j,ind
1043      real(dl) c(24),wt(EV%nvar
1044
1045      !EV%q=0.2
1046      !EV%q2=EV%q**2
1047
1048      call initialv(EV,yv, taus
1049
1050      tau=taustart
1051      ind=1
1052      tol1=tol*0.01/exp(Accurac
1053
1054      !!Example code for plotti
1055      !if (.false.) then
1056      !      do j=1,6000
1057      !          tauend = taust
1058      !          call dverk(EV,E
1059      !          call fderivsv(
1060      !
1061      !          write (*,'(7E1
1062      !                  yv((E
1063      !                  yv((EV%l
1064      !          end do
1065      !      stop
1066      !nd if
1067
1068      !      Begin timestep loop
1069      do j=2,TimeSteps%npoints
1070          tauend=TimeSteps%poin
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1047

```
1047      real(dl) tau,tol1,tauend,
1048      integer j,ind
1049      real(dl) c(24),wt(EV%nvar
1050
1051      !EV%q=0.2
1052      !EV%q2=EV%q**2
1053
1054      call initialv(EV,yv, taus
1055
1056      tau=taustart
1057      ind=1
1058      tol1=tol*0.01/exp(Accurac
1059
1060      !!Example code for plotti
1061      !if (.false.) then
1062      !      do j=1,6000
1063      !          tauend = taust
1064      !          call dverk(EV,E
1065      !          call fderivsv(
1066      !
1067      !          write (*,'(7E1
1068      !                  yv((E
1069      !                  yv((EV%l
1070      !          end do
1071      !      stop
1072      !nd if
1073
1074      !      Begin timestep loop
1075      do j=2,TimeSteps%npoints
1076          tauend=TimeSteps%poin
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1071

```
1071
1072         if ( EV%q*tauend > ma
1073             Src(EV%q_ix,1:Sou
1074         else
1075             call dverk(EV,EV%
1076
1077             call outputv(EV,y
1078             Src(EV%q_ix,CT_E,
1079         end if
1080     end do
1081
1082     end subroutine CalcVector
1083
1084
1085     subroutine TransferOut
1086     !Output transfer function
1087     implicit none
1088     integer q_ix
1089     real(dl) tau
1090     type(EvolutionVars) EV
1091
1092
1093     if (DebugMsgs .and. Feedb
1094     write(*,*) MT%num_q_trans
1095
1096     !$OMP PARALLEL DO DEFAULT
1097     !$OMP & PRIVATE(EV, tau,
1098
1099     !      loop over wavenumbe
1100     do q_ix=Evolve_q%npoints+
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1077

```
1077
1078         if ( EV%q*tauend > ma
1079             Src(EV%q_ix,1:Sou
1080         else
1081             call dverk(EV,EV%
1082
1083             call outputv(EV,y
1084             Src(EV%q_ix,C
1085         end if
1086     end do
1087
1088     end subroutine CalcVector
1089
1090
1091     subroutine TransferOut
1092     !Output transfer function
1093     implicit none
1094     integer q_ix
1095     real(dl) tau
1096     type(EvolutionVars) EV
1097
1098
1099     if (DebugMsgs .and. Feedb
1100     write(*,*) MT%num_q_t
1101
1102     !      loop over wavenumbe
1103     !$OMP PARALLEL DO DEFAULT
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1101

```
1101      EV%TransferOnly=.true
1102
1103      EV%q= MT%q_trans(q_ix
1104
1105      EV%q2=EV%q**2
1106      EV%q_ix = q_ix
1107
1108      tau = GetTauStart(EV%
1109
1110      call GetNumEqns(EV)
1111
1112      call GetTransfer(EV,
1113  end do
1114  !$OMP END PARALLEL DO
1115
1116  end subroutine TransferOu
1117
1118  subroutine GetTransfer(EV
1119  type(EvolutionVars) EV
1120  real(dl) tau
1121  integer ind, i
1122  real(dl) c(24),w(EV%nvar,
1123  real(dl) atol
1124
1125  atol=tol/exp(AccuracyBoos
1126  if (CP%Transfer%high_prec
1127
1128  ind=1
1129  call initial(EV,y, tau)
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1104

```
1104      do q_ix=MT%num_q_trans, E
1105      EV%TransferOnly=.true
1106
1107      EV%q= MT%q_trans(q_ix
1108
1109      EV%q2=EV%q**2
1110      EV%q_ix = q_ix
1111
1112      tau = GetTauStart(EV%
1113
1114      call GetNumEqns(EV)
1115
1116      call GetTransfer(EV,
1117  end do
1118  !$OMP END PARALLEL DO
1119
1120  end subroutine TransferOu
1121
1122  subroutine GetTransfer(EV
1123  type(EvolutionVars) EV
1124  real(dl) tau
1125  integer ind, i
1126  real(dl) c(24),w(EV%nvar,
1127  real(dl) atol
1128
1129  atol=tol/exp(AccuracyBoos
1130  if (CP%Transfer%high_prec
1131
1132  ind=1
1133  call initial(EV,y, tau)
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1130

```
1130      if (global_error_flag/=0)
1131
1132      do i=1,CP%Transfer%num_re
1133          call GaugeInterface_E
1134          if (global_error_flag
1135              call outtransf(EV,y,t
1136      end do
1137
1138      end subroutine GetTransfe
1139
1140
1141      subroutine MakeNonlinearS
1142      !Scale lensing source ter
1143      use NonLinear
1144      integer i,ik,first_step
1145      real (dl) tau
1146      real(dl) scaling(CP%Trans
1147      real(dl) ho,a0,b0, ascale
1148      integer tf_lo, tf_hi
1149      type(MatterPowerData) ::
1150
1151      call Transfer_GetMatterPo
1152
1153      call NonLinear_GetNonLinR
1154
1155      if (CP%InitPower%nn > 1)
1156
1157      first_step=1
1158      do while(TimeSteps%points
1159          first_step = first_st
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1134

```
1134      if (global_error_flag/=0)
1135
1136      do i=1,CP%Transfer%num_re
1137          call GaugeInterface_E
1138          if (global_error_flag
1139              call outtransf(EV,y,t
1140      end do
1141
1142      end subroutine GetTransfe
1143
1144
1145      subroutine MakeNonlinearS
1146      !Scale lensing source ter
1147      use NonLinear
1148      integer i,ik,first_step
1149      real (dl) tau
1150      real(dl) scaling(CP%Trans
1151      real(dl) ho,a0,b0, ascale
1152      integer tf_lo, tf_hi
1153      type(MatterPowerData) ::
1154
1155      call Transfer_GetMatterPo
1156
1157      call NonLinear_GetNonLinR
1158
1159      if (CP%InitPower%nn > 1)
1160
1161      first_step=1
1162      do while(TimeSteps%points
1163          first_step = first_st
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1160

```
1160      end do
1161      !$OMP PARALLEL DO DEFAULT
1162      !$OMP & PRIVATE(ik, i, sca
1163      do ik=1, Evolve_q%npoints
1164          if (Evolve_q%points(i
1165              !Interpolate non-
1166              do i = 1, CP%Tran
1167                  scaling(i) =
1168              end do
1169              if (all(abs(scali
1170              call spline(tautf
1171              spl_large,spl_lar
1172
1173              tf_lo=1
1174              tf_hi=tf_lo+1
1175
1176              do i=first_step,T
1177                  tau = TimeSte
1178
1179                  do while (tau
1180                      tf_lo = t
1181                      tf_hi = t
1182                  end do
1183
1184                  ho=tautf(tf_h
1185                  a0=(tautf(tf_
1186                  b0=1-a0
1187
1188                  ascale = a0*s
1189                  ((a0**3-a0)*
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1164

```
1164      end do
1165      !$OMP PARALLEL DO DEFAULT
1166      !$OMP & PRIVATE(ik, i, sca
1167      do ik=1, Evolve_q%npoints
1168          if (Evolve_q%points(i
1169              !Interpolate non-
1170              do i = 1, CP%Tran
1171                  scaling(i) =
1172              end do
1173              if (all(abs(scali
1174              call spline(tautf
1175              spl_large,spl
1176
1177              tf_lo=1
1178              tf_hi=tf_lo+1
1179
1180              do i=first_step,T
1181                  tau = TimeSte
1182
1183                  do while (tau
1184                      tf_lo = t
1185                      tf_hi = t
1186                  end do
1187
1188                  ho=tautf(tf_h
1189                  a0=(tautf(tf_
1190                  b0=1-a0
1191
1192                  ascale = a0*s
1193                  ((a0**3-a0)*
```


/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1190

```
1190      +(b0**3-b0)*d
1191
1192      Src(ik,3:Sour
1193      end do
1194      end if
1195  end do
1196  !$OMP END PARALLEL DO
1197
1198  call MatterPowerdata_Free
1199
1200  end subroutine MakeNonlin
1201
1202
1203  subroutine InitSourceInte
1204  integer i,j
1205  !      get the interpolati
1206  !      for other k-values
1207  !$OMP PARALLEL DO DEFAULT
1208  do i=1,TimeSteps%npoints
1209      do j=1, SourceNum
1210          call spline(Evolv
1211      end do
1212  end do
1213  !$OMP END PARALLEL DO
1214  end subroutine InitSource
1215
1216
1217  subroutine SetkValuesForI
1218  implicit none
1219
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1194

```
1194      +(b0**3-b
1195
1196      Src(ik,3:Sour
1197      end do
1198      end if
1199  end do
1200  !$OMP END PARALLEL DO
1201
1202  call MatterPowerdata_Free
1203
1204  end subroutine MakeNonlin
1205
1206
1207  subroutine InitSourceInte
1208  integer i,j
1209  !      get the interpolati
1210  !      for other k-values
1211  !$OMP PARALLEL DO DEFAULT
1212  do i=1,TimeSteps%npoints
1213      do j=1, SourceNum
1214          call spline(Evolv
1215      end do
1216  end do
1217  !$OMP END PARALLEL DO
1218  end subroutine InitSource
1219
1220
1221  subroutine SetkValuesForI
1222  implicit none
1223
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1220

```
1220      integer no
1221      real(dl) dk,dk0,dlnk1, dk
1222      integer lognum
1223      real(dl)  qmax_int,IntSam
1224
1225
1226      qmax_int = min(qmax,max_b
1227
1228      IntSampleBoost=AccuracyBo
1229      if (do_bispectrum) then
1230          IntSampleBoost = IntS
1231          if (hard_bispectrum)
1232      end if
1233
1234      !      Fixing the # of k f
1235
1236      call Ranges_Init(ThisCT%q
1237
1238      if (CP%closed.and.ExactCl
1239          call Ranges_Add(ThisC
1240          call Init_ClTransfer(
1241          call Ranges_Getdpoint
1242      else
1243          !Split up into logari
1244          !then no-lognum*dk0 l
1245          !then at dk up to max
1246          lognum=nint(10*IntSam
1247          dlnk1=1._dl/lognum
1248          no=nint(600*IntSample
1249          dk0=1.8_dl/CP%r/CP%ch
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1224

```
1224      integer no
1225      real(dl) dk,dk0,dlnk1, dk
1226      integer lognum
1227      real(dl)  qmax_int,IntSam
1228
1229
1230      qmax_int = min(qmax,max_b
1231
1232      IntSampleBoost=AccuracyBo
1233      if (do_bispectrum) then
1234          IntSampleBoost = IntS
1235          if (hard_bispectrum)
1236      end if
1237
1238      !      Fixing the # of k f
1239
1240      call Ranges_Init(ThisCT%q
1241
1242      if (CP%closed.and.ExactCl
1243          call Ranges_Add(ThisC
1244          call Init_ClTransfer(
1245          call Ranges_Getdpoint
1246      else
1247          !Split up into logari
1248          !then no-lognum*dk0 l
1249          !then at dk up to max
1250          lognum=nint(10*IntSam
1251          dlnk1=1._dl/lognum
1252          no=nint(600*IntSample
1253          dk0=1.8_dl/CP%r/CP%ch
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1250

```
1250      dk=3._dl/CP%r/CP%chi0
1251
1252      if (HighAccuracyDefau
1253
1254      k_max_log = lognum*dk
1255      k_max_0   = no*dk0
1256
1257      if (do_bispectrum) k_
1258
1259      dk2 = 0.04/IntSampleB
1260
1261      call Ranges_Add_delta
1262      call Ranges_Add_delta
1263
1264      if (qmax_int > k_max
1265          max_k_dk = max(30
1266
1267          call Ranges_Add_d
1268          if (qmax_int > ma
1269              !This allows
1270              !without taki
1271              call Ranges_A
1272          end if
1273      end if
1274
1275      call Init_ClTransfer(
1276
1277      if (CP%closed) then
1278          call SetClosedkVa
1279          call Ranges_Getdp
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1254

```
1254      dk=3._dl/CP%r/CP%chi0
1255
1256      if (HighAccuracyDefau
1257
1258      k_max_log = lognum*dk
1259      k_max_0   = no*dk0
1260
1261      if (do_bispectrum) k_
1262
1263      dk2 = 0.04/IntSampleB
1264
1265      call Ranges_Add_delta
1266      call Ranges_Add_delta
1267
1268      if (qmax_int > k_max
1269          max_k_dk = max(30
1270
1271          call Ranges_Add_d
1272          if (qmax_int > ma
1273              !This allows
1274              !without taki
1275              call Ranges_A
1276          end if
1277      end if
1278
1279      call Init_ClTransfer(
1280
1281      if (CP%closed) then
1282          call SetClosedkVa
1283          call Ranges_Getdp
```

/Users/lplopa/Compare/camb_simdata/cmbmain.f90, Top line: 1280

```
1280      ThisCT%q%dpoints(  
1281      deallocate(ThisCT  
1282      allocate(ThisCT%D  
1283      ThisCT%Delta_p_l_  
1284      end if  
1285  
1286      end if !ExactClosedSum  
1287  
1288  
1289      end subroutine setkValues  
1290  
1291      subroutine InterpolateSou  
1292      implicit none  
1293      integer i,khi,klo, step  
1294      real(dl) xf,b0,ho,a0,ho2o  
1295      type(IntegrationVars) IV  
1296  
1297  
1298      !      finding position of  
1299  
1300      !Can't use the following  
1301      !      klo = min(Evo  
1302      !This is a bit inefficien  
1303      klo=1  
1304      do while ((IV%q > Evolve_  
1305      klo=klo+1  
1306      end do  
1307  
1308      khi=klo+1  
1309
```

/Users/lplopa/Compare/camb_des/cmbmain.f90, Top line: 1284

```
1284      ThisCT%q%dpoints(  
1285      deallocate(ThisCT  
1286      allocate(ThisCT%D  
1287      ThisCT%Delta_p_l_  
1288      end if  
1289  
1290      end if !ExactClosedSum  
1291  
1292  
1293      end subroutine setkValues  
1294  
1295      subroutine InterpolateSou  
1296      implicit none  
1297      integer i,khi,klo, step  
1298      real(dl) xf,b0,ho,a0,ho2o  
1299      type(IntegrationVars) IV  
1300  
1301  
1302      !      finding position of  
1303  
1304      !Can't use the following  
1305      !      klo = min(Evo  
1306      !This is a bit inefficien  
1307      klo=1  
1308      do while ((IV%q > Evolve_  
1309      klo=klo+1  
1310      end do  
1311  
1312      khi=klo+1  
1313
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1310

```
1310
1311      ho=Evolve_q%points(khi)-E
1312      a0=(Evolve_q%points(khi)-
1313      b0=(IV%q-Evolve_q%points(
1314      ho2o6 = ho**2/6
1315      a03=(a0**3-a0)
1316      b03=(b0**3-b0)
1317      IV%SourceSteps = 0
1318
1319      !      Interpolating the s
1320      !      wavelength.
1321      step=2
1322      do i=2, TimeSteps%npoints
1323          xf=IV%q*(CP%tau0-Time
1324          if (CP%WantTensors) t
1325              if (IV%q*TimeStep
1326                  step=i
1327                  IV%Source_q(i
1328                  b0*Src(khi,1:
1329                  b03*ddSrc(khi
1330              else
1331                  IV%Source_q(i
1332              end if
1333          end if
1334          if (CP%WantVectors) t
1335              if (IV%q*TimeStep
1336                  step=i
1337                  IV%Source_q(i
1338                  b0*Src(khi,1:
1339                  b03*ddSrc(khi
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1314

```
1314
1315      ho=Evolve_q%points(khi)-E
1316      a0=(Evolve_q%points(khi)-
1317      b0=(IV%q-Evolve_q%points(
1318      ho2o6 = ho**2/6
1319      a03=(a0**3-a0)
1320      b03=(b0**3-b0)
1321      IV%SourceSteps = 0
1322
1323      !      Interpolating the s
1324      !      wavelength.
1325      step=2
1326      do i=2, TimeSteps%npoints
1327          xf=IV%q*(CP%tau0-Time
1328          if (CP%WantTensors) t
1329              if (IV%q*TimeStep
1330                  step=i
1331                  IV%Source_q(i
1332                  b0*Src(kh
1333                  b03*ddSrc
1334              else
1335                  IV%Source_q(i
1336              end if
1337          end if
1338          if (CP%WantVectors) t
1339              if (IV%q*TimeStep
1340                  step=i
1341                  IV%Source_q(i
1342                  b0*Src(kh
1343                  b03*ddSrc
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1340

```
1340      else
1341          IV%Source_q(i
1342      end if
1343  end if
1344
1345      if (CP%WantScalars) t
1346          if ((DebugEvoluti
1347              .and. xf > 1.e-8_
1348              step=i
1349              IV%Source_q(i
1350              b0*Src(khi,1:
1351              +b03*ddSrc(kh
1352      else
1353          IV%Source_q(i
1354      end if
1355  end if
1356 end do
1357 IV%SourceSteps = step
1358
1359
1360 if (.not.CP%flat) then
1361     do i=1, SourceNum
1362         call spline(TimesS
1363             spl_large,spl_lar
1364     end do
1365 end if
1366
1367 IV%SourceSteps = IV%Sourc
1368 !This is a fix for a comp
1369
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1344

```
1344      else
1345          IV%Source_q(i
1346      end if
1347  end if
1348
1349      if (CP%WantScalars) t
1350          if ((DebugEvoluti
1351              .and. xf > 1.
1352              step=i
1353              IV%Source_q(i
1354              b0*Src(kh
1355              +b03*ddSr
1356      else
1357          IV%Source_q(i
1358      end if
1359  end if
1360 end do
1361 IV%SourceSteps = step
1362
1363
1364 if (.not.CP%flat) then
1365     do i=1, SourceNum
1366         call spline(TimesS
1367             spl_large,spl
1368     end do
1369 end if
1370
1371 IV%SourceSteps = IV%Sourc
1372 !This is a fix for a comp
1373
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1370

```
1370      end subroutine
1371
1372
1373      subroutine IntegrationVar
1374      type(IntegrationVars), in
1375
1376      IV%Source_q(1,1:SourceNum
1377      IV%Source_q(TimeSteps%np
1378      IV%Source_q(TimeSteps%np
1379
1380      end  subroutine Integrati
1381
1382
1383      subroutine DoSourceIntegr
1384      integer j,ll,llmax
1385      real(dl) nu
1386      type(IntegrationVars) IV
1387
1388      nu=IV%q*CP%r
1389
1390      if (CP%closed) then
1391          if (nu<20 .or. CP%tau
1392              llmax=nint(nu)-1
1393          else
1394              llmax=nint(nu*rof
1395              llmax=min(llmax,n
1396          end if
1397      else
1398          llmax=nint(nu*CP%chi0
1399          if (llmax<15) then
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1374

```
1374      end subroutine
1375
1376
1377      subroutine IntegrationVar
1378      type(IntegrationVars), in
1379
1380      IV%Source_q(1,1:SourceNum
1381      IV%Source_q(TimeSteps%np
1382      IV%Source_q(TimeSteps%np
1383
1384      end  subroutine Integrati
1385
1386
1387      subroutine DoSourceIntegr
1388      integer j,ll,llmax
1389      real(dl) nu
1390      type(IntegrationVars) IV
1391
1392      nu=IV%q*CP%r
1393
1394      if (CP%closed) then
1395          if (nu<20 .or. CP%tau
1396              llmax=nint(nu)-1
1397          else
1398              llmax=nint(nu*rof
1399              llmax=min(llmax,n
1400          end if
1401      else
1402          llmax=nint(nu*CP%chi0
1403          if (llmax<15) then
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1400

```
1400          llmax=17 !AL Sept
1401          else
1402              llmax=nint(nu*rof
1403          end if
1404      end if
1405
1406      if (CP%flat) then
1407          call DoFlatIntegratio
1408      else
1409          do j=1,lSamp%10
1410              ll=lSamp%1(j)
1411              if (ll>llmax) exi
1412              call IntegrateSou
1413          end do !j loop
1414      end if
1415
1416      end subroutine DoSourceIn
1417
1418      function UseLimber(l,k)
1419      !Calculate lensing potent
1420      !even when sources calcul
1421      !(Limber better on small
1422      !This affects speed, esp.
1423      logical :: UseLimber
1424      integer l
1425      real(dl) :: k
1426
1427      !note increasing non-limb
1428      !use **0.5 to at least gi
1429      !Could be lower but care
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1404

```
1404          llmax=17 !AL Sept
1405          else
1406              llmax=nint(nu*rof
1407          end if
1408      end if
1409
1410      if (CP%flat) then
1411          call DoFlatIntegratio
1412      else
1413          do j=1,lSamp%10
1414              ll=lSamp%1(j)
1415              if (ll>llmax) exi
1416              call IntegrateSou
1417          end do !j loop
1418      end if
1419
1420      end subroutine DoSourceIn
1421
1422      function UseLimber(l,k)
1423      !Calculate lensing potent
1424      !even when sources calcul
1425      !(Limber better on small
1426      !This affects speed, esp.
1427      logical :: UseLimber
1428      integer l
1429      real(dl) :: k
1430
1431      !note increasing non-limb
1432      !use **0.5 to at least gi
1433      !Could be lower but care
```


/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1430

```
1430      UseLimber = 1 > 400*Accur
1431
1432      end function UseLimber
1433
1434      !cccccccccccccccccccccccccccccccccc
1435      !flat source integration
1436      subroutine DoFlatIntegrat
1437      implicit none
1438      type(IntegrationVars) IV
1439      integer llmax
1440      integer j
1441      logical DoInt
1442      real(dl) xlim,xlmax1
1443      real(dl) tmin, tmax
1444      real(dl) a2, J_1, aa(IV%S
1445      real(dl) xf, sums(SourceN
1446      real(dl) qmax_int
1447      integer bes_ix,n, bes_ind
1448
1449      !      Find the position i
1450      !      timestep
1451
1452      do j=1,IV%SourceSteps !Pr
1453          xf=abs(IV%q*(CP%tau0-
1454          bes_index(j)=Ranges_i
1455          !Precomputed values f
1456          bes_ix= bes_index(j)
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1434

```
1434      UseLimber = 1 > 400*Accur
1435
1436      end function UseLimber
1437
1438      !cccccccccccccccccccccccccccccccccc
1439      !flat source integration
1440      subroutine DoFlatIntegrat
1441      implicit none
1442      type(IntegrationVars) IV
1443      integer llmax
1444      integer j
1445      logical DoInt
1446      real(dl) xlim,xlmax1
1447      real(dl) tmin, tmax
1448      real(dl) a2, J_1, aa(IV%S
1449      real(dl) xf, sums(SourceN
1450      real(dl) qmax_int
1451      integer bes_ix,n, bes_ind
1452      integer custom_source_off
1453
1454      custom_source_off = num_r
1455
1456      !      Find the position i
1457      !      timestep
1458
1459      do j=1,IV%SourceSteps !Pr
1460          xf=abs(IV%q*(CP%tau0-
1461          bes_index(j)=Ranges_i
1462          !Precomputed values f
1463          bes_ix= bes_index(j)
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1457

```
1457      fac(j)=BessRanges%poi
1458      aa(j)=(BessRanges%poi
1459      fac(j)=fac(j)**2*aa(j
1460      end do
1461
1462      do j=1,max_bessels_l_inde
1463          if (lSamp%l(j) > llma
1464              xlim=xlimfrac*lSamp%l
1465              xlim=max(xlim,xlimmin
1466              xlim=lSamp%l(j)-xlim
1467              if (full_bessel_integ
1468                  tmin = TimeSteps%
1469              else
1470                  xlmxl=80*lSamp%l
1471                  tmin=CP%tau0-xlma
1472                  tmin=max(TimeStep
1473              end if
1474              tmax=CP%tau0-xlim/IV%
1475              tmax=min(CP%tau0,tmax
1476              tmin=max(TimeSteps%po
1477
1478              if (tmax < TimeSteps%
1479              sums(1:SourceNum) = 0
1480
1481              !As long as we sample
1482              !interpolate the Bess
1483
1484              if (SourceNum==2) the
1485                  !This is the inne
1486                  do n= Ranges_Inde
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1464

```
1464      fac(j)=BessRanges%poi
1465      aa(j)=(BessRanges%poi
1466      fac(j)=fac(j)**2*aa(j
1467      end do
1468
1469      do j=1,max_bessels_l_inde
1470          if (lSamp%l(j) > llma
1471              xlim=xlimfrac*lSamp%l
1472              xlim=max(xlim,xlimmin
1473              xlim=lSamp%l(j)-xlim
1474              if (full_bessel_integ
1475                  tmin = TimeSteps%
1476              else
1477                  xlmxl=80*lSamp%l
1478                  tmin=CP%tau0-xlma
1479                  tmin=max(TimeStep
1480              end if
1481              tmax=CP%tau0-xlim/IV%
1482              tmax=min(CP%tau0,tmax
1483              tmin=max(TimeSteps%po
1484
1485              if (tmax < TimeSteps%
1486              sums(1:SourceNum) = 0
1487
1488              !As long as we sample
1489              !interpolate the Bess
1490
1491              if (SourceNum==2) the
1492                  !This is the inne
1493                  do n= Ranges_Inde
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1487

```
1487      a2=aa(n)
1488      bes_ix=bes_in
1489
1490      J_1=a2*ajl(be
1491      *ajlpr(bes_ix
1492
1493      J_1 = J_1*Tim
1494      sums(1) = sum
1495      sums(2) = sum
1496      end do
1497  else
1498      qmax_int= max(850
1499      if (HighAccuracyD
1500      DoInt = .not. CP%
1501      if (DoInt) then
1502
1503      do n= Ranges_
1504          !Full Bes
1505          a2=aa(n)
1506          bes_ix=be
1507
1508          J_1=a2*aj
1509          *ajlpr(be
1510          J_1 = J_1
1511
1512          !The unwr
1513          sums(1) =
1514          sums(2) =
1515          sums(3) =
1516      end do
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1494

```
1494      a2=aa(n)
1495      bes_ix=bes_in
1496
1497      J_1=a2*ajl(be
1498      *ajlpr(be
1499
1500      J_1 = J_1*Tim
1501      sums(1) = sum
1502      sums(2) = sum
1503      end do
1504  else
1505      qmax_int= max(850
1506      if (HighAccuracyD
1507      DoInt = .not. CP%
1508      if (DoInt) then
1509          if (num_custo
1510              do n= Ran
1511                  !Full
1512                  a2=aa
1513                  bes_i
1514
1515                  J_1=a
1516                  *
1517                  J_1 =
1518
1519                  !The
1520                  sums(
1521                  sums(
1522                  sums(
1523      end do
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1516

```
1516      end if
1517      if (.not. DoInt .
1518          !Limber appro
1519          xf = CP%tau0-
1520          if (xf < Time
1521              n=Ranges_
1522              xf= (xf-T
1523              sums(3) =
1524          else
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1524

```
1524      else
1525          do n= Ran
1526              !Full
1527              a2=aa
1528              bes_i
1529
1530              J_l=a
1531              *
1532              J_l =
1533
1534              !The
1535              sums (
1536              sums (
1537              sums (
1538              sums (
1539              do s_
1540                  s
1541              end d
1542          end do
1543
1544      end if
1545      end if
1546      if (.not. DoInt .
1547          !Limber appro
1548          xf = CP%tau0-
1549          if (xf < Time
1550              n=Ranges_
1551              xf= (xf-T
1552              sums(3) =
1553      else
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1525

```
1525             sums(3)=0
1526             end if
1527         end if
1528     end if
1529
1530     ThisCT%Delta_p_l_k(1:
1531 end do
1532
1533 end subroutine DoFlatInte
1534
1535
1536
1537 !non-flat source integrat
1538
1539 subroutine IntegrateSourc
1540 use SpherBessels
1541 type(IntegrationVars) IV
1542 logical DoInt
1543 integer l,j, nstart,nDiss
1544 real(dl) nu,ChiDissipativ
1545 real(dl) xf,x,chi, minyl
1546 real(dl) sums(SourceNum),
1547
1548 !Calculate chi where for
1549 x=sqrt(real(l*(l+1),dl))/
1550
1551 ChiDissipative=invsinfunc
1552
1553 ChiStart=ChiDissipative
1554 !Move down a bit to get s
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1554

```
1554             sums(3)=0
1555             end if
1556         end if
1557     end if
1558
1559     ThisCT%Delta_p_l_k(1:
1560 end do
1561
1562 end subroutine DoFlatInte
1563
1564
1565
1566 !non-flat source integrat
1567
1568 subroutine IntegrateSourc
1569 use SpherBessels
1570 type(IntegrationVars) IV
1571 logical DoInt
1572 integer l,j, nstart,nDiss
1573 real(dl) nu,ChiDissipativ
1574 real(dl) xf,x,chi, minyl
1575 real(dl) sums(SourceNum),
1576
1577 !Calculate chi where for
1578 x=sqrt(real(l*(l+1),dl))/
1579
1580 ChiDissipative=invsinfunc
1581
1582 ChiStart=ChiDissipative
1583 !Move down a bit to get s
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1555

```
1555      if (nu<300) ChiStart = ma
1556
1557      !Then get nearest source
1558      tDissipative=CP%tau0 - CP
1559      if (tDissipative<TimeStep
1560          nDissipative=2
1561      else
1562          nDissipative = Ranges
1563      endif
1564      nDissipative=min(nDissipa
1565
1566      tDissipative = TimeSteps%
1567
1568      ChiStart = max(1d-8,(CP%
1569
1570      !Get values at ChiStart
1571
1572      call USpherBesselWithDeri
1573
1574      nstart=nDissipative
1575      chi=ChiStart
1576
1577      if ((CP%WantScalars)) the
1578          !Integrate chi down i
1579          ! cuts off when ujl g
1580          minyl= 0.5d-4/l/Accur
1581          sums=0
1582          qmax_int= max(850,lSa
1583          if (HighAccuracyDefau
1584          DoInt = SourceNum/=3
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1584

```
1584      if (nu<300) ChiStart = ma
1585
1586      !Then get nearest source
1587      tDissipative=CP%tau0 - CP
1588      if (tDissipative<TimeStep
1589          nDissipative=2
1590      else
1591          nDissipative = Ranges
1592      endif
1593      nDissipative=min(nDissipa
1594
1595      tDissipative = TimeSteps%
1596
1597      ChiStart = max(1d-8,(CP%
1598
1599      !Get values at ChiStart
1600
1601      call USpherBesselWithDeri
1602
1603      nstart=nDissipative
1604      chi=ChiStart
1605
1606      if ((CP%WantScalars)) the
1607          !Integrate chi down i
1608          ! cuts off when ujl g
1609          minyl= 0.5d-4/l/Accur
1610          sums=0
1611          qmax_int= max(850,lSa
1612          if (HighAccuracyDefau
1613          DoInt = SourceNum/=3
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1585

```
1585         if (DoInt) then
1586             if ((nstart < min
1587                 y1=y1dis
1588                 y2=y2dis
1589                 nnow=nstart
1590                 do nrange = 1
1591                     if (nrang
1592                         ntop
1593                     else
1594                         ntop
1595                     end if
1596                     if (nnow
1597                         call
1598                         nu,l,
1599                         sums
1600                         nnow
1601                         if (c
1602                     end if
1603                 end do
1604             end if !integrate
1605
1606             !Integrate chi up
1607             if (nstart > 2) t
1608                 y1=y1dis
1609                 y2=y2dis
1610                 chi=ChiStart
1611                 nnow=nstart
1612                 do nrange = T
1613                     nbot = Ti
1614                     if (nnow
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1614

```
1614         if (DoInt) then
1615             if ((nstart < min
1616                 y1=y1dis
1617                 y2=y2dis
1618                 nnow=nstart
1619                 do nrange = 1
1620                     if (nrang
1621                         ntop
1622                     else
1623                         ntop
1624                     end if
1625                     if (nnow
1626                         call
1627                         n
1628                         sums
1629                         nnow
1630                     if (c
1631                     end if
1632                 end do
1633             end if !integrate
1634
1635             !Integrate chi up
1636             if (nstart > 2) t
1637                 y1=y1dis
1638                 y2=y2dis
1639                 chi=ChiStart
1640                 nnow=nstart
1641                 do nrange = T
1642                     nbot = Ti
1643                     if (nnow
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1615

```
1615      call
1616      nu,l,
1617      sums=
1618      if (c
1619      nnow
1620      end if
1621      end do
1622      end if
1623      end if !DoInt
1624      if (SourceNum==3 .and
1625      !Limber approxima
1626      xf = CP%tau0-invs
1627      if (xf < TimeStep
1628      nbot=Ranges_I
1629      xf= (xf-Times
1630      sums(3) = (IV
1631      sqrt(pi/2/(1+
1632      else
1633      sums(3) = 0
1634      end if
1635      end if
1636
1637      ThisCT%Delta_p_l_k(1:
1638
1639      end if !Do Scalars
1640
1641      if ((CP%WantTensors)) the
1642      chi=ChiStart
1643
1644      !Integrate chi down i
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1644

```
1644      call
1645      n
1646      sums=
1647      if (c
1648      nnow
1649      end if
1650      end do
1651      end if
1652      end if !DoInt
1653      if (SourceNum==3 .and
1654      !Limber approxima
1655      xf = CP%tau0-invs
1656      if (xf < TimeStep
1657      nbot=Ranges_I
1658      xf= (xf-Times
1659      sums(3) = (IV
1660      sqrt(pi/2
1661      else
1662      sums(3) = 0
1663      end if
1664      end if
1665
1666      ThisCT%Delta_p_l_k(1:
1667
1668      end if !Do Scalars
1669
1670      if ((CP%WantTensors)) the
1671      chi=ChiStart
1672
1673      !Integrate chi down i
```


/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1645

```
1645      !DoRangeInt cuts off
1646      miny1= 1.d-6/l/Accura
1647      if ((nstart < TimeSte
1648          y1=y1dis
1649          y2=y2dis
1650          nnow=nstart
1651          do nrange = 1, Tim
1652              if (nrange ==
1653                  ntop = Ti
1654              else
1655                  ntop = Ti
1656              end if
1657              if (nnow < nt
1658                  call DoRa
1659                      nu,l,y1,y
1660
1661                  ThisCT%De
1662
1663                  nnow = nt
1664                  if (chi==
1665                      end if
1666              end do
1667          end if
1668
1669
1670      !Integrate chi up in
1671      if (nstart > 2) then
1672          y1=y1dis
1673          y2=y2dis
1674          chi=ChiStart
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1674

```
1674      !DoRangeInt cuts off
1675      miny1= 1.d-6/l/Accura
1676      if ((nstart < TimeSte
1677          y1=y1dis
1678          y2=y2dis
1679          nnow=nstart
1680          do nrange = 1, Tim
1681              if (nrange ==
1682                  ntop = Ti
1683              else
1684                  ntop = Ti
1685              end if
1686              if (nnow < nt
1687                  call DoRa
1688                      nu,l,
1689
1690                  ThisCT%De
1691
1692                  nnow = nt
1693                  if (chi==
1694                      end if
1695              end do
1696          end if
1697
1698
1699      !Integrate chi up in
1700      if (nstart > 2) then
1701          y1=y1dis
1702          y2=y2dis
1703          chi=ChiStart
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1675

```
1675
1676      nnow=nstart
1677      do nrange = Times
1678          nbot = TimeSt
1679          if (nnow > n
1680              call DoRa
1681              nu,l,y1,y
1682              ThisCT%De
1683
1684              nnow = nb
1685              if (chi==
1686                  end if
1687          end do
1688      end if
1689
1690  end if !Do Tensors
1691
1692  end subroutine Integrates
1693
1694
1695
1696  subroutine DoRangeInt(IV,
1697      !Non-flat version
1698
1699      !returns chi at end of in
1700      ! This subroutine integra
1701      ! It calculates ujl by in
1702      ! differential equation f
1703      ! dtau is the spacing of
1704
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1704

```
1704
1705      nnow=nstart
1706      do nrange = Times
1707          nbot = TimeSt
1708          if (nnow > n
1709              call DoRa
1710              nu,l,
1711              ThisCT%De
1712
1713              nnow = nb
1714              if (chi==
1715                  end if
1716          end do
1717      end if
1718
1719  end if !Do Tensors
1720
1721  end subroutine Integrates
1722
1723
1724
1725  subroutine DoRangeInt(IV,
1726      !Non-flat version
1727
1728      !returns chi at end of in
1729      ! This subroutine integra
1730      ! It calculates ujl by in
1731      ! differential equation f
1732      ! dtau is the spacing of
1733
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1705

```
1705      use precision
1706      use ModelParams
1707      type(IntegrationVars) IV
1708      integer l,nIntSteps,nstar
1709      real(dl) nu,dtau,num1,num
1710      real(dl) a,b,tmpa,tmpb,hh
1711      real(dl) nu2,chi,chiDisp,
1712
1713      real(dl) tmp,dtau2o6,y1,y
1714      real(dl) dchimax,dchisour
1715      real(dl), parameter:: MIN
1716      logical Interpolate
1717      real(dl) scale1
1718      real(dl) IntAccuracyBoost
1719      real(dl) sources(SourceNu
1720
1721      IntAccuracyBoost=Accuracy
1722
1723      ! atau0 is the array with
1724      if (nend==nstart) then
1725          out = 0
1726          return
1727      end if
1728
1729      dchisource=dtau/CP%r
1730
1731      num1=1._dl/nu
1732
1733      scale1=1/scale
1734      if (scale1>=2400) then
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1734

```
1734      use precision
1735      use ModelParams
1736      type(IntegrationVars) IV
1737      integer l,nIntSteps,nstar
1738      real(dl) nu,dtau,num1,num
1739      real(dl) a,b,tmpa,tmpb,hh
1740      real(dl) nu2,chi,chiDisp,
1741
1742      real(dl) tmp,dtau2o6,y1,y
1743      real(dl) dchimax,dchisour
1744      real(dl), parameter:: MIN
1745      logical Interpolate
1746      real(dl) scale1
1747      real(dl) IntAccuracyBoost
1748      real(dl) sources(SourceNu
1749
1750      IntAccuracyBoost=Accuracy
1751
1752      ! atau0 is the array with
1753      if (nend==nstart) then
1754          out = 0
1755          return
1756      end if
1757
1758      dchisource=dtau/CP%r
1759
1760      num1=1._dl/nu
1761
1762      scale1=1/scale
1763      if (scale1>=2400) then
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1735

```
1735      num2=num1*2.5
1736      else if (scale1< 50) then
1737          num2=num1*0.8_dl
1738      else
1739          num2=num1*1.5_dl
1740      end if
1741      !Dec 2003, since decrease
1742      if (dtau==dtaurec_q) then
1743          num2=num2/4
1744      end if
1745
1746      if (HighAccuracyDefault .
1747      IntAccuracyBoost=IntAccur
1748
1749      if (num2*IntAccuracyBoost
1750      .or. (nstart>IV%SourceSte
1751          out = 0
1752          y1=0._dl !So we know
1753          y2=0._dl
1754          chi=(CP%tau0-TimeStep
1755          return
1756      end if
1757
1758      Startn=nstart
1759      if (nstart>IV%SourceSteps
1760          chi=(CP%tau0-TimeStep
1761          Startn=IV%SourceSteps
1762          call USpherBesselWith
1763      else if ((y2==0._dl).and.
1764          call USpherBesselWith
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1764

```
1764      num2=num1*2.5
1765      else if (scale1< 50) then
1766          num2=num1*0.8_dl
1767      else
1768          num2=num1*1.5_dl
1769      end if
1770      !Dec 2003, since decrease
1771      if (dtau==dtaurec_q) then
1772          num2=num2/4
1773      end if
1774
1775      if (HighAccuracyDefault .
1776      IntAccuracyBoost=IntA
1777
1778      if (num2*IntAccuracyBoost
1779      .or. (nstart>IV%Sourc
1780          out = 0
1781          y1=0._dl !So we know
1782          y2=0._dl
1783          chi=(CP%tau0-TimeStep
1784          return
1785      end if
1786
1787      Startn=nstart
1788      if (nstart>IV%SourceSteps
1789          chi=(CP%tau0-TimeStep
1790          Startn=IV%SourceSteps
1791          call USpherBesselWith
1792      else if ((y2==0._dl).and.
1793          call USpherBesselWith
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1765

```
1765      end if
1766
1767      if (CP%closed) then
1768          !Need to cut off when
1769          chiDispTop = pi - chi
1770      else
1771          chiDispTop = 1d20
1772      end if
1773
1774      minuj1=MINUJ11/1/IntAccur
1775      isgn=sign(1,Startn-nend)!
1776      !higher n, later time, sm
1777
1778      sgn= isgn
1779
1780      nlowest=min(Startn,nend)
1781      aux1=1._dl*CP%r/dtau !us
1782      aux2=(CP%tau0-TimeSteps%p
1783
1784      nu2=nu*nu
1785      ap1=1*(1+1)
1786      sh=rofChi(chi)
1787
1788      if (scale1 < 1100) then
1789          dchimax= 0.3*num1
1790      else if (scale1 < 1400) t
1791          dchimax=0.25_dl*num1
1792      else
1793          dchimax=0.35_dl*num1
1794      end if
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1794

```
1794      end if
1795
1796      if (CP%closed) then
1797          !Need to cut off when
1798          chiDispTop = pi - chi
1799      else
1800          chiDispTop = 1d20
1801      end if
1802
1803      minuj1=MINUJ11/1/IntAccur
1804      isgn=sign(1,Startn-nend)!
1805      !higher n, later time, sm
1806
1807      sgn= isgn
1808
1809      nlowest=min(Startn,nend)
1810      aux1=1._dl*CP%r/dtau !us
1811      aux2=(CP%tau0-TimeSteps%p
1812
1813      nu2=nu*nu
1814      ap1=1*(1+1)
1815      sh=rofChi(chi)
1816
1817      if (scale1 < 1100) then
1818          dchimax= 0.3*num1
1819      else if (scale1 < 1400) t
1820          dchimax=0.25_dl*num1
1821      else
1822          dchimax=0.35_dl*num1
1823      end if
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1795

```
1795
1796      dchimax=dchimax/IntAccura
1797
1798      ujl=y1/sh
1799      sources = IV%Source_q(Sta
1800
1801      out = 0.5_d1*ujl*sources
1802
1803      Interpolate = dchisource
1804      if (Interpolate) then !sp
1805          delchi=dchimax
1806          Deltachi=sgn*(TimeSte
1807          nIntSteps=int(Deltach
1808          delchi=Deltachi/nIntS
1809          dtau2o6=(CP%r*delchi)
1810      else !step size is that o
1811          delchi=dchisource
1812          nIntSteps=isgn*(Start
1813      end if
1814
1815      sgndelchi=delchi*sgn
1816      tmp=(ap1/sh**2 - nu2)
1817      hh=0.5_d1*sgndelchi
1818      h6=sgndelchi/6._d1
1819
1820
1821      do i=1,nIntSteps
1822          ! One step in the ujl
1823          ! fourth-order Runge-
1824
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1824

```
1824
1825      dchimax=dchimax/IntAccura
1826
1827      ujl=y1/sh
1828      sources = IV%Source_q(Sta
1829
1830      out = 0.5_d1*ujl*sources
1831
1832      Interpolate = dchisource
1833      if (Interpolate) then !sp
1834          delchi=dchimax
1835          Deltachi=sgn*(TimeSte
1836          nIntSteps=int(Deltach
1837          delchi=Deltachi/nIntS
1838          dtau2o6=(CP%r*delchi)
1839      else !step size is that o
1840          delchi=dchisource
1841          nIntSteps=isgn*(Start
1842      end if
1843
1844      sgndelchi=delchi*sgn
1845      tmp=(ap1/sh**2 - nu2)
1846      hh=0.5_d1*sgndelchi
1847      h6=sgndelchi/6._d1
1848
1849
1850      do i=1,nIntSteps
1851          ! One step in the ujl
1852          ! fourth-order Runge-
1853
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1825

1825	dydchi1=y2	!d
1826	dydchi2=tmp*y1	!d
1827	xh=chi+hh	!m
1828	yt1=y1+hh*dydchi1	!y
1829	yt2=y2+hh*dydchi2	!y
1830	dzt1=yt2	!d
1831	tmp=(ap1/rofChi(xh)**	
1832		
1833		
1834	dzt2=tmp*yt1	!d
1835		
1836	yt1=y1+hh*dzt1	!y
1837	yt2=y2+hh*dzt2	!y
1838		
1839	dym1=yt2	!d
1840	dym2=tmp*yt1	!d
1841	yt1=y1+sgndelchi*dym1	
1842	dym1=dzt1+dym1	
1843	yt2=y2+sgndelchi*dym2	
1844	dym2=dzt2+dym2	
1845		
1846	chi=chi+sgndelchi	
1847	sh=rofChi(chi)	
1848	dzt1=yt2	!d
1849	tmp=(ap1/sh**2 - nu2)	
1850	dzt2=tmp*yt1	!d
1851	y1=y1+h6*(dydchi1+dzt	
1852	y2=y2+h6*(dydchi2+dzt	
1853		
1854	uj1=y1/sh	

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1854

1854	dydchi1=y2	!d
1855	dydchi2=tmp*y1	!d
1856	xh=chi+hh	!m
1857	yt1=y1+hh*dydchi1	!y
1858	yt2=y2+hh*dydchi2	!y
1859	dzt1=yt2	!d
1860	tmp=(ap1/rofChi(xh)**	
1861		
1862		
1863	dzt2=tmp*yt1	!d
1864		
1865	yt1=y1+hh*dzt1	!y
1866	yt2=y2+hh*dzt2	!y
1867		
1868	dym1=yt2	!d
1869	dym2=tmp*yt1	!d
1870	yt1=y1+sgndelchi*dym1	
1871	dym1=dzt1+dym1	
1872	yt2=y2+sgndelchi*dym2	
1873	dym2=dzt2+dym2	
1874		
1875	chi=chi+sgndelchi	
1876	sh=rofChi(chi)	
1877	dzt1=yt2	!d
1878	tmp=(ap1/sh**2 - nu2)	
1879	dzt2=tmp*yt1	!d
1880	y1=y1+h6*(dydchi1+dzt	
1881	y2=y2+h6*(dydchi2+dzt	
1882		
1883	uj1=y1/sh	

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1855

```
1855      if ((isgn<0).and.(y1*  
1856          chi=0._dl  
1857          exit    !If this h  
1858      end if  
  
1859  
1860  
1861      if (Interpolate) then  
1862          ! Interpolate the  
1863          tau1=aux2-aux1*ch  
1864          is=int(tau1)  
1865          b=tau1-is  
1866  
1867          if (b > 0.998) th  
1868              !may save tim  
1869              sources = IV%  
1870          else  
1871              a=1._dl-b  
1872              tmpa=(a**3-a)  
1873              tmpb=(b**3-b)  
1874              sources=a*IV%  
1875              (tmpa*IV%ddSo  
1876              tmpb*IV%ddSou  
1877          end if  
1878      else  
1879          sources = IV%Sour  
1880      end if  
1881  
1882      out = out + ujl*sourc  
1883  
1884      if (((isgn<0).or.(chi
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1884

```
1884      if ((isgn<0).and.(y1*  
1885          chi=0._dl  
1886          exit    !If this h  
1887      end if  
  
1888  
1889  
1890      if (Interpolate) then  
1891          ! Interpolate the  
1892          tau1=aux2-aux1*ch  
1893          is=int(tau1)  
1894          b=tau1-is  
1895  
1896          if (b > 0.998) th  
1897              !may save tim  
1898              sources = IV%  
1899          else  
1900              a=1._dl-b  
1901              tmpa=(a**3-a)  
1902              tmpb=(b**3-b)  
1903              sources=a*IV%  
1904              (tmpa*IV%  
1905              tmpb*IV%d  
1906          end if  
1907      else  
1908          sources = IV%Sour  
1909      end if  
1910  
1911      out = out + ujl*sourc  
1912  
1913      if (((isgn<0).or.(chi
```


/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1885

```
1885             chi=0
1886             exit !break when
1887         end if
1888     end do
1889
1890     out = (out - sources*ujl/
1891
1892 end subroutine DoRangeInt
1893
1894 subroutine DoRangeIntTens
1895 ! It calculates ujl by in
1896 ! differential equation f
1897 ! nstart and nend are the
1898 ! integration.
1899 ! dtau is the spacing of
1900
1901 use precision
1902 use ModelParams
1903 type(IntegrationVars), ta
1904 integer l,nIntSteps,nstar
1905 real(dl) nu,dtau,num1,num
1906 real(dl) a,b,tmpa,tmpb,hh
1907 real(dl) nu2,chi,chiDisp,
1908 real(dl) dydchi1,dydchi2,
1909
1910 real(dl) tmp,dtau2o6,y1,y
1911 real(dl) dchimax,dchisour
1912 real(dl), parameter:: MIN
1913 logical Interpolate
1914 real(dl) out(SourceNum),
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1914

```
1914             chi=0
1915             exit !break when
1916         end if
1917     end do
1918
1919     out = (out - sources*ujl/
1920
1921 end subroutine DoRangeInt
1922
1923 subroutine DoRangeIntTens
1924 ! It calculates ujl by in
1925 ! differential equation f
1926 ! nstart and nend are the
1927 ! integration.
1928 ! dtau is the spacing of
1929
1930 use precision
1931 use ModelParams
1932 type(IntegrationVars), ta
1933 integer l,nIntSteps,nstar
1934 real(dl) nu,dtau,num1,num
1935 real(dl) a,b,tmpa,tmpb,hh
1936 real(dl) nu2,chi,chiDisp,
1937 real(dl) dydchi1,dydchi2,
1938
1939 real(dl) tmp,dtau2o6,y1,y
1940 real(dl) dchimax,dchisour
1941 real(dl), parameter:: MIN
1942 logical Interpolate
1943 real(dl) out(SourceNum),
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1915

```
1915      real(dl), dimension(:, :),
1916
1917      sourcep => IV%Source_q(:,
1918      ddsourcep => IV%ddSource_
1919
1920
1921      if (nend==nstart) then
1922          out=0
1923          return
1924      end if
1925      minujl=MINUJL1*AccuracyBo
1926      isgn=sign(1,nstart-nend)!
1927      !higher n, later time, sm
1928
1929      if (CP%closed) then
1930          !Need to cut off when
1931          chiDispTop = pi - chi
1932      else
1933          chiDispTop = 1d20
1934      end if
1935
1936      num1=1._dl/nu
1937      dchisource=dtau/CP%r
1938
1939      scale1=1/scale
1940      if (scale1>=2000) then
1941          num2=num1*4
1942      else if (scale1>=1000) th
1943          num2=num1*2.5
1944      else if (scale1< 75) then
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1944

```
1944      real(dl), dimension(:, :),
1945
1946      sourcep => IV%Source_q(:,
1947      ddsourcep => IV%ddSource_
1948
1949
1950      if (nend==nstart) then
1951          out=0
1952          return
1953      end if
1954      minujl=MINUJL1*AccuracyBo
1955      isgn=sign(1,nstart-nend)!
1956      !higher n, later time, sm
1957
1958      if (CP%closed) then
1959          !Need to cut off when
1960          chiDispTop = pi - chi
1961      else
1962          chiDispTop = 1d20
1963      end if
1964
1965      num1=1._dl/nu
1966      dchisource=dtau/CP%r
1967
1968      scale1=1/scale
1969      if (scale1>=2000) then
1970          num2=num1*4
1971      else if (scale1>=1000) th
1972          num2=num1*2.5
1973      else if (scale1< 75) then
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1945

```
1945      num2=num1*0.1_dl
1946      else if (scale1<180) then
1947          num2=num1*0.3_dl
1948      else if (scale1 < 600) th
1949          num2=num1*0.8_dl
1950      else
1951          num2=num1
1952      end if
1953
1954      if ((isgn==1).and.(num2*A
1955          out = 0
1956          y1=0._dl !!So we know
1957          y2=0._dl
1958          chi=(CP%tau0-TimeStep
1959          return
1960      end if
1961      if ((y2==0._dl).and.(y1==
1962
1963      sgn=isgn
1964
1965      nlowest=min(nstart,nend)
1966      aux1=1._dl*CP%r/dtau !us
1967      aux2=(CP%tau0-TimeSteps%p
1968
1969
1970      nu2=nu*nu
1971      ap1=1*(1+1)
1972
1973      sh=rofChi(chi)
1974
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 1974

```
1974      num2=num1*0.1_dl
1975      else if (scale1<180) then
1976          num2=num1*0.3_dl
1977      else if (scale1 < 600) th
1978          num2=num1*0.8_dl
1979      else
1980          num2=num1
1981      end if
1982
1983      if ((isgn==1).and.(num2*A
1984          out = 0
1985          y1=0._dl !!So we know
1986          y2=0._dl
1987          chi=(CP%tau0-TimeStep
1988          return
1989      end if
1990      if ((y2==0._dl).and.(y1==
1991
1992      sgn=isgn
1993
1994      nlowest=min(nstart,nend)
1995      aux1=1._dl*CP%r/dtau !us
1996      aux2=(CP%tau0-TimeSteps%p
1997
1998
1999      nu2=nu*nu
2000      ap1=1*(1+1)
2001
2002      sh=rofChi(chi)
2003
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 1975

```
1975      if (scale1 < 120) then
1976          dchimax=0.6_dl*num1
1977      else if (scale1 < 1400) t
1978          dchimax=0.25_dl*num1
1979      else
1980          dchimax=0.35_dl*num1
1981      end if
1982
1983      dchimax=dchimax/AccuracyB
1984
1985      ujl=y1/sh
1986      out = ujl * sourcep(nstar
1987
1988      Interpolate = dchisource
1989      if (Interpolate) then !sp
1990          delchi=dchimax
1991          Deltachi=sgn*(TimeSte
1992          nIntSteps=int(Deltach
1993          delchi=Deltachi/nIntS
1994          dtau2o6=(CP%r*delchi)
1995      else !step size is that o
1996          delchi=dchisource
1997          nIntSteps=isgn*(nstar
1998      end if
1999
2000
2001      sgndelchi=delchi*sgn
2002      tmp=(ap1/sh**2 - nu2)
2003      hh=0.5_dl*sgndelchi
2004      h6=sgndelchi/6._dl
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 2004

```
2004      if (scale1 < 120) then
2005          dchimax=0.6_dl*num1
2006      else if (scale1 < 1400) t
2007          dchimax=0.25_dl*num1
2008      else
2009          dchimax=0.35_dl*num1
2010      end if
2011
2012      dchimax=dchimax/AccuracyB
2013
2014      ujl=y1/sh
2015      out = ujl * sourcep(nstar
2016
2017      Interpolate = dchisource
2018      if (Interpolate) then !sp
2019          delchi=dchimax
2020          Deltachi=sgn*(TimeSte
2021          nIntSteps=int(Deltach
2022          delchi=Deltachi/nIntS
2023          dtau2o6=(CP%r*delchi)
2024      else !step size is that o
2025          delchi=dchisource
2026          nIntSteps=isgn*(nstar
2027      end if
2028
2029
2030      sgndelchi=delchi*sgn
2031      tmp=(ap1/sh**2 - nu2)
2032      hh=0.5_dl*sgndelchi
2033      h6=sgndelchi/6._dl
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 2005

```
2005
2006
2007      do i=1,nIntSteps
2008          ! One step in the ujl
2009          ! fourth-order Runge-
2010
2011          dydchi1=y2          !d
2012          dydchi2=tmp*y1      !d
2013          xh=chi+hh          !m
2014          yt1=y1+hh*dydchi1   !y
2015          yt2=y2+hh*dydchi2   !y
2016          dyt1=yt2           !d
2017          tmp=(ap1/rofChi(xh)**
2018
2019
2020          dyt2=tmp*yt1        !d
2021          yt1=y1+hh*dyt1      !y
2022          yt2=y2+hh*dyt2      !y
2023
2024          dym1=yt2            !d
2025          dym2=tmp*yt1        !d
2026          yt1=y1+sgndelchi*dym1
2027          dym1=dyt1+dym1
2028          yt2=y2+sgndelchi*dym2
2029          dym2=dyt2+dym2
2030
2031          chi=chi+sgndelchi
2032          sh=rofChi(chi)
2033          dyt1=yt2            !d
2034          tmp=(ap1/sh**2 - nu2)
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 2034

```
2034
2035
2036      do i=1,nIntSteps
2037          ! One step in the ujl
2038          ! fourth-order Runge-
2039
2040          dydchi1=y2          !d
2041          dydchi2=tmp*y1      !d
2042          xh=chi+hh          !m
2043          yt1=y1+hh*dydchi1   !y
2044          yt2=y2+hh*dydchi2   !y
2045          dyt1=yt2           !d
2046          tmp=(ap1/rofChi(xh)**
2047
2048
2049          dyt2=tmp*yt1        !d
2050          yt1=y1+hh*dyt1      !y
2051          yt2=y2+hh*dyt2      !y
2052
2053          dym1=yt2            !d
2054          dym2=tmp*yt1        !d
2055          yt1=y1+sgndelchi*dym1
2056          dym1=dyt1+dym1
2057          yt2=y2+sgndelchi*dym2
2058          dym2=dyt2+dym2
2059
2060          chi=chi+sgndelchi
2061          sh=rofChi(chi)
2062          dyt1=yt2            !d
2063          tmp=(ap1/sh**2 - nu2)
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 2035

2035	dyt2=tmp*yt1	!d	2035
2036	y1=y1+h6*(dydchi1+dyt		2036
2037	y2=y2+h6*(dydchi2+dyt		2037
2038			2038
2039	uj1=y1/sh		2039
2040	if ((isgn<0).and.(y1*		2040
2041	chi=0._dl		2041
2042	exit !exit beca		2042
2043	end if		2043
2044			2044
2045	if (Interpolate) then		2045
2046	! Interpolate the		2046
2047	taui=aux2-aux1*ch		2047
2048	is=int(taui)		2048
2049	b=taui-is		2049
2050	if (b > 0.995) th		2050
2051	!may save tim		2051
2052	is=is+1		2052
2053	source = sour		2053
2054	else		2054
2055	a=1._dl-b		2055
2056	tmpa=(a**3-a)		2056
2057	tmpb=(b**3-b)		2057
2058	source = a*so		2058
2059	(tmpa*ddsourc		2059
2060	end if		2060
2061	else		2061
2062	source = sourcep(2062
2063	end if		2063
2064	out = out + source *		2064

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 2064

2064	dyt2=tmp*yt1	!d	2064
2065	y1=y1+h6*(dydchi1+dyt		2065
2066	y2=y2+h6*(dydchi2+dyt		2066
2067			2067
2068	uj1=y1/sh		2068
2069	if ((isgn<0).and.(y1*		2069
2070	chi=0._dl		2070
2071	exit !exit beca		2071
2072	end if		2072
2073			2073
2074	if (Interpolate) then		2074
2075	! Interpolate the		2075
2076	taui=aux2-aux1*ch		2076
2077	is=int(taui)		2077
2078	b=taui-is		2078
2079	if (b > 0.995) th		2079
2080	!may save tim		2080
2081	is=is+1		2081
2082	source = sour		2082
2083	else		2083
2084	a=1._dl-b		2084
2085	tmpa=(a**3-a)		2085
2086	tmpb=(b**3-b)		2086
2087	source = a*so		2087
2088	(tmpa*dds		2088
2089	end if		2089
2090	else		2090
2091	source = sourcep(2091
2092	end if		2092
2093	out = out + source *		2093

/Users/lplop/Compare/camb_simdata/cmbmain.f90, Top line: 2065

```
2065
2066         if (((isgn<0).or.(chi
2067             chi=0
2068             exit !break when
2069         end if
2070     end do
2071
2072     out = (out - source * ujl
2073
2074     end subroutine DoRangeInt
2075
2076     subroutine GetInitPowerAr
2077     integer, intent(in) :: nu
2078     real(dl) pows(numks), ks(
2079     integer i
2080
2081     do i = 1, numks
2082         !!change to vec...
2083         pows(i) = ScalarPowe
2084         if (global_error_flag
2085     end do
2086
2087     end subroutine GetInitPow
2088
2089
2090     subroutine GetInitPowerAr
2091     integer, intent(in) :: nu
2092     real(dl) pows(numks), ks(
2093     integer i
2094
```

/Users/lplop/Compare/camb_des/cmbmain.f90, Top line: 2094

```
2094
2095         if (((isgn<0).or.(chi
2096             chi=0
2097             exit !break when
2098         end if
2099     end do
2100
2101     out = (out - source * ujl
2102
2103     end subroutine DoRangeInt
2104
2105     subroutine GetInitPowerAr
2106     integer, intent(in) :: nu
2107     real(dl) pows(numks), ks(
2108     integer i
2109
2110     do i = 1, numks
2111         !!change to vec...
2112         pows(i) = ScalarPowe
2113         if (global_error_flag
2114     end do
2115
2116     end subroutine GetInitPow
2117
2118
2119     subroutine GetInitPowerAr
2120     integer, intent(in) :: nu
2121     real(dl) pows(numks), ks(
2122     integer i
2123
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 2095

```
2095      do i = 1, numks
2096          pows(i) = TensorPowe
2097          if (global_error_flag
2098      end do
2099
2100      end subroutine GetInitPow
2101
2102
2103      subroutine CalcScalCls(CT
2104      use Bispectrum
2105      implicit none
2106      Type(ClTransferData) :: C
2107      integer pix,j, q_ix, w_ix
2108      real(dl) apowers
2109      real(dl) dlnk, ell, ctnor
2110      real(dl), allocatable ::
2111
2112      allocate(ks(CTrans%q%npoi
2113
2114      do pix=1,CP%InitPower%nn
2115          do q_ix = 1, CTrans%q
2116              if (CP%flat) then
2117                  ks(q_ix) = CT
2118                  dlnks(q_ix) =
2119              else
2120                  ks(q_ix) = sq
2121                  dlnks(q_ix) =
2122              end if
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 2124

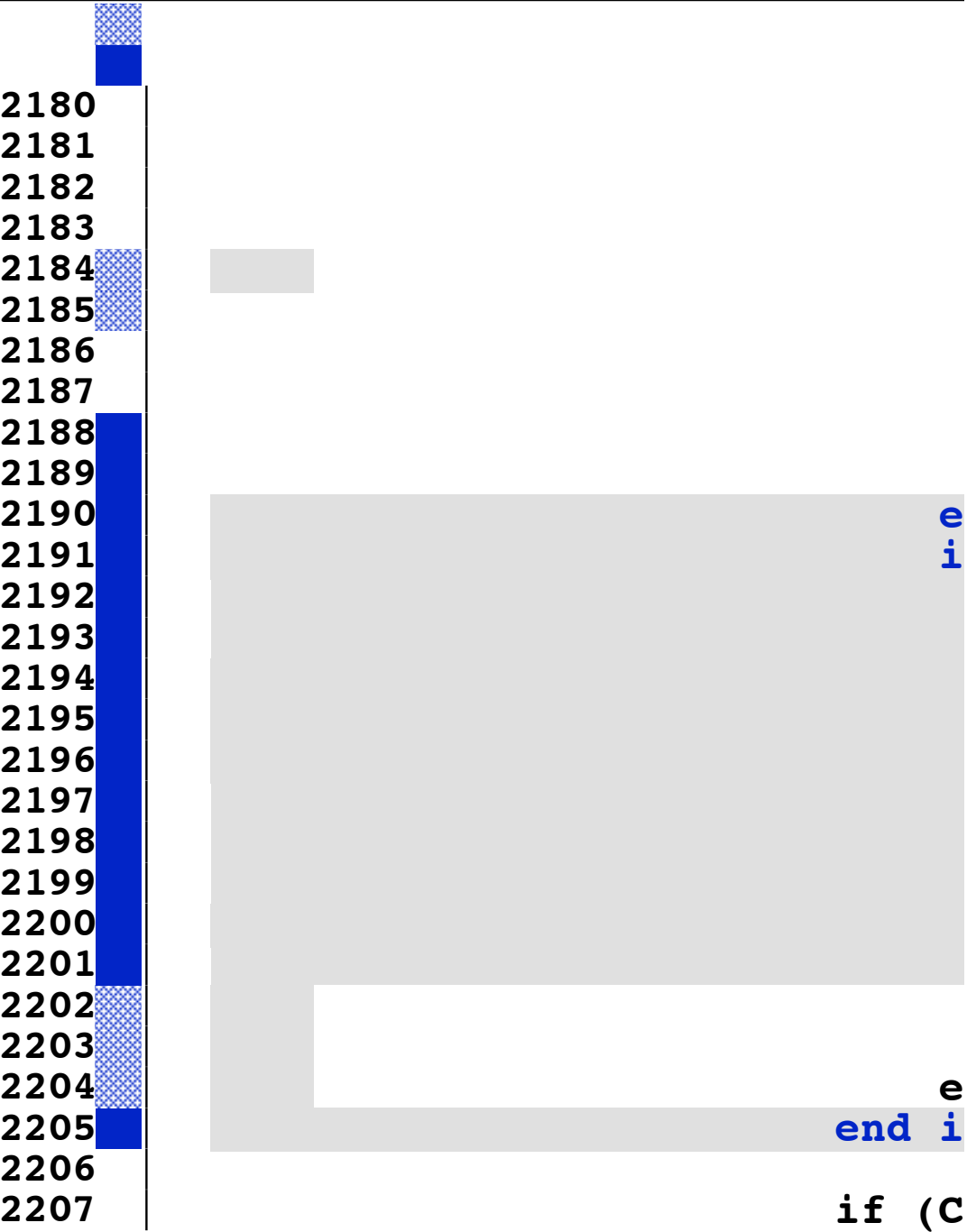
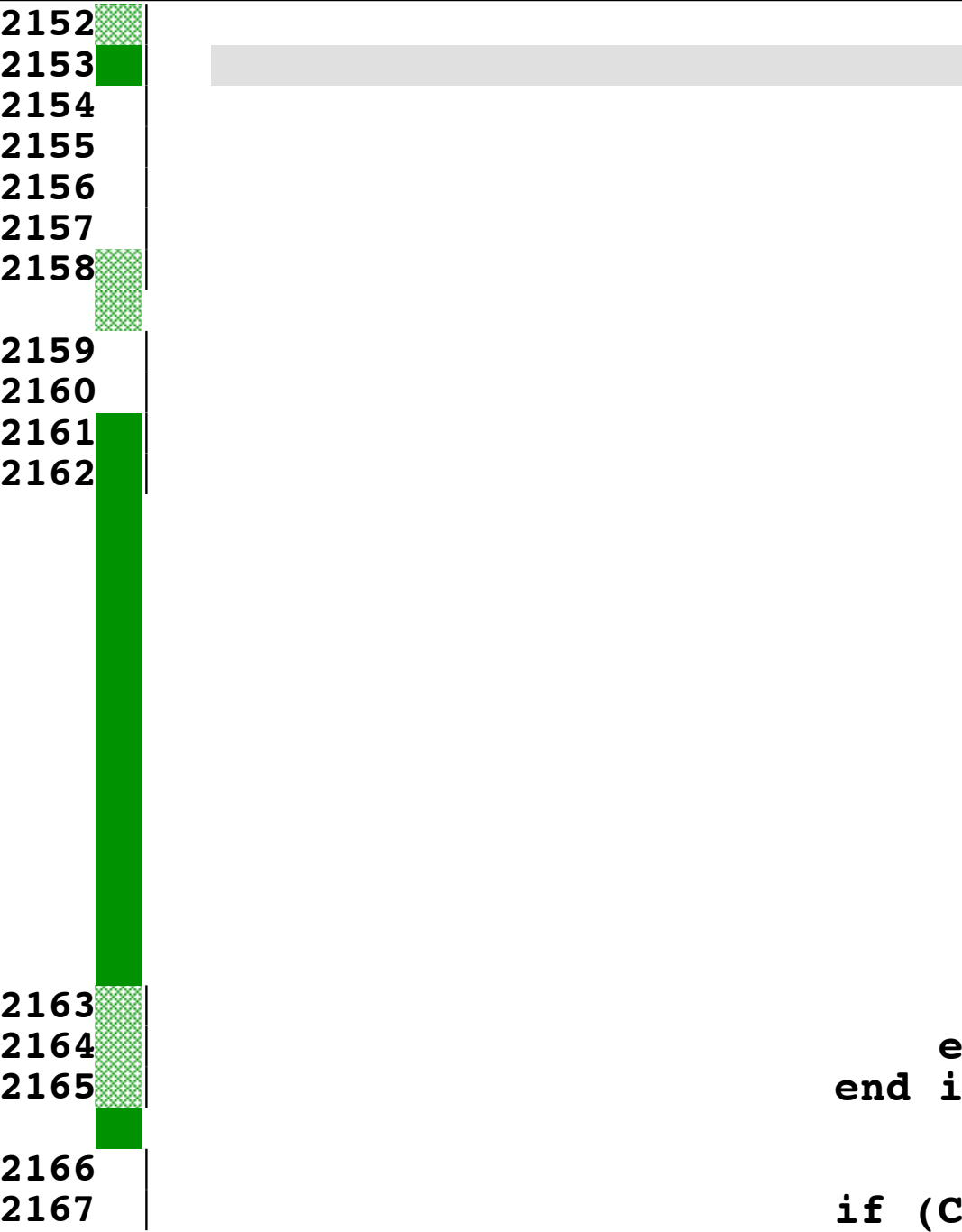
```
2124      do i = 1, numks
2125          pows(i) = TensorPowe
2126          if (global_error_flag
2127      end do
2128
2129      end subroutine GetInitPow
2130
2131
2132      subroutine CalcScalCls(CT
2133      use Bispectrum
2134      implicit none
2135      Type(ClTransferData) :: C
2136      integer pix,j, q_ix, w_ix
2137      real(dl) apowers
2138      real(dl) dlnk, ell, ctnor
2139      real(dl), allocatable ::
2140      real(dl) fac(3 + num_reds
2141      integer nscal, i
2142
2143      allocate(ks(CTrans%q%npoi
2144
2145      do pix=1,CP%InitPower%nn
2146          do q_ix = 1, CTrans%q
2147              if (CP%flat) then
2148                  ks(q_ix) = CT
2149                  dlnks(q_ix) =
2150              else
2151                  ks(q_ix) = sq
2152                  dlnks(q_ix) =
2153              end if
```


/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 2123

```
2123      pows(q_ix) = Sca
2124      if (global_error_
2125      end do
2126
2127      !Seems not to OMP wel
2128      !OMP PARALLEL DO DEFA
2129      !OMP & PRIVATE(j,q_ix
2130      do j=1,CTrans%ls%l0
2131      !Integrate dk/k D
2132      ell = real(CTrans
2133
2134      if (j<= CTrans%ma
2135      do q_ix = 1,
2136      if (.not.
2137      !cut
2138      dlnk
2139      apowe
2140      iCl_s
2141      apowe
2142      iCl_s
2143      apowe
2144
2145      if (C
2146      c
2147      d
2148
2149      d
2150
2151
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 2153

```
2153      pows(q_ix) = Sca
2154      if (global_error_
2155      end do
2156
2157      !Seems not to OMP wel
2158      !OMP PARALLEL DO DEFA
2159      !OMP & PRIVATE(j,q_ix
2160      do j=1,CTrans%ls%l0
2161      !Integrate dk/k D
2162      ell = real(CTrans
2163
2164      if (j<= CTrans%ma
2165      do q_ix = 1,
2166      if (.not.
2167      !cut
2168      dlnk
2169      apowe
2170
2171      iCl_s
2172      a
2173      iCl_s
2174      a
2175
2176      if (C
2177      d
2178
2179
```



/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 2168

```
2168                                     i
2169
2170
2171
2172
2173
2174
2175                                     e
2176                                     end i
2177                                     end if
2178                                     end do
2179
2180                                     end if !limber (j
2181
2182                                     !Output l(1+1)C_l
2183                                     ctnorm=(ell*ell-1
2184                                     dbletmp=(ell*(ell
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 2208

```
2208                                     i
2209
2210
2211
2212
2213
2214
2215                                     e
2216                                     end i
2217                                     end if
2218                                     end do
2219
2220                                     end if !limber (j
2221
2222                                     !Output l(1+1)C_l
2223                                     ctnorm=(ell*ell-1
2224                                     dbletmp=(ell*(ell
2225                                     if (CTrans%NumSou
2226                                         fac=1
2227                                         fac(2) = sqrt
2228                                         fac(3) = sqrt
2229                                         do w_ix=3 + n
2230                                             nscal= cu
2231                                             do i=1, n
2232                                                 fac(w
2233                                             end do
2234                                             fac(w_ix)
2235                                         end do
2236
2237                                     do w_ix=1,3 +
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 2185

```
2185
2186      iCl_scalar(j,C_Te
2187      iCl_scalar(j,C_E,
2188      iCl_scalar(j,C_Cr
2189      if (CTrans%NumSou
2190          iCl_scalar(j,
2191          !The lensing
2192          !We put pix e
2193          iCl_scalar(j,
2194          !Cross-correl
2195          iCl_scalar(j,
2196          !Cross-correl
2197      end if
2198  end do
2199      !OMP END PARALLEL DO
2200 end do
2201
2202 deallocate(ks,pows,dlnks)
2203
2204 end subroutine CalcScalCl
2205
2206 subroutine CalcScalCls2(C
!Calculate C_ll' for non-
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 2238

```
2238      do w_ix2=
2239          iCl_A
2240          *
2241          iCl_A
2242      end do
2243  end do
2244 end if
2245
2246      iCl_scalar(j,C_Te
2247      iCl_scalar(j,C_E,
2248      iCl_scalar(j,C_Cr
2249      if (CTrans%NumSou
2250          iCl_scalar(j,
2251          !The lensing
2252          !We put pix e
2253          iCl_scalar(j,
2254          !Cross-correl
2255          iCl_scalar(j,
2256          !Cross-correl
2257      end if
2258  end do
2259      !OMP END PARALLEL DO
2260 end do
2261
2262 deallocate(ks,pows,dlnks)
2263
2264 end subroutine CalcScalCl
2265
2266 subroutine CalcScalCls2(C
!Calculate C_ll' for non-
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 2207

```
2207      !Run with l_sample_boost=  
2208      !not used in normal CAMB  
2209      implicit none  
2210      Type(ClTransferData) :: C  
2211      integer j,j2,in  
2212      real(dl) apowers, pows(CT  
2213      integer q_ix  
2214      real(dl) ks(CTrans%q%npo  
2215      real(dl) ctnorm,dbltmp  
2216      real(dl), allocatable ::  
2217  
2218      allocate(iCl_Scalar2(CTra  
2219      iCl_scalar2 = 0  
2220  
2221      do in=1,CP%InitPower%nn  
2222          do q_ix = 1, CTrans%q  
2223              if (CP%flat) then  
2224                  ks(q_ix) = CT  
2225                  dlnks(q_ix) =  
2226              else  
2227                  ks(q_ix) = sq  
2228                  dlnks(q_ix) =  
2229              end if  
2230  
2231                  pows(q_ix) = Sca  
2232                  if (global_error_  
2233              end do  
2234  
2235      do j=1,CTrans%ls%10  
2236          do j2=1,CTrans%ls
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 2268

```
2268      !Run with l_sample_boost=  
2269      !not used in normal CAMB  
2270      implicit none  
2271      Type(ClTransferData) :: C  
2272      integer j,j2,in  
2273      real(dl) apowers, pows(CT  
2274      integer q_ix  
2275      real(dl) ks(CTrans%q%npo  
2276      real(dl) ctnorm,dbltmp  
2277      real(dl), allocatable ::  
2278  
2279      allocate(iCl_Scalar2(CTra  
2280      iCl_scalar2 = 0  
2281  
2282      do in=1,CP%InitPower%nn  
2283          do q_ix = 1, CTrans%q  
2284              if (CP%flat) then  
2285                  ks(q_ix) = CT  
2286                  dlnks(q_ix) =  
2287              else  
2288                  ks(q_ix) = sq  
2289                  dlnks(q_ix) =  
2290              end if  
2291  
2292                  pows(q_ix) = Sca  
2293                  if (global_error_  
2294              end do  
2295  
2296      do j=1,CTrans%ls%10  
2297          do j2=1,CTrans%ls
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 2237

```
2237      !Integrate dk
2238      do q_ix = 1,
2239          if (.not.
2240              !cut
2241              dlnk
2242              apowe
2243
2244              iCl_s
2245              apowe
2246              iCl_s
2247              apowe
2248          end if
2249      end do
2250
2251      !Output 1(1+1
2252
2253      !ctnorm = (CT
2254      ctnorm=(CTran
2255      ctnorm=sqrt(c
2256
2257      dbletmp=(CTra
2258      dbletmp=sqrt(
2259
2260      iCl_scalar2(j
2261      iCl_scalar2(j
2262      iCl_scalar2(j
2263      end do
2264      end do
2265      end do
2266
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 2298

```
2298      !Integrate dk
2299      do q_ix = 1,
2300          if (.not.
2301              !cut
2302              dlnk
2303              apowe
2304
2305              iCl_s
2306              a
2307              iCl_s
2308              a
2309      end if
2310      end do
2311
2312      !Output 1(1+1
2313
2314      !ctnorm = (CT
2315      ctnorm=(CTran
2316      ctnorm=sqrt(c
2317
2318      dbletmp=(CTra
2319      dbletmp=sqrt(
2320
2321      iCl_scalar2(j
2322      iCl_scalar2(j
2323      iCl_scalar2(j
2324      end do
2325      end do
2326      end do
2327
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 2267

```
2267      call CreateTxtFile('z:\cl
2268      do j=1,CTrans%ls%l0
2269          do j2=1,CTrans%ls%l0
2270              write (1,*) CTran
2271          end do
2272      end do
2273      close(1)
2274      call CreateTxtFile('cl112
2275      do j=1,999
2276          write (1,'(999E15.5) '
2277      end do
2278      stop
2279
2280      end subroutine CalcScalCl
2281
2282
2283      subroutine CalcTensCls(CT
2284      implicit none
2285      Type(ClTransferData) :: C
2286      external GetInitPowers
2287      integer in,j, q_ix
2288      real(dl) nu
2289      real(dl) apowert, measur
2290      real(dl) ctnorm,dbltmp
2291      real(dl) pows(CTrans%q%np
2292      real(dl) ks(CTrans%q%npo
2293
2294      !For tensors we want Inte
2295
2296      do in=1,CP%InitPower%nn
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 2328

```
2328      call CreateTxtFile('z:\cl
2329      do j=1,CTrans%ls%l0
2330          do j2=1,CTrans%ls%l0
2331              write (1,*) CTran
2332          end do
2333      end do
2334      close(1)
2335      call CreateTxtFile('cl112
2336      do j=1,999
2337          write (1,'(999E15.5) '
2338      end do
2339      stop
2340
2341      end subroutine CalcScalCl
2342
2343
2344      subroutine CalcTensCls(CT
2345      implicit none
2346      Type(ClTransferData) :: C
2347      external GetInitPowers
2348      integer in,j, q_ix
2349      real(dl) nu
2350      real(dl) apowert, measur
2351      real(dl) ctnorm,dbltmp
2352      real(dl) pows(CTrans%q%np
2353      real(dl) ks(CTrans%q%npo
2354
2355      !For tensors we want Inte
2356
2357      do in=1,CP%InitPower%nn
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 2297

```
2297      do q_ix = 1, CTrans%q
2298          if (CP%flat) then
2299              ks(q_ix) = CT
2300              measures(q_ix
2301          else
2302              nu = CTrans%q
2303              ks(q_ix) = sq
2304              measures(q_ix
2305          end if
2306      end do
2307
2308      call GetInitPowers(po
2309
2310      !$OMP PARALLEL DO DEF
2311      !$OMP & PRIVATE(j,q_i
2312      do j=1,CTrans%ls%10
2313          do q_ix = 1, CTra
2314              if (.not.(CP%
2315                  !cut off
2316                  apowert =
2317                  measure =
2318
2319                  iCl_tenso
2320                  apowert*C
2321
2322                  iCl_tenso
2323                  +apowert*
2324              end if
2325          end do
2326
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 2358

```
2358      do q_ix = 1, CTrans%q
2359          if (CP%flat) then
2360              ks(q_ix) = CT
2361              measures(q_ix
2362          else
2363              nu = CTrans%q
2364              ks(q_ix) = sq
2365              measures(q_ix
2366          end if
2367      end do
2368
2369      call GetInitPowers(po
2370
2371      !$OMP PARALLEL DO DEF
2372      !$OMP & PRIVATE(j,q_i
2373      do j=1,CTrans%ls%10
2374          do q_ix = 1, CTra
2375              if (.not.(CP%
2376                  !cut off
2377                  apowert =
2378                  measure =
2379
2380                  iCl_tenso
2381                  apowe
2382
2383                  iCl_tenso
2384                  +apow
2385              end if
2386          end do
2387
```


/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 2327

```
2327         ctnorm=(CTrans%ls
2328         dbletmp=(CTrans%l
2329         iCl_tensor(j, CT_
2330         if (CTrans%ls%l(j
2331         iCl_tensor(j, CT_
2332         iCl_tensor(j, CT_
2333         end do
2334     end do
2335
2336     end subroutine CalcTensCl
2337
2338
2339     subroutine CalcVecCls(CTr
2340     implicit none
2341     Type(ClTransferData) :: C
2342     external GetInitPowers
2343     integer in,j, q_ix
2344     real(dl) power, measure
2345     real(dl) ctnorm,lfac,dble
2346     real(dl) pows(CTrans%q%np
2347     real(dl) ks(CTrans%q%np
2348
2349     do in=1,CP%InitPower%nn
2350         do q_ix = 1, CTrans%q
2351             ks(q_ix) = CTrans
2352             measures(q_ix) =
2353         end do
2354
2355         call GetInitPowers(po
2356
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 2388

```
2388         ctnorm=(CTrans%ls
2389         dbletmp=(CTrans%l
2390         iCl_tensor(j, CT_
2391         if (CTrans%ls%l(j
2392         iCl_tensor(j, CT_
2393         iCl_tensor(j, CT_
2394         end do
2395     end do
2396
2397     end subroutine CalcTensCl
2398
2399
2400     subroutine CalcVecCls(CTr
2401     implicit none
2402     Type(ClTransferData) :: C
2403     external GetInitPowers
2404     integer in,j, q_ix
2405     real(dl) power, measure
2406     real(dl) ctnorm,lfac,dble
2407     real(dl) pows(CTrans%q%np
2408     real(dl) ks(CTrans%q%np
2409
2410     do in=1,CP%InitPower%nn
2411         do q_ix = 1, CTrans%q
2412             ks(q_ix) = CTrans
2413             measures(q_ix) =
2414         end do
2415
2416         call GetInitPowers(po
2417
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 2357

```
2357      !$OMP PARALLEL DO DEF
2358      !$OMP & PRIVATE(j,q_i
2359      do j=1,CTrans%ls%10
2360          do q_ix = 1, CTra
2361              if (.not.(CP%
2362                  !cut off
2363                  power = p
2364                  measure =
2365
2366                  iCl_vecto
2367                  power*CTr
2368
2369                  iCl_vecto
2370                  +power*CT
2371              end if
2372          end do
2373
2374          ctnorm=CTrans%ls%
2375          dbletmp=(CTrans%l
2376          iCl_vector(j, CT_
2377          lfac = (CTrans%ls
2378          iCl_vector(j, CT_
2379          iCl_vector(j, CT_
2380      end do
2381  end do
2382
2383  end subroutine CalcVecCls
2384
2385
2386  subroutine InterpolateCls
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 2418

```
2418      !$OMP PARALLEL DO DEF
2419      !$OMP & PRIVATE(j,q_i
2420      do j=1,CTrans%ls%10
2421          do q_ix = 1, CTra
2422              if (.not.(CP%
2423                  !cut off
2424                  power = p
2425                  measure =
2426
2427                  iCl_vecto
2428                  power
2429
2430                  iCl_vecto
2431                  +powe
2432              end if
2433          end do
2434
2435          ctnorm=CTrans%ls%
2436          dbletmp=(CTrans%l
2437          iCl_vector(j, CT_
2438          lfac = (CTrans%ls
2439          iCl_vector(j, CT_
2440          iCl_vector(j, CT_
2441      end do
2442  end do
2443
2444  end subroutine CalcVecCls
2445
2446
2447  subroutine InterpolateCls
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 2387

```
2387      implicit none
2388      Type(ClTransferData) :: C
2389      integer in,i,j
2390      integer, dimension(2,2),
2391      !use verbose form above f
2392
2393      !Note using log interpola
2394
2395      !$OMP PARALLEL DO DEFAULT
2396      do in=1,CP%InitPower%nn
2397          if (CP%WantScalars) t
2398              do i = C_Temp, C_
2399                  call Interpol
2400                      CTransS%ls%10
2401              end do
2402
2403              if (CTransScal%Nu
2404                  do i=1,3+num_
2405                      do j=i,3+
2406                          if (i
2407                              C
2408                          else
2409                              c
2410                              C
2411                          end i
2412                          if (i
2413                              end do
2414                      end do
2415                  end if
2416              end if
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 2448

```
2448      implicit none
2449      Type(ClTransferData) :: C
2450      integer in,i,j
2451      integer, dimension(2,2),
2452      !use verbose form above f
2453
2454      !Note using log interpola
2455
2456      !$OMP PARALLEL DO DEFAULT
2457      do in=1,CP%InitPower%nn
2458          if (CP%WantScalars) t
2459              do i = C_Temp, C_
2460                  call Interpol
2461                      CTransS%l
2462              end do
2463
2464              if (CTransScal%Nu
2465                  do i=1,3+num_
2466                      do j=i,3+
2467                          if (i
2468                              C
2469                          else
2470                              c
2471                              C
2472                          end i
2473                          if (i
2474                              end do
2475                      end do
2476                  end if
2477              end if
```

/Users/lp1opa/Compare/camb_simdata/cmbmain.f90, Top line: 2417

```
2417
2418         if (CP%WantVectors) t
2419             do i = C_Temp, CT
2420                 call Interpol
2421             end do
2422         end if
2423
2424         if (CP%WantTensors) t
2425             do i = CT_Temp, C
2426                 call Interpol
2427             end do
2428         end if
2429     end do
2430     !$OMP END PARALLEL DO
2431 end subroutine Interpolat
2432
2433
2434 end module CAMBmain
2435
```

/Users/lp1opa/Compare/camb_des/cmbmain.f90, Top line: 2478

```
2478
2479         if (CP%WantVectors) t
2480             do i = C_Temp, CT
2481                 call Interpol
2482             end do
2483         end if
2484
2485         if (CP%WantTensors) t
2486             do i = CT_Temp, C
2487                 call Interpol
2488             end do
2489         end if
2490     end do
2491     !$OMP END PARALLEL DO
2492 end subroutine Interpolat
2493
2494
2495 end module CAMBmain
2496
```