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

```


    ! Equations module for da
    ! allowing for perturbati
    ! by Antony Lewis (http:/

    ! Dec 2003, fixed (fatal)
    ! Changes to tight coupli
    ! June 2004, fixed proble
    ! Generate vector modes o
    ! August 2004, fixed reio
    ! Nov 2004, change massiv
    ! Apr 2005, added DoLateR
    ! June 2006, added suppor
    ! Nov 2006, tweak to high
    ! June 2011, improved rad
    !           merged fderi
    !           optimized ne
    ! Feb 2013: fixed various
    ! Mar 2014: fixes for ten

    module LambdaGeneral
    use precision

!#SimDataAdd
    use ModelParams
!#SimDataAdd

    implicit none

```

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

```
    ! Equations module for da
    ! allowing for perturbati
    ! by Antony Lewis (http:/

    ! Dec 2003, fixed (fatal)
    ! Changes to tight coupli
    ! June 2004, fixed proble
    ! Generate vector modes o
    ! August 2004, fixed reio
    ! Nov 2004, change massiv
    ! Apr 2005, added DoLateR
    ! June 2006, added suppor
    ! Nov 2006, tweak to high
    ! June 2011, improved rad
    !           merged fderi
    !           optimized ne
    ! Feb 2013: fixed various
    ! Mar 2014: fixes for ten

    module LambdaGeneral
    use precision

    implicit none

```

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

```
0031 !#SimDataNoUse
0032 !     real(dl)   :: w_lam = -1_
0033 !     real(dl) :: cs2_lam = 1_
0034     !comoving sound speed. Al
0035     !(otherwise assumed const
0036
0037 !de vazut ce e cu asta, in ve
0038     real(dl), parameter :: wa
0039
0040 !     logical :: w_perturb = .
0041     !If you are tempted to se
0042     ! http://cosmocoffee.info
0043     ! http://cosmocoffee.info
0044 !#SimDataNoUse
0045
0046 !#SimDataAdd
0047     logical :: is_cosmologica
0048 !#SimDataAdd
0049
0050     contains
0051 !#SimDataNoUse
0052 !     subroutine DarkEnergy_Re
0053 !     use IniFile
0054 !     Type(TIniFile) :: Ini
0055
0056 !      w_lam = Ini_Read_Double_
0057 !      cs2_lam = Ini_Read_Doubl
0058 !     end subroutine DarkEnerg
0059 !#SimDataNoUse
0060
```

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

```
0024     real(dl)   :: w_lam = -1_d
0025     real(dl) :: cs2_lam = 1_d

0026     !comoving sound speed. Al
0027     !(otherwise assumed const
0028

0029     real(dl), parameter :: wa
0030
0031     logical :: w_perturb = .t
0032     !If you are tempted to se
0033     ! http://cosmocoffee.info
0034     ! http://cosmocoffee.info





0035
0036     contains




0037
0038     subroutine DarkEnergy_Rea
0039     use IniFile
0040     Type(TIniFile) :: Ini
```

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

```
0061 !#SimDataAdd
0062     function w_de(a)
0063     real(dl) :: w_de
0064     real(dl), intent(IN) :: a
0065
0066     w_de = CP%w0+CP%wa*(1.d0-
0067     end function w_de
0068
0069     function grho_de(a)
0070     real(dl) :: grho_de
0071     real(dl), intent(IN) :: a
0072
0073     grho_de = grhov*a**(1.d0-
0074     end function grho_de
0075 !#SimDataAdd
0076
0077     end module LambdaGeneral
0078
0079
0080
0081     !Return OmegaK - modify t
0082     function GetOmegak()
0083     use precision
0084     use ModelParams
0085     real(dl)  GetOmegak
0086     GetOmegak = 1 - (CP%omega
0087
0088     end function GetOmegak
0089
0090
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 41

```
0041
0042     w_lam = Ini_Read_Double_F
0043     cs2_lam = Ini_Read_Double
0044
0045     end subroutine DarkEnergy
0046
0047     end module LambdaGeneral
0048
0049
0050
0051     !Return OmegaK - modify t
0052     function GetOmegak()
0053     use precision
0054     use ModelParams
0055     real(dl)  GetOmegak
0056     GetOmegak = 1 - (CP%omega
0057
0058     end function GetOmegak
0059
0060
```

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

```
    subroutine init_backgroun
    !This is only called once
    !It is called before firs
    !massive neutrinos are in


!#SimDataAdd
    use LambdaGeneral
    is_cosmological_constant
!#SimDataAdd

     end  subroutine init_bac


    !Background evolution
    function dtauda(a)
    !get d tau / d a
    use precision
    use ModelParams
    use MassiveNu
    use LambdaGeneral
    implicit none
    real(dl) dtauda
    real(dl), intent(IN) :: a
    real(dl) rhonu,grhoa2, a2
    integer nu_i

    a2=a**2

    !  8*pi*G*rho*a**4.
```

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

```
    subroutine init_backgroun
    !This is only called once
    !It is called before firs
    !massive neutrinos are in
    end  subroutine init_back


    !Background evolution
    function dtauda(a)
    !get d tau / d a
    use precision
    use ModelParams
    use MassiveNu
    use LambdaGeneral
    implicit none
    real(dl) dtauda
    real(dl), intent(IN) :: a
    real(dl) rhonu,grhoa2, a2
    integer nu_i

    a2=a**2

    !  8*pi*G*rho*a**4.
```

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

```
0121 !#SimDataReplace
0122     grhoa2=grhok*a2+(grhoc+gr
0123 !#SimDataReplace
0124 !    grhoa2=grhok*a2+(grhoc+g
0125
0126 !    if (w_lam == -1._dl) the
0127 !        grhoa2=grhoa2+grhov*
0128 !    else
0129 !        grhoa2=grhoa2+grhov*
0130 !    end if
0131
0132
0133 !#SimDataReplace
0134
0135
0136     if (CP%Num_Nu_massive /=
0137         !Get massive neutrino
0138         do nu_i = 1, CP%nu_ma
0139             call Nu_rho(a*nu_
0140             grhoa2=grhoa2+rho
0141         end do
0142     end if
0143
0144     dtauda=sqrt(3/grhoa2)
0145
0146     end function dtauda
0147
0148     !cccccccccccccccccccccccc
0149
0150     !Gauge-dependent perturba
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 84

```
0084     grhoa2=grhok*a2+(grhoc+gr
0085     if (w_lam == -1._dl) then
0086         grhoa2=grhoa2+grhov*a
0087     else
0088         grhoa2=grhoa2+grhov*a
0089     end if
0090     if (CP%Num_Nu_massive /=
0091         !Get massive neutrino
0092         do nu_i = 1, CP%nu_ma
0093             call Nu_rho(a*nu_
0094             grhoa2=grhoa2+rho
0095         end do
0096     end if
0097
0098     dtauda=sqrt(3/grhoa2)
0099
0100     end function dtauda
0101
0102     !cccccccccccccccccccccccc
0103
0104     !Gauge-dependent perturba
```

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

```

    module GaugeInterface
    use precision
    use ModelParams
    use MassiveNu
    use LambdaGeneral
    use Errors
    use Transfer
    implicit none
    public

    !Description of this file

!#SimDataReplace
!     character(LEN=*), parame

    character(LEN=*), paramet
!#SimDataReplace

    integer, parameter :: bas

    logical :: DoTensorNeutri

    logical :: DoLateRadTrunc
    !if true, use smooth appr
    !small scales, saving evo

    logical, parameter :: sec

    real(dl) :: Magnetic = 0.
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 105

```

    module GaugeInterface
    use precision
    use ModelParams
    use MassiveNu
    use LambdaGeneral
    use Errors
    use Transfer
    implicit none
    public

    !Description of this file
    character(LEN=*), paramet






    integer, parameter :: bas

    logical :: DoTensorNeutri

    logical :: DoLateRadTrunc
    !if true, use smooth appr
    !small scales, saving evo

    logical, parameter :: sec

    real(dl) :: Magnetic = 0.
```

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

```
    !Vector mode anisotropic
    real(dl) :: vec_sig0 = 1.
    !Vector mode shear
    integer, parameter :: max
    !Note higher values incre

    !Supported scalar initial
    integer, parameter :: ini
    initial_iso_baryon=3,  in
    integer, parameter :: ini

    type EvolutionVars
        real(dl) q, q2
        real(dl) k_buf,k2_buf

        integer w_ix !Index o
        integer r_ix !Index o
        integer g_ix !Index o

        integer q_ix !index i
        logical TransferOnly

        !       nvar  - numbe
        integer nvar,nvart, n

        !Max_l for the variou
        integer lmaxg,lmaxnr,
        integer lmaxnrt, lmax
        logical EvolveTensorM
        integer lmaxnrv, lmax
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 130

```
    !Vector mode anisotropic
    real(dl) :: vec_sig0 = 1.
    !Vector mode shear
    integer, parameter :: max
    !Note higher values incre

    !Supported scalar initial
    integer, parameter :: ini
        initial_iso_baryon=3,
    integer, parameter :: ini

    type EvolutionVars
        real(dl) q, q2
        real(dl) k_buf,k2_buf

        integer w_ix !Index o
        integer r_ix !Index o
        integer g_ix !Index o

        integer q_ix !index i
        logical TransferOnly

        !       nvar  - numbe
        integer nvar,nvart, n

        !Max_l for the variou
        integer lmaxg,lmaxnr,
        integer lmaxnrt, lmax
        logical EvolveTensorM
        integer lmaxnrv, lmax
```

```

          integer polind  !inde

          !array indices for ma
          integer nu_ix(max_nu)
          integer nq(max_nu), l
          logical has_nu_relati

          !Initial values for m
          !to non-relativistic
          real(dl) G11(max_nu),
          !True when using non-
          logical MassiveNuAppr
          real(dl) MassiveNuApp

          !True when truncating
          logical high_ktau_neu

          !Massive neutrino sch
          integer NuMethod

          !True when using tigh
          logical TightCoupling
          real(dl) TightSwitcho

          !Numer of scalar equa
          integer ScalEqsToProp
          integer TensEqsToProp
          !beta > l for closed
          integer FirstZerolFor
```

```

          integer polind  !inde

          !array indices for ma
          integer nu_ix(max_nu)
          integer nq(max_nu), l
          logical has_nu_relati

          !Initial values for m
          !to non-relativistic
          real(dl) G11(max_nu),
          !True when using non-
          logical MassiveNuAppr
          real(dl) MassiveNuApp

          !True when truncating
          logical high_ktau_neu

          !Massive neutrino sch
          integer NuMethod

          !True when using tigh
          logical TightCoupling
          real(dl) TightSwitcho

          !Numer of scalar equa
          integer ScalEqsToProp
          integer TensEqsToProp
          !beta > l for closed
          integer FirstZerolFor
```

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

```
        !Tensor vars
        real(dl) aux_buf

        real(dl) pig, pigdot
        real(dl) poltruncfac

        logical no_nu_multpol
        integer lmaxnu_tau(ma
        logical nu_nonrelativ

        real(dl) denlk(max_l_
        real(dl) Kf(max_l_evo

        integer E_ix, B_ix !t
        real(dl) denlkt(4,max
        real, pointer :: Outp



    end type EvolutionVars

```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 190

```
        !Tensor vars
        real(dl) aux_buf

        real(dl) pig, pigdot
        real(dl) poltruncfac

        logical no_nu_multpol
        integer lmaxnu_tau(ma
        logical nu_nonrelativ

        real(dl) denlk(max_l_
        real(dl) Kf(max_l_evo

        integer E_ix, B_ix !t
        real(dl) denlkt(4,max
        real, pointer :: Outp
        real(dl), pointer ::
        real(dl), pointer ::

    end type EvolutionVars

    ABSTRACT INTERFACE
    SUBROUTINE TSource_func(s
        grhob_t,grhor_t,grhoc
        k,etak, etakdot, phi,
        dgrho, clxg,clxb,clxc
        dgq, qg, qr, qde, vb,
        dgpi, pig, pir, pigdo
        polter, polterdot, po
        opacity, dopacity, dd
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 259

```
      !precalculated arrays
      real(dl) polfac(max_l_evo

      real(dl), parameter :: ep
      integer, parameter :: lma

      real(dl) epsw
      real(dl) nu_tau_notmassle
      contains


      subroutine GaugeInterface
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 220

```
          tau0, tau_maxvis, Kf,
      real*8, intent(out) :: so
      real*8, intent(in) :: tau
          grhob_t,grhor_t,grhoc
          k,etak, etakdot, phi,
          dgrho, clxg,clxb,clxc
          dgq, qg, qr, qde, vb,
          dgpi, pig, pir, pigdo
          polter, polterdot, po
          opacity, dopacity, dd
          tau0, tau_maxvis
      REAL*8, intent(in) :: Kf(
      real*8, external :: f_K
      END SUBROUTINE TSource_fu
      END INTERFACE

      procedure(TSource_func),

      !precalculated arrays
      real(dl) polfac(max_l_evo

      real(dl), parameter :: ep
      integer, parameter :: lma

      real(dl) epsw
      real(dl) nu_tau_notmassle
      contains


      subroutine GaugeInterface
```

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

```
      type(EvolutionVars) EV
      real(dl) c(24),w(EV%nvar,
      integer ind

      call dverk(EV,EV%ScalEqsT
      if (ind==-3) then
          call GlobalError('Dve
          //'requirement  with
          //'equal to hmin, whi
          //'--- but most likel
          //'compiling with bou
      end if
      end subroutine GaugeInter

      function next_nu_nq(nq) r
      integer, intent(in) :: nq
      integer q, next_nq

      if (nq==0) then
          next_nq=1
      else
          q = nu_q(nq)
          if (q>=10) then
              next_nq = nqmax
          else
              next_nq = nq+1
          end if
      end if

      end function next_nu_nq
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 250

```
      type(EvolutionVars) EV
      real(dl) c(24),w(EV%nvar,
      integer ind

      call dverk(EV,EV%ScalEqsT
      if (ind==-3) then
          call GlobalError('Dve
              //'requirement  w
              //'equal to hmin,
              //'--- but most l
              //'compiling with
      end if
      end subroutine GaugeInter

      function next_nu_nq(nq) r
      integer, intent(in) :: nq
      integer q, next_nq

      if (nq==0) then
          next_nq=1
      else
          q = nu_q(nq)
          if (q>=10) then
              next_nq = nqmax
          else
              next_nq = nq+1
          end if
      end if

      end function next_nu_nq
```

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

```

      recursive subroutine Gaug
      use ThermoData
      type(EvolutionVars) EV, E
      real(dl) c(24),w(EV%nvar,
      integer ind, nu_i
      real(dl) cs2, opacity, do
      real(dl) tau_switch_ktau,
      real(dl) tau_switch_no_nu
      real(dl) noSwitch, smallT

      noSwitch= CP%tau0+1
      smallTime =  min(tau, 1/E

      tau_switch_ktau = noSwitc
      tau_switch_no_nu_multpole
      tau_switch_no_phot_multpo

      !Massive neutrino switche
      tau_switch_nu_massless =
      tau_switch_nu_nonrel = no
      tau_switch_nu_massive= no

      !Evolve equations from ta

      if (.not. EV%high_ktau_ne
          tau_switch_ktau=  max
      end if

      if (CP%Num_Nu_massive /=
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 280

```

      recursive subroutine Gaug
      use ThermoData
      type(EvolutionVars) EV, E
      real(dl) c(24),w(EV%nvar,
      integer ind, nu_i
      real(dl) cs2, opacity, do
      real(dl) tau_switch_ktau,
      real(dl) tau_switch_no_nu
      real(dl) noSwitch, smallT

      noSwitch= CP%tau0+1
      smallTime =  min(tau, 1/E

      tau_switch_ktau = noSwitc
      tau_switch_no_nu_multpole
      tau_switch_no_phot_multpo

      !Massive neutrino switche
      tau_switch_nu_massless =
      tau_switch_nu_nonrel = no
      tau_switch_nu_massive= no

      !Evolve equations from ta

      if (.not. EV%high_ktau_ne
          tau_switch_ktau=  max
      end if

      if (CP%Num_Nu_massive /=
```

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

```
0331          do nu_i = 1, CP%Nu_ma
0332              if (EV%nq(nu_i) /
0333                  tau_switch_nu
0334              else if (.not. EV
0335                  tau_switch_nu
0336              else if (EV%NuMet
0337                  tau_switch_nu
0338              end if
0339          end do
0340      end if
0341
0342      if (DoLateRadTruncation)
0343          if (.not. EV%no_nu_mu
0344          tau_switch_no_nu_mult
0345
0346          if (.not. EV%no_phot_
0347          tau_switch_no_phot_mu
0348      end if
0349
0350      next_switch = min(tau_swi
0351      tau_switch_no_nu_multpole
0352
0353      if (next_switch < tauend)
0354          if (next_switch > tau
0355              call GaugeInterfa
0356              if (global_error_
0357          end if
0358
0359          EVout=EV
0360
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 310

```
0310          do nu_i = 1, CP%Nu_ma
0311              if (EV%nq(nu_i) /
0312                  tau_switch_nu
0313              else if (.not. EV
0314                  tau_switch_nu
0315              else if (EV%NuMet
0316                  tau_switch_nu
0317              end if
0318          end do
0319      end if
0320
0321      if (DoLateRadTruncation)
0322          if (.not. EV%no_nu_mu
0323              tau_switch_no_nu_
0324
0325          if (.not. EV%no_phot_
0326              tau_switch_no_pho
0327      end if
0328
0329      next_switch = min(tau_swi
0330          tau_switch_no_nu_mult
0331
0332      if (next_switch < tauend)
0333          if (next_switch > tau
0334              call GaugeInterfa
0335              if (global_error_
0336          end if
0337
0338          EVout=EV
0339
```

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

```
          if (next_switch == EV
              !TightCoupling
              EVout%TightCoupli
              EVout%TightSwitch
              call SetupScalarA
              call CopyScalarVa
              EV=EVout
              y=yout
              ind=1
              !Set up variables
              y(EV%g_ix+2) = EV
              call thermo(tau,c

              if (second_order_
                  ! Francis-Yan

                  y(EV%g_ix+3)
                  (3._dl/7._dl)

                  y(EV%polind+2
                  (25._dl/16._d
                  EV%pig*(EV%k_
                  y(EV%polind+3
                  dopacity/opac
                  (1._dl+(5._dl
              else
                  y(EV%g_ix+3)
                  y(EV%polind+2
                  y(EV%polind+3
              end if
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 340

```
          if (next_switch == EV
              !TightCoupling
              EVout%TightCoupli
              EVout%TightSwitch
              call SetupScalarA
              call CopyScalarVa
              EV=EVout
              y=yout
              ind=1
              !Set up variables
              y(EV%g_ix+2) = EV
              call thermo(tau,c

              if (second_order_
                  ! Francis-Yan

                  y(EV%g_ix+3)
                      (3._dl/7.

                  y(EV%polind+2
                      (25._dl/1
                      EV%pig*(E
                  y(EV%polind+3
                      dopacity/
                      (1._dl+(5
              else
                  y(EV%g_ix+3)
                  y(EV%polind+2
                  y(EV%polind+3
              end if
```

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

```
0391          else if (next_switch=
0392              !k tau >> 1, evol
0393              EVout%high_ktau_n
0394              EV%nq(1:CP%Nu_mas
0395              call SetupScalarA
0396              call CopyScalarVa
0397              y=yout
0398              EV=EVout
0399          else if (next_switch
0400              !Mass starts to b
0401              do nu_i = 1, CP%N
0402                  if (EV%nq(nu_
0403                  next_switch =
0404                      EVOut%nq(
0405                      call Setu
0406                      call Copy
0407                      EV=EVout
0408                      y=yout
0409                      exit
0410                  end if
0411              end do
0412          else if (next_switch
0413              !Neutrino becomes
0414              do nu_i = 1, CP%N
0415                  if (.not. EV%
0416                      EVout%nu_
0417                      call Setu
0418                      call Copy
0419                      EV=EVout
0420                      y=yout
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 370

```
0370          else if (next_switch=
0371              !k tau >> 1, evol
0372              EVout%high_ktau_n
0373              EVout%nq(1:CP%Nu_
0374              call SetupScalarA
0375              call CopyScalarVa
0376              y=yout
0377              EV=EVout
0378          else if (next_switch
0379              !Mass starts to b
0380              do nu_i = 1, CP%N
0381                  if (EV%nq(nu_
0382                      next_swit
0383                  EVOut%nq(nu_i
0384                  call SetupSca
0385                  call CopyScal
0386                  EV=EVout
0387                  y=yout
0388                  exit
0389                  end if
0390              end do
0391          else if (next_switch
0392              !Neutrino becomes
0393              do nu_i = 1, CP%N
0394                  if (.not. EV%
0395                      EVout%nu_
0396                      call Setu
0397                      call Copy
0398                      EV=EVout
0399                      y=yout
```

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

```
0421                        exit
0422                    end if
0423                end do
0424            else if (next_switch
0425                !Very non-relativ
0426                do nu_i = 1, CP%N
0427                    if (.not. EV%
0428                        call Swit
0429                        exit
0430                    end if
0431                end do
0432            else if (next_switch=
0433                !Turn off neutrin
0434                ind=1
0435                EVout%no_nu_multp
0436                EVOut%nq(1:CP%Nu_
0437                call SetupScalarA
0438                call CopyScalarVa
0439                y=yout
0440                EV=EVout
0441            else if (next_switch=
0442                !Turn off photon
0443                ind=1
0444                EVout%no_phot_mul
0445                call SetupScalarA
0446                call CopyScalarVa
0447                y=yout
0448                EV=EVout
0449            end if
0450
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 400

```
0400                        exit
0401                    end if
0402                end do
0403            else if (next_switch
0404                !Very non-relativ
0405                do nu_i = 1, CP%N
0406                    if (.not. EV%
0407                        call Swit
0408                        exit
0409                    end if
0410                end do
0411            else if (next_switch=
0412                !Turn off neutrin
0413                ind=1
0414                EVout%no_nu_multp
0415                EVOut%nq(1:CP%Nu_
0416                call SetupScalarA
0417                call CopyScalarVa
0418                y=yout
0419                EV=EVout
0420            else if (next_switch=
0421                !Turn off photon
0422                ind=1
0423                EVout%no_phot_mul
0424                call SetupScalarA
0425                call CopyScalarVa
0426                y=yout
0427                EV=EVout
0428            end if
0429
```

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

```
            call GaugeInterface_E
            return
        end if

        call GaugeInterface_ScalE

        end subroutine GaugeInter

        subroutine GaugeInterface
        use ThermoData
        type(EvolutionVars) EV, E
        real(dl) c(24),w(EV%nvart
        integer ind
        real(dl) opacity, cs2

        if (EV%TensTightCoupling
            if (EV%TightSwitchoff
                call dverk(EV,EV%
            end if
            EVOut=EV
            EVOut%TensTightCoupli
            call SetupTensorArray
            call CopyTensorVariab
            Ev = EvOut
            y=yout
            call thermo(tau,cs2,o
            y(EV%g_ix+2)= 32._dl/
            y(EV%E_ix+2) = y(EV%g
        end if

```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 430

```
            call GaugeInterface_E
            return
        end if

        call GaugeInterface_ScalE

        end subroutine GaugeInter

        subroutine GaugeInterface
        use ThermoData
        type(EvolutionVars) EV, E
        real(dl) c(24),w(EV%nvart
        integer ind
        real(dl) opacity, cs2

        if (EV%TensTightCoupling
            if (EV%TightSwitchoff
                call dverk(EV,EV%
            end if
            EVOut=EV
            EVOut%TensTightCoupli
            call SetupTensorArray
            call CopyTensorVariab
            Ev = EvOut
            y=yout
            call thermo(tau,cs2,o
            y(EV%g_ix+2)= 32._dl/
            y(EV%E_ix+2) = y(EV%g
        end if

```

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

```
    call dverk(EV,EV%TensEqsT


    end subroutine GaugeInter

    function DeltaTimeMaxed(a
    real(dl) a1,a2,t
    real(dl), optional :: tol
    if (a1>1._dl) then
        t=0
    elseif (a2 > 1._dl) then
        t = DeltaTime(a1,1.01
    else
        t= DeltaTime(a1,a2, t
    end if
    end function DeltaTimeMax

    subroutine GaugeInterface
    !Precompute various array
    integer j, nu_i
    real(dl) a_nonrel, a_mass

    epsw = 100/CP%tau0

    if (CP%WantScalars) then
        do j=2,max_l_evolve
            polfac(j)=real((j
        end do
    end if

```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 460

```
    call dverk(EV,EV%TensEqsT


    end subroutine GaugeInter

    function DeltaTimeMaxed(a
    real(dl) a1,a2,t
    real(dl), optional :: tol
    if (a1>1._dl) then
        t=0
    elseif (a2 > 1._dl) then
        t = DeltaTime(a1,1.01
    else
        t= DeltaTime(a1,a2, t
    end if
    end function DeltaTimeMax

    subroutine GaugeInterface
    !Precompute various array
    integer j, nu_i
    real(dl) a_nonrel, a_mass

    epsw = 100/CP%tau0

    if (CP%WantScalars) then
        do j=2,max_l_evolve
            polfac(j)=real((j
        end do
    end if

```

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

```
0511        if (CP%WantVectors) then
0512            do j=2,max_l_evolve
0513                vecfac(j)=real((j
0514                vecfacpol(j)=real
0515            end do
0516        end if
0517
0518        do j=1,max_l_evolve
0519            denl(j)=1._dl/(2*j+1)
0520        end do
0521
0522        do nu_i=1, CP%Nu_Mass_eig
0523            nu_mass = max(0.1_dl,
0524            a_mass =  1.e-1_dl/nu
0525            !if (HighAccuracyDefa
0526            time=DeltaTime(0._dl,
0527            nu_tau_notmassless(1,
0528            do j=2,nqmax
0529                !times when each
0530                time= time + Delt
0531                nu_tau_notmassles
0532            end do
0533
0534            a_nonrel =  2.5d0/nu_
0535            nu_tau_nonrelativisti
0536            a_massive =  17.d0/nu
0537            nu_tau_massive(nu_i)
0538        end do
0539
0540        end subroutine GaugeInter
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 490

```
0490        if (CP%WantVectors) then
0491            do j=2,max_l_evolve
0492                vecfac(j)=real((j
0493                vecfacpol(j)=real
0494            end do
0495        end if
0496
0497        do j=1,max_l_evolve
0498            denl(j)=1._dl/(2*j+1)
0499        end do
0500
0501        do nu_i=1, CP%Nu_Mass_eig
0502            nu_mass = max(0.1_dl,
0503            a_mass =  1.e-1_dl/nu
0504            !if (HighAccuracyDefa
0505            time=DeltaTime(0._dl,
0506            nu_tau_notmassless(1,
0507            do j=2,nqmax
0508                !times when each
0509                time= time + Delt
0510                nu_tau_notmassles
0511            end do
0512
0513            a_nonrel =  2.5d0/nu_
0514            nu_tau_nonrelativisti
0515            a_massive =  17.d0/nu
0516            nu_tau_massive(nu_i)
0517        end do
0518
0519        end subroutine GaugeInter
```

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

```


        subroutine SetupScalarArr
        !Set up array indices aft
        use MassiveNu
        !Set the numer of equatio
        type(EvolutionVars) EV
        integer, intent(out), opt
        integer neq, maxeq, nu_i

        neq=basic_num_eqns
        maxeq=neq
        if (.not. EV%no_phot_mult
            !Photon multipoles
            EV%g_ix=basic_num_eqn
            if (EV%TightCoupling)
                neq=neq+2
            else
                neq = neq+ (EV%lm
                !Polarization mul
                EV%polind = neq -
                neq=neq + EV%lmax
            end if
        end if
        if (.not. EV%no_nu_multpo
            !Massless neutrino mu
            EV%r_ix= neq+1
            if (EV%high_ktau_neut
                neq=neq + 3
            else
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 520

```


        subroutine SetupScalarArr
        !Set up array indices aft
        use MassiveNu
        !Set the numer of equatio
        type(EvolutionVars) EV
        integer, intent(out), opt
        integer neq, maxeq, nu_i

        neq=basic_num_eqns
        maxeq=neq
        if (.not. EV%no_phot_mult
            !Photon multipoles
            EV%g_ix=basic_num_eqn
            if (EV%TightCoupling)
                neq=neq+2
            else
                neq = neq+ (EV%lm
                !Polarization mul
                EV%polind = neq -
                neq=neq + EV%lmax
            end if
        end if
        if (.not. EV%no_nu_multpo
            !Massless neutrino mu
            EV%r_ix= neq+1
            if (EV%high_ktau_neut
                neq=neq + 3
            else
```

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

```
            neq=neq + (EV%lma
        end if
    end if
    maxeq = maxeq +  (EV%lmax

!#SimDataReplace
!Dark energy
!    if (w_lam /= -1 .and. w_
 if (.not. is_cosmological_co
        EV%w_ix = neq+1
        neq=neq+2
        maxeq=maxeq+2
    else
        EV%w_ix=0
    end if
!#SimDataReplace

    !Massive neutrinos
    if (CP%Num_Nu_massive /=
        EV%has_nu_relativisti
        if (EV%has_nu_relativ
            EV%lmaxnu_pert=EV
            EV%nu_pert_ix=neq
            neq = neq+ EV%lma
            maxeq=maxeq+ EV%l
        else
            EV%lmaxnu_pert=0
        end if

        do nu_i=1, CP%Nu_Mass
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 550

```
            neq=neq + (EV%lma
        end if
    end if
    maxeq = maxeq +  (EV%lmax

    !Dark energy
    if (w_lam /= -1 .and. w_P
        EV%w_ix = neq+1
        neq=neq+2
        maxeq=maxeq+2
    else
        EV%w_ix=0
    end if

    !Massive neutrinos
    if (CP%Num_Nu_massive /=
        EV%has_nu_relativisti
        if (EV%has_nu_relativ
            EV%lmaxnu_pert=EV
            EV%nu_pert_ix=neq
            neq = neq+ EV%lma
            maxeq=maxeq+ EV%l
        else
            EV%lmaxnu_pert=0
        end if

        do nu_i=1, CP%Nu_Mass
```

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

```
              if (EV%high_ktau_
                  if (HighAccur
                      EV%lmaxnu
                  else
                      EV%lmaxnu
                  end if
              else
                  EV%lmaxnu_tau
                  !!!Feb13tweak
                  if (EV%nu_non
              end if
              EV%lmaxnu_tau(nu_

              EV%nu_ix(nu_i)=ne
              if (EV%MassiveNuA
                  neq = neq+4
              else
                  neq = neq+ EV
              endif
              maxeq = maxeq + n
          end do
      else
          EV%has_nu_relativisti
      end if

      EV%ScalEqsToPropagate = n
      if (present(max_num_eqns)
          max_num_eqns=maxeq
      end if
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 577

```
              if (EV%high_ktau_
                  EV%lmaxnu_tau
                  if (CP%Transf
              else
                  EV%lmaxnu_tau
                  !!!Feb13tweak
                  if (EV%nu_non
              end if
              if (nu_masses(nu_
              EV%lmaxnu_tau(nu_

              EV%nu_ix(nu_i)=ne
              if (EV%MassiveNuA
                  neq = neq+4
              else
                  neq = neq+ EV
              endif
              maxeq = maxeq + n
          end do
      else
          EV%has_nu_relativisti
      end if

      EV%ScalEqsToPropagate = n
      if (present(max_num_eqns)
          max_num_eqns=maxeq
      end if
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 630

```

    end subroutine SetupScala

    subroutine CopyScalarVari
    type(EvolutionVars) EV, E
    real(dl), intent(in) :: y
    real(dl), intent(out) ::
    integer lmax,i, nq
    integer nnueq,nu_i, ix_of
    real(dl) q, pert_scale

    yout=0
    yout(1:basic_num_eqns) =

!#SimDataReplace
    if (.not. is_cosmological
!    if (w_lam /= -1 .and. w_
        yout(EVout%w_ix)=y(EV
        yout(EVout%w_ix+1)=y(
    end if
!#SimDataReplace

    if (.not. EV%no_phot_mult
        if (EV%TightCoupling
            lmax=1
        else
            lmax = min(EV%lma
        end if
        yout(EVout%g_ix:EVout
        if (.not. EV%TightCou
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 604

```

    end subroutine SetupScala

    subroutine CopyScalarVari
    type(EvolutionVars) EV, E
    real(dl), intent(in) :: y
    real(dl), intent(out) ::
    integer lmax,i, nq
    integer nnueq,nu_i, ix_of
    real(dl) q, pert_scale

    yout=0
    yout(1:basic_num_eqns) =
    if (w_lam /= -1 .and. w_P
        yout(EVout%w_ix)=y(EV
        yout(EVout%w_ix+1)=y(
    end if

    if (.not. EV%no_phot_mult
        if (EV%TightCoupling
            lmax=1
        else
            lmax = min(EV%lma
        end if
        yout(EVout%g_ix:EVout
        if (.not. EV%TightCou
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 660

```
          lmax = min(EV%lma
          yout(EVout%polind
        end if
      end if

      if (.not. EV%no_nu_multpo
        if (EV%high_ktau_neut
          lmax=2
        else
          lmax = min(EV%lma
        end if
        yout(EVout%r_ix:EVout
      end if

      if (CP%Num_Nu_massive /=
        do nu_i=1,CP%Nu_mass_
          ix_off=EV%nu_ix(n
          ix_off2=EVOut%nu_
          if (EV%MassiveNuA
            nnueq=4
            yout(ix_off2:
          else if (.not. EV
            lmax=min(EV%l
            nq = min(EV%n
            do i=1,nq
              ind= ix_o
              ind2=ix_o
              yout(ind2
            end do
            do i=nq+1, EV
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 630

```
          lmax = min(EV%lma
          yout(EVout%polind
        end if
      end if

      if (.not. EV%no_nu_multpo
        if (EV%high_ktau_neut
          lmax=2
        else
          lmax = min(EV%lma
        end if
        yout(EVout%r_ix:EVout
      end if

      if (CP%Num_Nu_massive /=
        do nu_i=1,CP%Nu_mass_
          ix_off=EV%nu_ix(n
          ix_off2=EVOut%nu_
          if (EV%MassiveNuA
            nnueq=4
            yout(ix_off2:
          else if (.not. EV
            lmax=min(EV%l
            nq = min(EV%n
            do i=1,nq
              ind= ix_o
              ind2=ix_o
              yout(ind2
            end do
            do i=nq+1, EV
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 690

```
        lmax = mi
        ind2=ix_o
        yout(ind2

        !Add lead
        q=nu_q(i)
        pert_scal
        lmax = mi
        yout(ind2
        + y(EV%nu
      end do
    end if
  end do

  if (EVOut%has_nu_rela
    lmax = min(EVOut%
    yout(EVout%nu_per
  end if
end if

end subroutine CopyScalar


subroutine SetupTensorArr
type(EvolutionVars) EV
integer nu_i, neq
integer, optional, intent
neq=3
EV%g_ix = neq-1 !EV%g_ix+
if (.not. EV%TensTightCou
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 660

```
        lmax = mi
        ind2=ix_o
        yout(ind2

        !Add lead
        q=nu_q(i)
        pert_scal
        lmax = mi
        yout(ind2
          + y(E
      end do
    end if
  end do

  if (EVOut%has_nu_rela
    lmax = min(EVOut%
    yout(EVout%nu_per
  end if
end if

end subroutine CopyScalar


subroutine SetupTensorArr
type(EvolutionVars) EV
integer nu_i, neq
integer, optional, intent
neq=3
EV%g_ix = neq-1 !EV%g_ix+
if (.not. EV%TensTightCou
```

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

```
      EV%E_ix = EV%g_ix + (
      EV%B_ix = EV%E_ix + (
      neq = neq+ (EV%lmaxt-
  end if
  if (present(maxeq)) then
      maxeq =3 + (EV%lmaxt-
  end if
  EV%r_ix = neq -1
  if (DoTensorNeutrinos) th
      neq = neq + EV%lmaxnr
      if (present(maxeq)) m
      if (CP%Num_Nu_massive
          do nu_i=1, CP%nu_
              EV%EvolveTens
              if (EV%Evolve
                  EV%nu_ix(
                  neq = neq
                  if (prese
              end if
          end do
      end if
  end if

  EV%TensEqsToPropagate = n

  end  subroutine SetupTens

  subroutine CopyTensorVari
  type(EvolutionVars) EV, E
  real(dl), intent(in) :: y
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 690

```
      EV%E_ix = EV%g_ix + (
      EV%B_ix = EV%E_ix + (
      neq = neq+ (EV%lmaxt-
  end if
  if (present(maxeq)) then
      maxeq =3 + (EV%lmaxt-
  end if
  EV%r_ix = neq -1
  if (DoTensorNeutrinos) th
      neq = neq + EV%lmaxnr
      if (present(maxeq)) m
      if (CP%Num_Nu_massive
          do nu_i=1, CP%nu_
              EV%EvolveTens
              if (EV%Evolve
                  EV%nu_ix(
                  neq = neq
                  if (prese
              end if
          end do
      end if
  end if

  EV%TensEqsToPropagate = n

  end  subroutine SetupTens

  subroutine CopyTensorVari
  type(EvolutionVars) EV, E
  real(dl), intent(in) :: y
```

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

```
0750        real(dl), intent(out) ::
0751        integer lmaxpolt, lmaxt,
0752
0753        yout=0
0754        yout(1:3) = y(1:3)
0755        if (.not. EVOut%TensTight
0756            lmaxt = min(EVOut%lma
0757            yout(EVout%g_ix+2:EVo
0758            lmaxpolt = min(EV%lma
0759            yout(EVout%E_ix+2:EVo
0760            yout(EVout%B_ix+2:EVo
0761        end if
0762        if (DoTensorNeutrinos) th
0763            lmaxt=min(EV%lmaxnrt,
0764            yout(EVout%r_ix+2:EVo
0765            do nu_i =1, CP%nu_mas
0766                if (EV%EvolveTens
0767                    lmaxt=min(EV%
0768                    do i=1,nqmax
0769                        ind= EV%n
0770                        ind2=EVOu
0771                        yout(ind2
0772                    end do
0773                end if
0774            end do
0775        end if
0776
0777        end subroutine CopyTensor
0778
0779        subroutine GetNumEqns(EV)
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 720

```
0720        real(dl), intent(out) ::
0721        integer lmaxpolt, lmaxt,
0722
0723        yout=0
0724        yout(1:3) = y(1:3)
0725        if (.not. EVOut%TensTight
0726            lmaxt = min(EVOut%lma
0727            yout(EVout%g_ix+2:EVo
0728            lmaxpolt = min(EV%lma
0729            yout(EVout%E_ix+2:EVo
0730            yout(EVout%B_ix+2:EVo
0731        end if
0732        if (DoTensorNeutrinos) th
0733            lmaxt=min(EV%lmaxnrt,
0734            yout(EVout%r_ix+2:EVo
0735            do nu_i =1, CP%nu_mas
0736                if (EV%EvolveTens
0737                    lmaxt=min(EV%
0738                    do i=1,nqmax
0739                        ind= EV%n
0740                        ind2=EVOu
0741                        yout(ind2
0742                    end do
0743                end if
0744            end do
0745        end if
0746
0747        end subroutine CopyTensor
0748
0749        subroutine GetNumEqns(EV)
```

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

```
        use MassiveNu
        !Set the numer of equatio
        type(EvolutionVars) EV
        real(dl) scal, max_nu_mas
        integer nu_i,q_rel,j

        if (CP%Num_Nu_massive ==
            EV%lmaxnu=0
            max_nu_mass=0
        else
            max_nu_mass = maxval(
            do nu_i = 1, CP%Nu_ma
                !Start with momen
                q_rel=0
                do j=1, nqmax
                    !two differen
                    if (nu_q(j) >
                    q_rel = q_rel
                end do

                if (q_rel>= nqmax
                    EV%nq(nu_i)=n
                else
                    EV%nq(nu_i)=q
                end if
                !q_rel = nint(nu_
                !EV%nq(nu_i)=max(
                EV%nu_nonrelativi
            end do

```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 750

```
        use MassiveNu
        !Set the numer of equatio
        type(EvolutionVars) EV
        real(dl) scal, max_nu_mas
        integer nu_i,q_rel,j

        if (CP%Num_Nu_massive ==
            EV%lmaxnu=0
            max_nu_mass=0
        else
            max_nu_mass = maxval(
            do nu_i = 1, CP%Nu_ma
                !Start with momen
                q_rel=0
                do j=1, nqmax
                    !two differen
                    if (nu_q(j) >
                    q_rel = q_rel
                end do

                if (q_rel>= nqmax
                    EV%nq(nu_i)=n
                else
                    EV%nq(nu_i)=q
                end if
                !q_rel = nint(nu_
                !EV%nq(nu_i)=max(
                EV%nu_nonrelativi
            end do

```

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

```
0810            EV%NuMethod = CP%Mass
0811            if (EV%NuMethod == Nu
0812            !l_max for massive ne
0813            if (CP%Transfer%high_
0814                EV%lmaxnu=nint(25
0815            else
0816                EV%lmaxnu=max(3,n
0817                if (max_nu_mass>7
0818            endif
0819        end if
0820
0821        if (CP%closed) then
0822            EV%FirstZerolForBeta
0823        else
0824            EV%FirstZerolForBeta=
0825        end if
0826
0827        EV%high_ktau_neutrino_app
0828        if (CP%WantScalars) then
0829            EV%TightCoupling=.tru
0830            EV%no_phot_multpoles
0831            EV%no_nu_multpoles =.
0832            EV%MassiveNuApprox=.f
0833
0834            if (HighAccuracyDefau
0835                EV%lmaxg  = max(n
0836            else
0837                EV%lmaxg  = max(n
0838            end if
0839            EV%lmaxnr = max(nint(
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 780

```
0780            EV%NuMethod = CP%Mass
0781            if (EV%NuMethod == Nu
0782            !l_max for massive ne
0783            if (CP%Transfer%high_
0784                EV%lmaxnu=nint(25
0785            else
0786                EV%lmaxnu=max(3,n
0787                if (max_nu_mass>7
0788            endif
0789        end if
0790
0791        if (CP%closed) then
0792            EV%FirstZerolForBeta
0793        else
0794            EV%FirstZerolForBeta=
0795        end if
0796
0797        EV%high_ktau_neutrino_app
0798        if (CP%WantScalars) then
0799            EV%TightCoupling=.tru
0800            EV%no_phot_multpoles
0801            EV%no_nu_multpoles =.
0802            EV%MassiveNuApprox=.f
0803
0804            if (HighAccuracyDefau
0805                EV%lmaxg  = max(n
0806            else
0807                EV%lmaxg  = max(n
0808            end if
0809            EV%lmaxnr = max(nint(
          if (max_nu_mass>700 .

          EV%lmaxgpol = EV%lmax
          if (.not.CP%AccurateP

          if (EV%q < 0.05) then
              !Large scales nee
              scal   = 1
              if (CP%AccuratePo
              EV%lmaxgpol=max(3
              EV%lmaxnr=max(3,n
              EV%lmaxg=max(3,ni
              if (CP%AccurateRe
                  EV%lmaxg=EV%l
                  EV%lmaxgpol=E
              end if
          end if

          if (EV%TransferOnly)
              EV%lmaxgpol = min
              EV%lmaxg = min(EV
          end if
          if (CP%Transfer%high_
              if (HighAccuracyD
                  EV%lmaxnr=max
              else
                  EV%lmaxnr=max
              endif
              if (EV%q > 0.04 .
                  EV%lmaxg=max(
```

```
          if (max_nu_mass>700 .

          EV%lmaxgpol = EV%lmax
          if (.not.CP%AccurateP

          if (EV%q < 0.05) then
              !Large scales nee
              scal   = 1
              if (CP%AccuratePo
              EV%lmaxgpol=max(3
              EV%lmaxnr=max(3,n
              EV%lmaxg=max(3,ni
              if (CP%AccurateRe
                  EV%lmaxg=EV%l
                  EV%lmaxgpol=E
              end if
          end if

          if (EV%TransferOnly)
              EV%lmaxgpol = min
              EV%lmaxg = min(EV
          end if
          if (CP%Transfer%high_
              if (HighAccuracyD
                  EV%lmaxnr=max
              else
                  EV%lmaxnr=max
              endif
              if (EV%q > 0.04 .
                  EV%lmaxg=max(
```

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

```
                end if
            end if

            if (CP%closed) then
                EV%lmaxnu=min(EV%
                EV%lmaxnr=min(EV%
                EV%lmaxg=min(EV%l
                EV%lmaxgpol=min(E
            end if

            EV%poltruncfac=real(E
            EV%MaxlNeeded=max(EV%
            if (EV%MaxlNeeded > m
            call SetupScalarArray
            if (CP%closed) EV%nva
            EV%lmaxt=0
        else
            EV%nvar=0
        end if

        if (CP%WantTensors) then
            EV%TensTightCoupling
            EV%lmaxt=max(3,nint(8
            EV%lmaxpolt = max(3,n
            ! if (EV%q < 1e-3) EV
            if (DoTensorNeutrinos
                EV%lmaxnrt=nint(6
                EV%lmaxnut=EV%lma
            else
                EV%lmaxnut=0
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 840

```
                end if
            end if

            if (CP%closed) then
                EV%lmaxnu=min(EV%
                EV%lmaxnr=min(EV%
                EV%lmaxg=min(EV%l
                EV%lmaxgpol=min(E
            end if

            EV%poltruncfac=real(E
            EV%MaxlNeeded=max(EV%
            if (EV%MaxlNeeded > m
            call SetupScalarArray
            if (CP%closed) EV%nva
            EV%lmaxt=0
        else
            EV%nvar=0
        end if

        if (CP%WantTensors) then
            EV%TensTightCoupling
            EV%lmaxt=max(3,nint(8
            EV%lmaxpolt = max(3,n
            ! if (EV%q < 1e-3) EV
            if (DoTensorNeutrinos
                EV%lmaxnrt=nint(6
                EV%lmaxnut=EV%lma
            else
                EV%lmaxnut=0
```

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

```
                EV%lmaxnrt=0
            end if
            if (CP%closed) then
                EV%lmaxt=min(EV%F
                EV%lmaxpolt=min(E
                EV%lmaxnrt=min(EV
                EV%lmaxnut=min(EV
            end if
            EV%MaxlNeededt=max(EV
            if (EV%MaxlNeededt >
            call SetupTensorArray
        else
            EV%nvart=0
        end if


        if (CP%WantVectors) then
            EV%lmaxv=max(10,nint(
            EV%lmaxpolv = max(5,n

            EV%nvarv=(EV%lmaxv)+(

            EV%lmaxnrv=nint(30*lA

            EV%nvarv=EV%nvarv+EV%
            if (CP%Num_Nu_massive
                stop 'massive neu
            end if
        else
            EV%nvarv=0
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 870

```
                EV%lmaxnrt=0
            end if
            if (CP%closed) then
                EV%lmaxt=min(EV%F
                EV%lmaxpolt=min(E
                EV%lmaxnrt=min(EV
                EV%lmaxnut=min(EV
            end if
            EV%MaxlNeededt=max(EV
            if (EV%MaxlNeededt >
            call SetupTensorArray
        else
            EV%nvart=0
        end if


        if (CP%WantVectors) then
            EV%lmaxv=max(10,nint(
            EV%lmaxpolv = max(5,n

            EV%nvarv=(EV%lmaxv)+(

            EV%lmaxnrv=nint(30*lA

            EV%nvarv=EV%nvarv+EV%
            if (CP%Num_Nu_massive
                call MpiStop('mas
            end if
        else
            EV%nvarv=0
```

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

```
end if

end subroutine GetNumEqns

!cccccccccccccccccccccc
subroutine SwitchToMassiv
!When the neutrinos are n
!energy-integrated hierar
type(EvolutionVars) EV, E
integer, intent(in) :: nu

real(dl) a,a2,pnu,clxnu,d
real(dl) qnu
real(dl) y(EV%nvar), yout

a=y(1)
a2=a*a
EVout=EV
EVout%MassiveNuApprox(nu_
call SetupScalarArrayIndi
call CopyScalarVariableAr

!Get density and pressure
call Nu_background(a*nu_m

!Integrate over q
call Nu_Integrate_L012(EV
!clxnu_here  = rhonu*clxn
dpnu=dpnu/rhonu
qnu=qnu/rhonu
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 900

```
end if

end subroutine GetNumEqns

!cccccccccccccccccccccc
subroutine SwitchToMassiv
!When the neutrinos are n
!energy-integrated hierar
type(EvolutionVars) EV, E
integer, intent(in) :: nu

real(dl) a,a2,pnu,clxnu,d
real(dl) qnu
real(dl) y(EV%nvar), yout

a=y(1)
a2=a*a
EVout=EV
EVout%MassiveNuApprox(nu_
call SetupScalarArrayIndi
call CopyScalarVariableAr

!Get density and pressure
call Nu_background(a*nu_m

!Integrate over q
call Nu_Integrate_L012(EV
!clxnu_here  = rhonu*clxn
dpnu=dpnu/rhonu
qnu=qnu/rhonu
```

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

```
0960           clxnu = clxnu/rhonu
0961           pinu=pinu/rhonu
0962
0963           yout(EVout%nu_ix(nu_i))=c
0964           yout(EVout%nu_ix(nu_i)+1)
0965           yout(EVout%nu_ix(nu_i)+2)
0966           yout(EVout%nu_ix(nu_i)+3)
0967
0968           call Nu_Intvsq(EV,y, a, n
0969           !Analytic solution for hi
0970           EVout%G11(nu_i)=EVout%G11
0971           EVout%G30(nu_i)=EVout%G30
0972
0973           EV=EVout
0974           y=yout
0975
0976           end subroutine SwitchToMa
0977
0978🟩         subroutine MassiveNuVarsO
0979           implicit none
0980           type(EvolutionVars) EV
0981▨          real(dl) :: y(EV%nvar), y
0982🟩         real(dl), optional :: grh
0983           !grho = a^2 kappa rho
0984           !gpres = a^2 kappa p
0985           !dgrho = a^2 kappa \delta
0986           !dgp =  a^2 kappa \delta
0987           !dgq = a^2 kappa q (heat
0988           !dgpi = a^2 kappa pi (ani
0989           !dgpi_diff = a^2 kappa (3
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 930

```
0930           clxnu = clxnu/rhonu
0931           pinu=pinu/rhonu
0932
0933           yout(EVout%nu_ix(nu_i))=c
0934           yout(EVout%nu_ix(nu_i)+1)
0935           yout(EVout%nu_ix(nu_i)+2)
0936           yout(EVout%nu_ix(nu_i)+3)
0937
0938           call Nu_Intvsq(EV,y, a, n
0939           !Analytic solution for hi
0940           EVout%G11(nu_i)=EVout%G11
0941           EVout%G30(nu_i)=EVout%G30
0942
0943           EV=EVout
0944           y=yout
0945
0946           end subroutine SwitchToMa
0947
0948🟦         subroutine MassiveNuVarsO
0949           implicit none
0950           type(EvolutionVars) EV
0951▨          real(dl) :: y(EV%nvar), y
0952🟦         real(dl), optional :: grh
0953           !grho = a^2 kappa rho
0954           !gpres = a^2 kappa p
0955           !dgrho = a^2 kappa \delta
0956           !dgp =  a^2 kappa \delta
0957           !dgq = a^2 kappa q (heat
0958           !dgpi = a^2 kappa pi (ani
0959           !dgpi_diff = a^2 kappa (3
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 990

```

        integer nu_i
        real(dl) pinudot,grhormas
        real(dl) adotoa, grhonu_t
        real(dl) clxnu, qnu, pinu
        real(dl) dtauda

        grhonu=0
        dgrhonu=0
        do nu_i = 1, CP%Nu_mass_e
            grhormass_t=grhormass

            !Get density and pres
            call Nu_background(a*

            if (EV%MassiveNuAppro
                clxnu=y(EV%nu_ix(
                !dpnu = y(EV%iq0+
                qnu=y(EV%nu_ix(nu
                pinu=y(EV%nu_ix(n
                pinudot=yprime(EV
            else
                !Integrate over q
                call Nu_Integrate
                !clxnu_here  = rh
                !dpnu=dpnu/rhonu
                qnu=qnu/rhonu
                clxnu = clxnu/rho
                pinu=pinu/rhonu
                adotoa = 1/(a*dta
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 960

```

        integer nu_i
        real(dl) pinudot,grhormas
        real(dl) adotoa, grhonu_t
        real(dl) clxnu, qnu, pinu
        real(dl) dtauda

        grhonu=0
        dgrhonu=0
        do nu_i = 1, CP%Nu_mass_e
            grhormass_t=grhormass

            !Get density and pres
            call Nu_background(a*

            if (EV%MassiveNuAppro
                clxnu=y(EV%nu_ix(
                !dpnu = y(EV%iq0+
                qnu=y(EV%nu_ix(nu
                pinu=y(EV%nu_ix(n
                pinudot=yprime(EV
            else
                !Integrate over q
                call Nu_Integrate
                !clxnu_here  = rh
                !dpnu=dpnu/rhonu
                qnu=qnu/rhonu
                clxnu = clxnu/rho
                pinu=pinu/rhonu
                adotoa = 1/(a*dta
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1020

```
              rhonudot = Nu_drh

              call Nu_pinudot(E
              pinudot=pinudot/r
          endif

          grhonu_t=grhormass_t*
          gpnu_t=grhormass_t*pn

          grhonu = grhonu  + gr
          if (present(gpres)) g

          dgrhonu= dgrhonu + gr
          if (present(dgq)) dgq
          if (present(dgpi)) dg
          if (present(gdpi_diff
          if (present(pidot_sum
      end do
      if (present(grho)) grho =
      if (present(dgrho)) dgrho
      if (present(clxnu_all)) c

      end subroutine MassiveNuV

      subroutine Nu_Integrate_L
      type(EvolutionVars) EV
      !  Compute the perturbati
      !  of one eigenstate of m
      !  density of one eigenst
      !  momentum.
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 990

```
              rhonudot = Nu_drh

              call Nu_pinudot(E
              pinudot=pinudot/r
          endif

          grhonu_t=grhormass_t*
          gpnu_t=grhormass_t*pn

          grhonu = grhonu  + gr
          if (present(gpres)) g

          dgrhonu= dgrhonu + gr
          if (present(dgq)) dgq
          if (present(dgpi)) dg
          if (present(dgpi_diff
          if (present(pidot_sum
      end do
      if (present(grho)) grho =
      if (present(dgrho)) dgrho
      if (present(clxnu_all)) c

      end subroutine MassiveNuV

      subroutine Nu_Integrate_L
      type(EvolutionVars) EV
      !  Compute the perturbati
      !  of one eigenstate of m
      !  density of one eigenst
      !  momentum.
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1050

```
        integer, intent(in) :: nu
        real(dl), intent(in) :: a
        real(dl), intent(OUT) ::
        real(dl), optional, inten
        real(dl) tmp, am, aq,v, p
        integer iq, ind

        !  q is the comoving mome

        drhonu=0
        fnu=0
        if (present(dpnu)) then
            dpnu=0
            pinu=0
        end if
        am=a*nu_masses(nu_i)
        ind=EV%nu_ix(nu_i)
        do iq=1,EV%nq(nu_i)
            aq=am/nu_q(iq)
            v=1._dl/sqrt(1._dl+aq
            drhonu=drhonu+ nu_int
            fnu=fnu+nu_int_kernel
            if (present(dpnu)) th
                dpnu=dpnu+  nu_in
                pinu=pinu+ nu_int
            end if
            ind=ind+EV%lmaxnu_tau
        end do
        ind = EV%nu_pert_ix
        do iq=EV%nq(nu_i)+1,nqmax
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1020

```
        integer, intent(in) :: nu
        real(dl), intent(in) :: a
        real(dl), intent(OUT) ::
        real(dl), optional, inten
        real(dl) tmp, am, aq,v, p
        integer iq, ind

        !  q is the comoving mome

        drhonu=0
        fnu=0
        if (present(dpnu)) then
            dpnu=0
            pinu=0
        end if
        am=a*nu_masses(nu_i)
        ind=EV%nu_ix(nu_i)
        do iq=1,EV%nq(nu_i)
            aq=am/nu_q(iq)
            v=1._dl/sqrt(1._dl+aq
            drhonu=drhonu+ nu_int
            fnu=fnu+nu_int_kernel
            if (present(dpnu)) th
                dpnu=dpnu+  nu_in
                pinu=pinu+ nu_int
            end if
            ind=ind+EV%lmaxnu_tau
        end do
        ind = EV%nu_pert_ix
        do iq=EV%nq(nu_i)+1,nqmax
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1080

```
            !Get the rest from pe
            aq=am/nu_q(iq)
            v=1._dl/sqrt(1._dl+aq
            pert_scale=(nu_masses
            tmp = nu_int_kernel(i
            drhonu=drhonu+ tmp/v
            fnu=fnu+nu_int_kernel
            if (present(dpnu)) th
                dpnu=dpnu+ tmp*v
                pinu = pinu+ nu_i
            end if
        end do

        if (present(dpnu)) then
            dpnu = dpnu/3
        end if

        end subroutine Nu_Integra

        subroutine Nu_pinudot(EV,
        type(EvolutionVars) EV
        integer, intent(in) :: nu
        real(dl), intent(in) :: a

        !  Compute the time deriv
        !  and the shear perturba
        real(dl) pinudot
        real(dl) aq,q,v,aqdot,vdo
        real(dl) psi2,psi2dot
        real(dl) am, pert_scale
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1050

```
            !Get the rest from pe
            aq=am/nu_q(iq)
            v=1._dl/sqrt(1._dl+aq
            pert_scale=(nu_masses
            tmp = nu_int_kernel(i
            drhonu=drhonu+ tmp/v
            fnu=fnu+nu_int_kernel
            if (present(dpnu)) th
                dpnu=dpnu+ tmp*v
                pinu = pinu+ nu_i
            end if
        end do

        if (present(dpnu)) then
            dpnu = dpnu/3
        end if

        end subroutine Nu_Integra

        subroutine Nu_pinudot(EV,
        type(EvolutionVars) EV
        integer, intent(in) :: nu
        real(dl), intent(in) :: a

        !  Compute the time deriv
        !  and the shear perturba
        real(dl) pinudot
        real(dl) aq,q,v,aqdot,vdo
        real(dl) psi2,psi2dot
        real(dl) am, pert_scale
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1110

```
      integer iq,ind

      !  q is the comoving mome
      pinudot=0._dl
      ind=EV%nu_ix(nu_i)+2
      am=a*nu_masses(nu_i)
      do iq=1,EV%nq(nu_i)
          q=nu_q(iq)
          aq=am/q
          aqdot=aq*adotoa
          v=1._dl/sqrt(1._dl+aq
          vdot=-aq*aqdot/(1._dl
          pinudot=pinudot+nu_in
          ind=ind+EV%lmaxnu_tau
      end do
      ind = EV%nu_pert_ix+2
      do iq=EV%nq(nu_i)+1,nqmax
          q=nu_q(iq)
          aq=am/q
          aqdot=aq*adotoa
          pert_scale=(nu_masses
          v=1._dl/sqrt(1._dl+aq
          vdot=-aq*aqdot/(1._dl
          psi2dot=ydot(EV%r_ix+
          psi2=y(EV%r_ix+2)  +
          pinudot=pinudot+nu_in
      end do

      end subroutine Nu_pinudot

```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1080

```
      integer iq,ind

      !  q is the comoving mome
      pinudot=0._dl
      ind=EV%nu_ix(nu_i)+2
      am=a*nu_masses(nu_i)
      do iq=1,EV%nq(nu_i)
          q=nu_q(iq)
          aq=am/q
          aqdot=aq*adotoa
          v=1._dl/sqrt(1._dl+aq
          vdot=-aq*aqdot/(1._dl
          pinudot=pinudot+nu_in
          ind=ind+EV%lmaxnu_tau
      end do
      ind = EV%nu_pert_ix+2
      do iq=EV%nq(nu_i)+1,nqmax
          q=nu_q(iq)
          aq=am/q
          aqdot=aq*adotoa
          pert_scale=(nu_masses
          v=1._dl/sqrt(1._dl+aq
          vdot=-aq*aqdot/(1._dl
          psi2dot=ydot(EV%r_ix+
          psi2=y(EV%r_ix+2)  +
          pinudot=pinudot+nu_in
      end do

      end subroutine Nu_pinudot

```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1140

```
      !cccccccccccccccccccccccc
      function Nu_pi(EV, y, a,
      type(EvolutionVars) EV
      integer, intent(in) :: nu
      real(dl), intent(in) :: a
      real(dl) :: am
      real(dl) pinu,q,aq,v
      integer iq, ind

      if (EV%nq(nu_i)/=nqmax) s
      pinu=0
      ind=EV%nu_ix(nu_i)+2
      am=a*nu_masses(nu_i)
      do iq=1, EV%nq(nu_i)
          q=nu_q(iq)
          aq=am/q
          v=1._dl/sqrt(1._dl+aq
          pinu=pinu+nu_int_kern
          ind =ind+EV%lmaxnut+1
      end do

      end function Nu_pi

      !cccccccccccccccccccccccc
      subroutine Nu_Intvsq(EV,y
      type(EvolutionVars) EV
      integer, intent(in) :: nu
      real(dl), intent(in) :: a
      real(dl), intent(OUT) ::

```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1110

```
      !cccccccccccccccccccccccc
      function Nu_pi(EV, y, a,
      type(EvolutionVars) EV
      integer, intent(in) :: nu
      real(dl), intent(in) :: a
      real(dl) :: am
      real(dl) pinu,q,aq,v
      integer iq, ind

      if (EV%nq(nu_i)/=nqmax) c
      pinu=0
      ind=EV%nu_ix(nu_i)+2
      am=a*nu_masses(nu_i)
      do iq=1, EV%nq(nu_i)
          q=nu_q(iq)
          aq=am/q
          v=1._dl/sqrt(1._dl+aq
          pinu=pinu+nu_int_kern
          ind =ind+EV%lmaxnut+1
      end do

      end function Nu_pi

      !cccccccccccccccccccccccc
      subroutine Nu_Intvsq(EV,y
      type(EvolutionVars) EV
      integer, intent(in) :: nu
      real(dl), intent(in) :: a
      real(dl), intent(OUT) ::

```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1170

```
!  Compute the third orde
!by integrating over mome
real(dl) aq,q,v, am
integer iq, ind

!  q is the comoving mome
am=a*nu_masses(nu_i)
ind=EV%nu_ix(nu_i)
G11=0._dl
G30=0._dl
if (EV%nq(nu_i)/=nqmax) s
do iq=1, EV%nq(nu_i)
    q=nu_q(iq)
    aq=am/q
    v=1._dl/sqrt(1._dl+aq
    G11=G11+nu_int_kernel
    if (EV%lmaxnu_tau(nu_
        G30=G30+nu_int_ke
    end if
    ind = ind+EV%lmaxnu_t
end do

end subroutine Nu_Intvsq


subroutine MassiveNuVars(
implicit none
type(EvolutionVars) EV
real(dl) :: y(EV%nvar), a
real(dl), intent(out), op
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1140

```
!  Compute the third orde
!by integrating over mome
real(dl) aq,q,v, am
integer iq, ind

!  q is the comoving mome
am=a*nu_masses(nu_i)
ind=EV%nu_ix(nu_i)
G11=0._dl
G30=0._dl
if (EV%nq(nu_i)/=nqmax) c
do iq=1, EV%nq(nu_i)
    q=nu_q(iq)
    aq=am/q
    v=1._dl/sqrt(1._dl+aq
    G11=G11+nu_int_kernel
    if (EV%lmaxnu_tau(nu_
        G30=G30+nu_int_ke
    end if
    ind = ind+EV%lmaxnu_t
end do

end subroutine Nu_Intvsq


subroutine MassiveNuVars(
implicit none
type(EvolutionVars) EV
real(dl) :: y(EV%nvar), a
real(dl), intent(out), op
```

```
        !grho = a^2 kappa rho
        !gpres = a^2 kappa p
        !dgrho = a^2 kappa \delta
        !dgp =  a^2 kappa \delta
        !dgq = a^2 kappa q (heat
        integer nu_i
        real(dl) grhormass_t, rho

        do nu_i = 1, CP%Nu_mass_e
            grhormass_t=grhormass

            !Get density and pres
            call Nu_background(a*

            if (EV%MassiveNuAppro
                clxnu=y(EV%nu_ix(
                qnu=y(EV%nu_ix(nu
            else
                !Integrate over q
                call Nu_Integrate
                !clxnu_here  = rh
                qnu=qnu/rhonu
                clxnu = clxnu/rho
            endif

            grhonu_t=grhormass_t*
            gpnu_t=grhormass_t*pn

            grho = grho  + grhonu
            gpres= gpres + gpnu_t
```

```
        !grho = a^2 kappa rho
        !gpres = a^2 kappa p
        !dgrho = a^2 kappa \delta
        !dgp =  a^2 kappa \delta
        !dgq = a^2 kappa q (heat
        integer nu_i
        real(dl) grhormass_t, rho

        do nu_i = 1, CP%Nu_mass_e
            grhormass_t=grhormass

            !Get density and pres
            call Nu_background(a*

            if (EV%MassiveNuAppro
                clxnu=y(EV%nu_ix(
                qnu=y(EV%nu_ix(nu
            else
                !Integrate over q
                call Nu_Integrate
                !clxnu_here  = rh
                qnu=qnu/rhonu
                clxnu = clxnu/rho
            endif

            grhonu_t=grhormass_t*
            gpnu_t=grhormass_t*pn

            grho = grho  + grhonu
            gpres= gpres + gpnu_t
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1230

```
1230              dgrho= dgrho + grhonu
1231              dgq   = dgq    + grhonu
1232
1233              if (present(wnu_arr))
1234                  wnu_arr(nu_i) =pn
1235              end if
1236          end do
1237
1238          end subroutine MassiveNuV
1239
1240          !cccccccccccccccccccccccc
1241          subroutine output(EV,y, j
1242          use ThermoData
1243          use lvalues
1244          use ModelData
1245          implicit none
1246          integer j
1247          type(EvolutionVars) EV
1248          real(dl), target :: y(EV%
1249          real(dl), dimension(:),po
1250
1251          real(dl) dgq,grhob_t,grho
1252          real(dl) qgdot,pigdot,pir
1253          real(dl) a,a2,dz,z,clxc,c
1254
1255          real(dl) tau,x,divfac
1256          real(dl) dgpi_diff, pidot
1257          real(dl), target :: pol(3
1258          !dgpi_diff = sum (3*p_nu
1259
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1200

```
1200              dgrho= dgrho + grhonu
1201              dgq   = dgq    + grhonu
1202
1203              if (present(wnu_arr))
1204                  wnu_arr(nu_i) =pn
1205              end if
1206          end do
1207
1208          end subroutine MassiveNuV
1209
1210          !cccccccccccccccccccccccc
1211          subroutine output(EV,y, t
1212          use ThermoData
1213          type(EvolutionVars) EV
1214          real(dl), target :: y(EV%
1215          real(dl) tau
1216          real(dl), target :: sourc
1217          integer, intent(in) :: nu
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1260

```
    real(dl) k,k2  ,adotoa, g
    real(dl) clxq, vq, diff_r
    real(dl) sources(CTransSc
    real(dl) ISW

    yprime = 0
    call derivs(EV,EV%ScalEqs

    if (EV%TightCoupling .or.
        pol=0
        polprime=0
        ypolprime => polprime
        ypol => pol
    else
        ypolprime => yprime(E
        ypol => y(EV%polind+1
    end if

    k=EV%k_buf
    k2=EV%k2_buf

    a   =y(1)
    a2  =a*a
    etak=y(2)
    clxc=y(3)
    clxb=y(4)
    vb  =y(5)
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1218

```

    yprime = 0
    EV%OutputSources => Sourc
    if (num_custom_sources>0)
        EV%CustomSources => s
    call derivs(EV,EV%ScalEqs
    nullify(EV%OutputSources,
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1287

```
    vbdot =yprime(5)

    !  Compute expansion rate

    grhob_t=grhob/a
    grhoc_t=grhoc/a
    grhor_t=grhornomass/a2
    grhog_t=grhog/a2

!#SimDataReplace
!    grhov_t=grhov*a**(-1-3*w
     
     grhov_t=grho_de(a)/a2
!#SimDataReplace

    grho=grhob_t+grhoc_t+grho

!#SimDataReplace
!    gpres=(grhog_t+grhor_t)/
     gpres=(grhog_t+grhor_t)/
!#SimDataReplace

    !  8*pi*a*a*SUM[rho_i*clx
    dgrho=grhob_t*clxb+grhoc_

    !  8*pi*a*a*SUM[(rho_i+p_
    dgq=grhob_t*vb

    dgpi=0
    dgpi_diff = 0
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1225

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1317

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1225

```
        pidot_sum = 0

        if (CP%Num_Nu_Massive /=
            call MassiveNuVarsOut
        end if

!#SimDataReplace
!         if (w_lam /= -1 .and. w_
          if (.not. is_cosmologica
             clxq=y(EV%w_ix)
             vq=y(EV%w_ix+1)
             dgrho=dgrho + clxq*gr
!            dgq = dgq + vq*grhov_
             dgq = dgq + vq*grhov_
         end if
!#SimDataReplace
         adotoa=sqrt((grho+grhok)/

         if (EV%no_nu_multpoles) t
             z=(0.5_dl*dgrho/k + e
             dz= -adotoa*z - 0.5_d
             clxr=-4*dz/k
             qr=-4._dl/3*z
             pir=0
             pirdot=0
         else
             clxr=y(EV%r_ix)
             qr   =y(EV%r_ix+1)
             pir =y(EV%r_ix+2)
             pirdot=yprime(EV%r_ix
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1347

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1225

```
    end if

    if (EV%no_phot_multpoles)
        z=(0.5_dl*dgrho/k + e
        dz= -adotoa*z - 0.5_d
        clxg=-4*dz/k -4/k*opa
        qg=-4._dl/3*z
        pig=0
        pigdot=0
        octg=0
        octgprime=0
        qgdot = -4*dz/3
    else
        if (EV%TightCoupling)
            pig = EV%pig
            !pigdot=EV%pigdot
            if (second_order_
                octg = (3._dl
                ypol(2) = EV%
                ypol(3) = (3.
            else
                ypol(2) = EV%
                octg=0
            end if
            octgprime=0
        else
            pig =y(EV%g_ix+2)
            pigdot=yprime(EV%
            octg=y(EV%g_ix+3)
            octgprime=yprime(
        end if
        clxg=y(EV%g_ix)
        qg  =y(EV%g_ix+1)
        qgdot =yprime(EV%g_ix
    end if

    dgrho = dgrho + grhog_t*c
    dgq   = dgq   + grhog_t*q
    dgpi  = dgpi  + grhor_t*p


    !  Get sigma (shear) and
    !  have to get z from eta
    z=(0.5_dl*dgrho/k + etak)
    sigma=(z+1.5_dl*dgq/k2)/E

    polter = 0.1_dl*pig+9._dl

    if (CP%flat) then
        x=k*(CP%tau0-tau)
        divfac=x*x
    else
        x=(CP%tau0-tau)/CP%r
        divfac=(CP%r*rofChi(x
    end if


    if (EV%TightCoupling) the
        if (second_order_tigh
            pigdot = EV%pigdo
            ypolprime(2)= (pi
        else
            pigdot = -dopac(j
            +etak/EV%Kf(1)-
            ypolprime(2)= pig
        end if
    end if

    pidot_sum =  pidot_sum +
    diff_rhopi = pidot_sum -

    !Maple's fortran output -
    !2phi' term (\phi' + \psi
    ISW = (4.D0/3.D0*k*EV%Kf(
    -diff_rhopi/k**2-1.D0/ado
    -2.D0/k*adotoa/EV%Kf(1)*e

    !e.g. to get only late-ti
    !  if (1/a-1 < 30) ISW=0

    !The rest, note y(9)->oct
    sources(1)= ISW +  ((-9.D
    (11.D0/10.D0*sigma- 3.D0/
    (-180.D0*ypolprime(2)-30.
    (-(9.D0*pigdot+ 54.D0*ypo
    (-21.D0/5.D0*adotoa*sigma
    vbdot+3.D0/40.D0*qgdot- 9
    (-9.D0/160.D0*dopac(j)*pi
    (3.D0/16.D0*ddvis(j)*pig+

    ! Doppler term
    !   sources(1)=  (sigma+v
    !         +1.D0/k/EV%Kf(1

    !Equivalent full result
    !    t4 = 1.D0/adotoa
    !    t92 = k**2
    !   sources(1) = (4.D0/3.
    !       (3.D0/8.D0*ypol(2
    !    sources(1) = sources
    !         3.D0/8.D0*EV%Kf
    !        gpres)*sigma*exp
    !        EV%Kf(1)+(vbdot-
    !        5.D0*sigma*adoto
    !        27.D0/80.D0*ypol
    !        -9.D0/160.D0*dop
    !        160.D0*pig)*opac
    !        8.D0*ddvis(j)*yp


    if (x > 0._dl) then
        !E polarization sourc
        sources(2)=vis(j)*pol
        !factor of four becau
    else
        sources(2)=0
    end if

    if (CTransScal%NumSources
        !Get lensing sources
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1467

```
            !Can modify this here
            if (tau>tau_maxvis .a
                !phi_lens = Phi -
                phi = -(dgrho +3*

                sources(3) = -2*p
                !We include the l
            else
                sources(3) = 0
            end if
        end if

        end subroutine output



        !cccccccccccccccccccccccc
        subroutine outputt(EV,yt,
        !calculate the tensor sou
        use ThermoData

        implicit none
        integer j,n
        type(EvolutionVars) :: EV
        real(dl), target :: yt(n)
        real(dl) tau,dt,dte,dtb,x
        real(dl) pig, pigdot, oct
        real(dl) sinhxr,cothxor
        real(dl) k,k2
        real(dl), dimension(:),po
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1225

```

        end subroutine output




        !cccccccccccccccccccccccc
        subroutine outputt(EV,yt,
        !calculate the tensor sou
        use ThermoData

        implicit none
        integer n
        type(EvolutionVars) :: EV
        real(dl), target :: yt(n)
        real(dl) tau,dt,dte,dtb,x
        real(dl) pig, pigdot, oct
        real(dl) sinhxr,cothxor
        real(dl) k,k2
        real(dl), dimension(:),po
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1496

```
1496        real(dl), target :: pol(3
1497        real(dl) dtauda

1498
1499        call derivst(EV,EV%nvart,
1500
1501        k2=EV%k2_buf
1502        k=EV%k_buf
1503        aux=EV%aux_buf
1504        shear = yt(3)
1505
1506        x=(CP%tau0-tau)/CP%r

1507
1508        !  And the electric part
1509        if (.not. EV%TensTightCou
1510            !  Use the full expre
1511            pig=yt(EV%g_ix+2)
1512            pigdot=ytprime(EV%g_i
1513            E => yt(EV%E_ix+1:)
1514            Eprime=> ytprime(EV%E
1515            Bprime => ytprime(EV%
1516            octg=ytprime(EV%g_ix+
1517        else
1518            !  Use the tight-coup
1519            a =yt(1)
1520            adotoa = 1/(a*dtauda(
1521            pigdot=32._dl/45._dl*
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1244

```
1244        real(dl), target :: pol(3
1245        real(dl) dtauda
1246        real(dl) opacity, dopacit
1247            visibility, dvisibili
1248
1249        call derivst(EV,EV%nvart,
1250
1251        k2=EV%k2_buf
1252        k=EV%k_buf
1253        aux=EV%aux_buf
1254        shear = yt(3)
1255
1256        x=(CP%tau0-tau)/CP%r
1257        call IonizationFunctionsA
1258            visibility, dvisibili
1259
1260        !  And the electric part
1261        if (.not. EV%TensTightCou
1262            !  Use the full expre
1263            pig=yt(EV%g_ix+2)
1264            pigdot=ytprime(EV%g_i
1265            E => yt(EV%E_ix+1:)
1266            Eprime=> ytprime(EV%E
1267            Bprime => ytprime(EV%
1268            octg=ytprime(EV%g_ix+
1269        else
1270            !  Use the tight-coup
1271            a =yt(1)
1272            adotoa = 1/(a*dtauda(
1273            pigdot=32._dl/45._dl*
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1522

```
1522          pig = 32._dl/45._dl*k
1523          pol=0
1524          polEprime=0
1525          polBprime=0
1526          E=>pol
1527          EPrime=>polEPrime
1528          BPrime=>polBPrime
1529          E(2)=pig/4._dl
1530          EPrime(2)=pigdot/4
1531          octg=0
1532      endif
1533
1534      sinhxr=rofChi(x)*CP%r
1535
1536      if (EV%q*sinhxr > 1.e-8_d
1537          prefac=sqrt(EV%q2*CP%
1538          cothxor=cosfunc(x)/si
1539
1540          polter = 0.1_dl*pig +
1541          polterdot=9._dl/15._d
1542          polterddot = 9._dl/15
1543          Eprime(2)-polterdot)
1544          +0.1_dl*(k*(-octg*EV%
1545          dopac(j)*(pig - polte
1546
1547          dt=(shear*expmmu(j) +
1548
1549          dte=CP%r*15._dl/8._dl
1550          ((ddvis(j)*polter + 2
1551          + 4._dl*cothxor*(dvis
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1274

```
1274          pig = 32._dl/45._dl*k
1275          pol=0
1276          polEprime=0
1277          polBprime=0
1278          E=>pol
1279          EPrime=>polEPrime
1280          BPrime=>polBPrime
1281          E(2)=pig/4._dl
1282          EPrime(2)=pigdot/4
1283          octg=0
1284      endif
1285
1286      sinhxr=rofChi(x)*CP%r
1287
1288      if (EV%q*sinhxr > 1.e-8_d
1289          prefac=sqrt(EV%q2*CP%
1290          cothxor=cosfunc(x)/si
1291
1292          polter = 0.1_dl*pig +
1293          polterdot=9._dl/15._d
1294          polterddot = 9._dl/15
1295              Eprime(2)-polterd
1296              +0.1_dl*(k*(-octg
1297              dopacity*(pig - p
1298
1299          dt=(shear*exptau + (1
1300
1301          dte=CP%r*15._dl/8._dl
1302              ((ddvisibility*po
1303              + 4._dl*cothxor*(
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1552

```
            vis(j)*polter*(k2 -6*

            dtb=15._dl/4._dl*EV%q
        else
            dt=0._dl
            dte=0._dl
            dtb=0._dl
        end if

        end subroutine outputt

        !cccccccccccccccccccccc
        subroutine outputv(EV,yv,
        !calculate the vector sou
        use ThermoData

        implicit none
        integer j,n
        type(EvolutionVars) :: EV
        real(dl), target :: yv(n)
        real(dl) tau,dt,dte,dtb,x
        real(dl) vb,qg, pig, polt
        real(dl) k,k2
        real(dl), dimension(:),po

        call derivsv(EV,EV%nvarv,

```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1304

```
                visibility*polter

            dtb=15._dl/4._dl*EV%q
        else
            dt=0._dl
            dte=0._dl
            dtb=0._dl
        end if

        end subroutine outputt

        !cccccccccccccccccccccc
        subroutine outputv(EV,yv,
        !calculate the vector sou
        use ThermoData

        implicit none
        integer n
        type(EvolutionVars) :: EV
        real(dl), target :: yv(n)
        real(dl) tau,dt,dte,dtb,x
        real(dl) vb,qg, pig, polt
        real(dl) k,k2
        real(dl), dimension(:),po
        real(dl) opacity, dopacit
            visibility, dvisibili


        call derivsv(EV,EV%nvarv,

```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1579

```
1579          k2=EV%k2_buf
1580          k=EV%k_buf
1581          sigma = yv(2)
1582          vb   = yv(3)
1583          qg   = yv(4)
1584          pig = yv(5)
1585
1586
1587          x=(CP%tau0-tau)*k
1588
1589          if (x > 1.e-8_dl) then
1590              E => yv(EV%lmaxv+3:)
1591              Eprime=> yvprime(EV%l
1592
1593              polter = 0.1_dl*pig +
1594              polterdot=9._dl/15._d
1595



1596              if (yv(1) < 1e-3) the
1597                  dt = 1
1598              else
1599                  dt =0
1600              end if
1601              dt= (4*(vb+sigma)*vis
1602              + 4*(expmmu(j)*yvprim
1603
1604              dte= 15._dl/2*2*polte
1605
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1334

```
1334          k2=EV%k2_buf
1335          k=EV%k_buf
1336          sigma = yv(2)
1337          vb   = yv(3)
1338          qg   = yv(4)
1339          pig = yv(5)
1340
1341
1342          x=(CP%tau0-tau)*k
1343
1344          if (x > 1.e-8_dl) then
1345              E => yv(EV%lmaxv+3:)
1346              Eprime=> yvprime(EV%l
1347
1348              polter = 0.1_dl*pig +
1349              polterdot=9._dl/15._d
1350
1351              call IonizationFuncti
1352                  visibility, dvisi
1353
1354              if (yv(1) < 1e-3) the
1355                  dt = 1
1356              else
1357                  dt =0
1358              end if
1359              dt= (4*(vb+sigma)*vis
1360                  + 4*(exptau*yvpri
1361
1362              dte= 15._dl/2*2*polte
1363
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1606

```
    dtb= -15._dl/2*polter
else
    dt=0
    dte=0
    dtb=0
end if

end subroutine outputv


!ccccccccccccccccccccccc
subroutine initial(EV,y,
!  Initial conditions.
use ThermoData
implicit none

type(EvolutionVars) EV
real(dl) y(EV%nvar)
real(dl) Rp15,tau,x,x2,x3
Rc,Rb,Rv,Rg,grhonu,chi
real(dl) k,k2
real(dl) a,a2, iqg, rhoma
integer l,i, nu_i, j, ind
integer, parameter :: i_c
i_qg=5,i_qr=6,i_vb=7,i_pi
integer, parameter :: i_m
real(dl) initv(6,1:i_max)

nullify(EV%OutputTransfer
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1364

```
    dtb= -15._dl/2*polter
else
    dt=0
    dte=0
    dtb=0
end if

end subroutine outputv


!ccccccccccccccccccccccc
subroutine initial(EV,y,
!  Initial conditions.
use ThermoData
implicit none

type(EvolutionVars) EV
real(dl) y(EV%nvar)
real(dl) Rp15,tau,x,x2,x3
    Rc,Rb,Rv,Rg,grhonu,ch
real(dl) k,k2
real(dl) a,a2, iqg, rhoma
integer l,i, nu_i, j, ind
integer, parameter :: i_c
    i_qg=5,i_qr=6,i_vb=7,
integer, parameter :: i_m
real(dl) initv(6,1:i_max)

nullify(EV%OutputTransfer
nullify(EV%OutputSources)
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1635

```

1635
1636        if (CP%flat) then
1637            EV%k_buf=EV%q
1638            EV%k2_buf=EV%q2
1639            EV%Kf(1:EV%MaxlNeeded
1640        else
1641            EV%k2_buf=EV%q2-CP%cu
1642            EV%k_buf=sqrt(EV%k2_b
1643
1644            do l=1,EV%MaxlNeeded
1645                EV%Kf(l)=1._dl-CP
1646            end do
1647        end if
1648
1649        k=EV%k_buf
1650        k2=EV%k2_buf
1651
1652        do j=1,EV%MaxlNeeded
1653            EV%denlk(j)=denl(j)*k
1654            EV%denlk2(j)=denl(j)*
1655            EV%polfack(j)=polfac(
1656        end do
1657
1658        !Get time to switch off t
1659        !The numbers here are a b
1660        !The high k increase save
1661        !The lower k ones are mor
1662        !as ensuring tight coupli
1663        if (EV%k_buf > epsw) then
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1394

```
1394          nullify(EV%CustomSources)
1395
1396        if (CP%flat) then
1397            EV%k_buf=EV%q
1398            EV%k2_buf=EV%q2
1399            EV%Kf(1:EV%MaxlNeeded
1400        else
1401            EV%k2_buf=EV%q2-CP%cu
1402            EV%k_buf=sqrt(EV%k2_b
1403
1404            do l=1,EV%MaxlNeeded
1405                EV%Kf(l)=1._dl-CP
1406            end do
1407        end if
1408
1409        k=EV%k_buf
1410        k2=EV%k2_buf
1411
1412        do j=1,EV%MaxlNeeded
1413            EV%denlk(j)=denl(j)*k
1414            EV%denlk2(j)=denl(j)*
1415            EV%polfack(j)=polfac(
1416        end do
1417
1418        !Get time to switch off t
1419        !The numbers here are a b
1420        !The high k increase save
1421        !The lower k ones are mor
1422        !as ensuring tight coupli
1423        if (EV%k_buf > epsw) then
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1664

```
            if (EV%k_buf > epsw*5
                ep=ep0*5/Accuracy
                if (HighAccuracyD
            else
                ep=ep0
            end if
        else
            ep=ep0
        end if
        if (second_order_tightcou
        EV%TightSwitchoffTime = m


        y=0

        !  k*tau, (k*tau)**2, (k*
        x=k*tau
        x2=x*x
        x3=x2*x
        rhomass =  sum(grhormass(
        grhonu=rhomass+grhornomas

        om = (grhob+grhoc)/sqrt(3
        omtau=om*tau
        Rv=grhonu/(grhonu+grhog)

        Rg = 1-Rv
        Rc=CP%omegac/(CP%omegac+C
        Rb=1-Rc
        Rp15=4*Rv+15
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1424

```
            if (EV%k_buf > epsw*5
                ep=ep0*5/Accuracy
                if (HighAccuracyD
            else
                ep=ep0
            end if
        else
            ep=ep0
        end if
        if (second_order_tightcou
        EV%TightSwitchoffTime = m


        y=0

        !  k*tau, (k*tau)**2, (k*
        x=k*tau
        x2=x*x
        x3=x2*x
        rhomass =  sum(grhormass(
        grhonu=rhomass+grhornomas

        om = (grhob+grhoc)/sqrt(3
        omtau=om*tau
        Rv=grhonu/(grhonu+grhog)

        Rg = 1-Rv
        Rc=CP%omegac/(CP%omegac+C
        Rb=1-Rc
        Rp15=4*Rv+15
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1694

```
1694
1695        if (CP%Scalar_initial_con
1696        stop 'Invalid initial con
1697
1698        a=tau*adotrad*(1+omtau/4)
1699        a2=a*a
1700
1701        initv=0
1702
1703        !  Set adiabatic initial
1704
1705        chi=1  !Get transfer func
1706        initv(1,i_clxg)=-chi*EV%K
1707        initv(1,i_clxr)= initv(1,
1708        initv(1,i_clxb)=0.75_dl*i
1709        initv(1,i_clxc)=initv(1,i
1710        initv(1,i_qg)=initv(1,i_c
1711        initv(1,i_qr)=-chi*EV%Kf(
1712        initv(1,i_vb)=0.75_dl*ini
1713        initv(1,i_pir)=chi*4._dl/
1714        initv(1,i_aj3r)=chi*4/21.
1715        initv(1,i_eta)=-chi*2*EV%
1716
1717        if (CP%Scalar_initial_con
1718            !CDM isocurvature
1719
1720            initv(2,i_clxg)= Rc*o
1721            initv(2,i_clxr)=initv
1722            initv(2,i_clxb)=initv
1723            initv(2,i_clxc)=1+ini
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1454

```
1454
1455        if (CP%Scalar_initial_con
1456            call MpiStop('Invalid
1457
1458        a=tau*adotrad*(1+omtau/4)
1459        a2=a*a
1460
1461        initv=0
1462
1463        !  Set adiabatic initial
1464
1465        chi=1  !Get transfer func
1466        initv(1,i_clxg)=-chi*EV%K
1467        initv(1,i_clxr)= initv(1,
1468        initv(1,i_clxb)=0.75_dl*i
1469        initv(1,i_clxc)=initv(1,i
1470        initv(1,i_qg)=initv(1,i_c
1471        initv(1,i_qr)=-chi*EV%Kf(
1472        initv(1,i_vb)=0.75_dl*ini
1473        initv(1,i_pir)=chi*4._dl/
1474        initv(1,i_aj3r)=chi*4/21.
1475        initv(1,i_eta)=-chi*2*EV%
1476
1477        if (CP%Scalar_initial_con
1478            !CDM isocurvature
1479
1480            initv(2,i_clxg)= Rc*o
1481            initv(2,i_clxr)=initv
1482            initv(2,i_clxb)=initv
1483            initv(2,i_clxc)=1+ini
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1724

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1484

```
1724  |              initv(2,i_qg)=-Rc/9*o      1484  |              initv(2,i_qg)=-Rc/9*o
1725  |              initv(2,i_qr)=initv(2      1485  |              initv(2,i_qr)=initv(2
1726  |              initv(2,i_vb)=0.75_dl      1486  |              initv(2,i_vb)=0.75_dl
1727  |              initv(2,i_pir)=-Rc*om      1487  |              initv(2,i_pir)=-Rc*om
1728  |              initv(2,i_eta)= Rc*om      1488  |              initv(2,i_eta)= Rc*om
1729  |              initv(2,i_aj3r)=0          1489  |              initv(2,i_aj3r)=0
1730  |              !Baryon isocurvature       1490  |              !Baryon isocurvature
1731🟩|              if (Rc==0) stop 'Isoc      1491🟦|              if (Rc==0) call MpiSt
1732  |                                         1492  |
1733  |              initv(3,:) = initv(2,      1493  |              initv(3,:) = initv(2,
1734  |              initv(3,i_clxc) = ini      1494  |              initv(3,i_clxc) = ini
1735  |              initv(3,i_clxb)= init      1495  |              initv(3,i_clxb)= init
1736  |                                         1496  |
1737  |              !neutrino isocurvatur      1497  |              !neutrino isocurvatur
1738  |                                         1498  |
1739  |              initv(4,i_clxg)=Rv/Rg      1499  |              initv(4,i_clxg)=Rv/Rg
1740  |              initv(4,i_clxr)=1-x2/      1500  |              initv(4,i_clxr)=1-x2/
1741  |              initv(4,i_clxc)=-omta      1501  |              initv(4,i_clxc)=-omta
1742  |              initv(4,i_clxb)= Rv/R      1502  |              initv(4,i_clxb)= Rv/R
1743  |              iqg = - Rv/Rg*(x/3 -       1503  |              iqg = - Rv/Rg*(x/3 -
1744  |              initv(4,i_qg) =iqg         1504  |              initv(4,i_qg) =iqg
1745  |              initv(4,i_qr) = x/3        1505  |              initv(4,i_qr) = x/3
1746  |              initv(4,i_vb)=0.75_dl      1506  |              initv(4,i_vb)=0.75_dl
1747  |              initv(4,i_pir)=x2/Rp1      1507  |              initv(4,i_pir)=x2/Rp1
1748  |              initv(4,i_eta)=EV%Kf(      1508  |              initv(4,i_eta)=EV%Kf(
1749  |                                         1509  |
1750  |              !neutrino isocurvatur      1510  |              !neutrino isocurvatur
1751  |                                         1511  |
1752  |              initv(5,i_clxg)=Rv/Rg      1512  |              initv(5,i_clxg)=Rv/Rg
1753  |              initv(5,i_clxr)=-x -3      1513  |              initv(5,i_clxr)=-x -3
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1754

```
            initv(5,i_clxc)=-9*om
            initv(5,i_clxb)= 3*Rv
            iqg = Rv/Rg*(-1 + 3*R
            initv(5,i_qg) =iqg
            initv(5,i_qr) = 1 - x
            initv(5,i_vb)=0.75_dl
            initv(5,i_pir)=2*x/(4
            initv(5,i_eta)=2*EV%K
            initv(5,i_aj3r) = 3._

            !quintessence isocurv
        end if

        if (CP%Scalar_initial_con
            InitVec = 0
            do i=1,initial_nummod
                InitVec = InitVec
            end do
        else
            InitVec = initv(CP%Sc
            if (CP%Scalar_initial
            !So we start with chi
        end if

        y(1)=a
        y(2)= -InitVec(i_eta)*k/2
        !get eta_s*k, where eta_s

        !  CDM
        y(3)=InitVec(i_clxc)
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1514

```
            initv(5,i_clxc)=-9*om
            initv(5,i_clxb)= 3*Rv
            iqg = Rv/Rg*(-1 + 3*R
            initv(5,i_qg) =iqg
            initv(5,i_qr) = 1 - x
            initv(5,i_vb)=0.75_dl
            initv(5,i_pir)=2*x/(4
            initv(5,i_eta)=2*EV%K
            initv(5,i_aj3r) = 3._

            !quintessence isocurv
        end if

        if (CP%Scalar_initial_con
            InitVec = 0
            do i=1,initial_nummod
                InitVec = InitVec
            end do
        else
            InitVec = initv(CP%Sc
            if (CP%Scalar_initial
            !So we start with chi
        end if

        y(1)=a
        y(2)= -InitVec(i_eta)*k/2
        !get eta_s*k, where eta_s

        !  CDM
        y(3)=InitVec(i_clxc)
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1784

```

        !  Baryons
        y(4)=InitVec(i_clxb)
        y(5)=InitVec(i_vb)

        !  Photons
        y(EV%g_ix)=InitVec(i_clxg
        y(EV%g_ix+1)=InitVec(i_qg

      !#SimDataReplace
      !     if (w_lam /= -1 .and. w_
         if (.not. is_cosmologica
      !#SimDataReplace
          y(EV%w_ix) = InitVec(
          y(EV%w_ix+1) = InitVe
        end if

        !  Neutrinos
        y(EV%r_ix)=InitVec(i_clxr
        y(EV%r_ix+1)=InitVec(i_qr
        y(EV%r_ix+2)=InitVec(i_pi

        if (EV%lmaxnr>2) then
          y(EV%r_ix+3)=InitVec(
        endif

        if (CP%Num_Nu_massive ==

        do nu_i = 1, CP%Nu_mass_e
          EV%MassiveNuApproxTim
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1544

```

        !  Baryons
        y(4)=InitVec(i_clxb)
        y(5)=InitVec(i_vb)

        !  Photons
        y(EV%g_ix)=InitVec(i_clxg
        y(EV%g_ix+1)=InitVec(i_qg

        if (w_lam /= -1 .and. w_P
          y(EV%w_ix) = InitVec(
          y(EV%w_ix+1) = InitVe
        end if

        !  Neutrinos
        y(EV%r_ix)=InitVec(i_clxr
        y(EV%r_ix+1)=InitVec(i_qr
        y(EV%r_ix+2)=InitVec(i_pi

        if (EV%lmaxnr>2) then
          y(EV%r_ix+3)=InitVec(
        endif

        if (CP%Num_Nu_massive ==

        do nu_i = 1, CP%Nu_mass_e
          EV%MassiveNuApproxTim
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1814

```
            a_massive =  20000*k/
            if (a_massive >=0.99)
                EV%MassiveNuAppro
            else if (a_massive >
                EV%MassiveNuAppro
            end if
            ind = EV%nu_ix(nu_i)
            do  i=1,EV%nq(nu_i)
                y(ind:ind+2)=y(EV
                if (EV%lmaxnu_tau
                ind = ind + EV%lm
            end do
        end do

        end subroutine initial


        !cccccccccccccccccccccc
        subroutine initialt(EV,yt
        !  Initial conditions for
        use ThermoData
        implicit none
        real(dl) bigR,tau,x,aj3r,
        integer l
        type(EvolutionVars) EV
        real(dl) k,k2 ,a, omtau
        real(dl) yt(EV%nvart)
        real(dl) tens0, ep, tensf

        if (CP%flat) then
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1571

```
            a_massive =  20000*k/
            if (a_massive >=0.99)
                EV%MassiveNuAppro
            else if (a_massive >
                EV%MassiveNuAppro
            end if
            ind = EV%nu_ix(nu_i)
            do  i=1,EV%nq(nu_i)
                y(ind:ind+2)=y(EV
                if (EV%lmaxnu_tau
                ind = ind + EV%lm
            end do
        end do

        end subroutine initial


        !cccccccccccccccccccccc
        subroutine initialt(EV,yt
        !  Initial conditions for
        use ThermoData
        implicit none
        real(dl) bigR,tau,x,aj3r,
        integer l
        type(EvolutionVars) EV
        real(dl) k,k2 ,a, omtau
        real(dl) yt(EV%nvart)
        real(dl) tens0, ep, tensf

        if (CP%flat) then
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1844

```
1844              EV%aux_buf=1._dl
1845              EV%k2_buf=EV%q2
1846              EV%k_buf=EV%q
1847              EV%Kft(1:EV%MaxlNeede
1848          else
1849              EV%k2_buf=EV%q2-3*CP%
1850              EV%k_buf=sqrt(EV%k2_b
1851              EV%aux_buf=sqrt(1._dl
1852          endif
1853
1854          k=EV%k_buf
1855          k2=EV%k2_buf
1856
1857          do l=1,EV%MaxlNeededt
1858              if (.not. CP%flat) EV
1859              EV%denlkt(1,l)=k*denl
1860              tensfac=real((l+3)*(l
1861              EV%denlkt(2,l)=k*denl
1862              EV%denlkt(3,l)=k*denl
1863              EV%denlkt(4,l)=k*4._d
1864          end do
1865
1866          if (k > 0.06_dl*epsw) the
1867              ep=ep0
1868          else
1869              ep=0.2_dl*ep0
1870          end if
1871
1872          !    finished_tightcoupli
1873          EV%TightSwitchoffTime = m
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1601

```
1601              EV%aux_buf=1._dl
1602              EV%k2_buf=EV%q2
1603              EV%k_buf=EV%q
1604              EV%Kft(1:EV%MaxlNeede
1605          else
1606              EV%k2_buf=EV%q2-3*CP%
1607              EV%k_buf=sqrt(EV%k2_b
1608              EV%aux_buf=sqrt(1._dl
1609          endif
1610
1611          k=EV%k_buf
1612          k2=EV%k2_buf
1613
1614          do l=1,EV%MaxlNeededt
1615              if (.not. CP%flat) EV
1616              EV%denlkt(1,l)=k*denl
1617              tensfac=real((l+3)*(l
1618              EV%denlkt(2,l)=k*denl
1619              EV%denlkt(3,l)=k*denl
1620              EV%denlkt(4,l)=k*4._d
1621          end do
1622
1623          if (k > 0.06_dl*epsw) the
1624              ep=ep0
1625          else
1626              ep=0.2_dl*ep0
1627          end if
1628
1629          !    finished_tightcoupli
1630          EV%TightSwitchoffTime = m
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1874

```

        a=tau*adotrad
        rhomass =  sum(grhormass(
        omtau = tau*(grhob+grhoc)

        if (DoTensorNeutrinos) th
            bigR = (rhomass+grhor
        else
            bigR = 0._dl
        end if

        x=k*tau

        yt(1)=a
        tens0 = 1

        yt(2)= tens0
        !commented things are for
        !-15/28._dl*x**2*(bigR-1)

        elec=-tens0*(1+2*CP%curv/

        !shear
        yt(3)=-5._dl/2/(bigR+5)*x
        !          + 15._dl/14*x*

        yt(4:EV%nvart)=0._dl

        !  Neutrinos
        if (DoTensorNeutrinos) th
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1631

```

        a=tau*adotrad
        rhomass =  sum(grhormass(
        omtau = tau*(grhob+grhoc)

        if (DoTensorNeutrinos) th
            bigR = (rhomass+grhor
        else
            bigR = 0._dl
        end if

        x=k*tau

        yt(1)=a
        tens0 = 1

        yt(2)= tens0
        !commented things are for
        !-15/28._dl*x**2*(bigR-1)

        elec=-tens0*(1+2*CP%curv/

        !shear
        yt(3)=-5._dl/2/(bigR+5)*x
        !          + 15._dl/14*x*

        yt(4:EV%nvart)=0._dl

        !  Neutrinos
        if (DoTensorNeutrinos) th
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1904

```
          pir=-2._dl/3._dl/(big
          !           + (bigR-1
          aj3r=  -2._dl/21._dl/
          !           + 3._dl/7
          yt(EV%r_ix+2)=pir
          yt(EV%r_ix+3)=aj3r
          !Should set up massiv
      end if

      end subroutine initialt

      !cccccccccccccccccccccc
      subroutine initialv(EV,yv
      !  Initial conditions for

      implicit none
      real(dl) bigR,Rc,tau,x,pi
      type(EvolutionVars) EV
      real(dl) k,k2 ,a, omtau
      real(dl) yv(EV%nvarv)

      if (CP%flat) then
          EV%k2_buf=EV%q2
          EV%k_buf=EV%q
      else
          stop 'Vectors not sup
      endif

      k=EV%k_buf
      k2=EV%k2_buf
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1661

```
          pir=-2._dl/3._dl/(big
          !           + (bigR-1
          aj3r=  -2._dl/21._dl/
              !           + 3._
              yt(EV%r_ix+2)=pir
          yt(EV%r_ix+3)=aj3r
          !Should set up massiv
      end if

      end subroutine initialt

      !cccccccccccccccccccccc
      subroutine initialv(EV,yv
      !  Initial conditions for

      implicit none
      real(dl) bigR,Rc,tau,x,pi
      type(EvolutionVars) EV
      real(dl) k,k2 ,a, omtau
      real(dl) yv(EV%nvarv)

      if (CP%flat) then
          EV%k2_buf=EV%q2
          EV%k_buf=EV%q
      else
          call MpiStop('Vectors
      endif

      k=EV%k_buf
      k2=EV%k2_buf
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1934

```

      omtau = tau*(grhob+grhoc)

      a=tau*adotrad*(1+omtau/4)

      x=k*tau

      bigR = (grhornomass)/(grh
      Rc=CP%omegac/(CP%omegac+C

      yv(1)=a


      yv(2)= vec_sig0*(1- 15._d
      !qg
      yv(4)= vec_sig0/3* (4*big
      (1 - 0.25_dl*omtau*(3*Rc-
      -x/2*Magnetic
      yv(3)= 3._dl/4*yv(4)

      yv(5:EV%nvarv) = 0

      !         if (.false.) the
      !          yv((EV%lmaxv-1+
      !          yv((EV%lmaxv-1+
      !          yv((EV%lmaxv-1+
      !          yv((EV%lmaxv-1+
      !          yv(4) = 0
      !          yv(3)= 3._dl/4*
      !           return
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1691

```

      omtau = tau*(grhob+grhoc)

      a=tau*adotrad*(1+omtau/4)

      x=k*tau

      bigR = (grhornomass)/(grh
      Rc=CP%omegac/(CP%omegac+C

      yv(1)=a


      yv(2)= vec_sig0*(1- 15._d
      !qg
      yv(4)= vec_sig0/3* (4*big
          (1 - 0.25_dl*omtau*(3
          -x/2*Magnetic
      yv(3)= 3._dl/4*yv(4)

      yv(5:EV%nvarv) = 0

      !         if (.false.) the
      !          yv((EV%lmaxv-1+
      !          yv((EV%lmaxv-1+
      !          yv((EV%lmaxv-1+
      !          yv((EV%lmaxv-1+
      !          yv(4) = 0
      !          yv(3)= 3._dl/4*
      !           return
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1964

```
1964           !          end if
1965
1966           !  Neutrinos
1967           !q_r
1968           yv((EV%lmaxv-1+1)+(EV%lma
1969           + x**2*vec_sig0/6/BigR +0
1970           !pi_r
1971           pir=-2._dl/3._dl*x*vec_si
1972           yv((EV%lmaxv-1+1)+(EV%lma
1973           yv((EV%lmaxv-1+1)+(EV%lma
1974
1975           end subroutine initialv
1976
1977
1978           subroutine outtransf(EV,
1979           !write out clxc, clxb, cl
1980           implicit none
1981           type(EvolutionVars) EV
1982           real(dl), intent(in) :: t
1983           real(dl) clxc, clxb, clxg
1984           real(dl) grho,gpres,dgrho
1985
1986       !#SimDataAdd
1987           real(dl) a21,grhob_t1,grh
1988           integer i1
1989       !#SimDataAdd
1990           real, target :: Arr(:)
1991           real(dl) y(EV%nvar),yprim
1992
1993       !ana simdata: atentie aici tr
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1721

```
1721           !          end if
1722
1723           !  Neutrinos
1724           !q_r
1725           yv((EV%lmaxv-1+1)+(EV%lma
1726                   + x**2*vec_sig0/6/Big
1727           !pi_r
1728           pir=-2._dl/3._dl*x*vec_si
1729           yv((EV%lmaxv-1+1)+(EV%lma
1730           yv((EV%lmaxv-1+1)+(EV%lma
1731
1732           end subroutine initialv
1733
1734
1735           subroutine outtransf(EV,
1736           !write out clxc, clxb, cl
1737           implicit none
1738           type(EvolutionVars) EV
1739           real(dl), intent(in) :: t
1740           real, target :: Arr(:)
1741           real(dl) y(EV%nvar),yprim
1742
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1994

```

    yprime = 0
    EV%OutputTransfer =>  Arr
    call derivs(EV,EV%ScalEqs
    nullify(EV%OutputTransfer

    Arr(Transfer_kh+1:Transfe

   !  do i1=Transfer_kh, Tran
            !print*, 'equatio

   !  end do

!#SimDataAdd +ana add

    a    = y(1)
    clxc = y(3)
    clxb = y(4)

    k    = EV%k_buf
    k2   = EV%k2_buf

    dgrho = 0
    grho =  0

    if (CP%Num_Nu_Massive > 0
        call MassiveNuVars(EV
    end if

    dgrho = dgrho+(clxc*grhoc
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1743

```
    yprime = 0
    EV%OutputTransfer =>  Arr
    call derivs(EV,EV%ScalEqs
    nullify(EV%OutputTransfer

    Arr(Transfer_kh+1:Transfe

```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2024

```
        grho =  grho+(grhoc+grhob

        a21 = a*a


        vb1 = y(5)

        grhob_t1 = grhob/a
        grhoc_t1 = grhoc/a
        grhor_t1 = grhornomass/a2
        grhog_t1 = grhog/a21
        grhov_t1 = grho_de(a)/a21

        grho = grhob_t1+grhoc_t1+

        adotoa1 = sqrt(grho/3.d0)

        dgq = grhob_t1*vb1

        phi1 = -(dgrho + 3.d0*dgq

        Arr(Transfer_Psi) = -phi1

 !print*, 'equations: i =', T
  !  print*, 'dgrho, dqq, k,
   ! print*, 'equations I: i

!#SimDataAdd

        end subroutine outtransf
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1750

```
        end subroutine outtransf
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2054

```

        !cccccccccccccccccccccc
        subroutine derivs(EV,n,ta
        !  Evaluate the time deri
        !  ayprime is not necessa
        use ThermoData
        use MassiveNu
        implicit none
        type(EvolutionVars) EV

        integer n,nu_i
        real(dl) ay(n),ayprime(n)
        real(dl) tau,w
        real(dl) k,k2

        !  Internal variables.

        real(dl) opacity
        real(dl) photbar,cs2,pb43
        clxcdot,clxbdot,adotdota,
        real(dl) q,aq,v
        real(dl) G11_t,G30_t, wnu

        real(dl) dgq,grhob_t,grho
        real(dl) qgdot,qrdot,pigd
        real(dl) a,a2,z,clxc,clxb
        real(dl) clxq, vq,  E2, d
        integer l,i,ind, ind2, of
        real(dl) dgs,sigmadot,dz
        real(dl) dgpi,dgrho_matte
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1751

```

        !cccccccccccccccccccccc
        subroutine derivs(EV,n,ta
        !  Evaluate the time deri
        !  ayprime is not necessa
        use ThermoData
        use MassiveNu
        implicit none
        type(EvolutionVars) EV

        integer n,nu_i
        real(dl) ay(n),ayprime(n)
        real(dl) tau,w
        real(dl) k,k2

        !  Internal variables.

        real(dl) opacity
        real(dl) photbar,cs2,pb43
            clxcdot,clxbdot,adotd
        real(dl) q,aq,v
        real(dl) G11_t,G30_t, wnu

        real(dl) dgq,grhob_t,grho
        real(dl) qgdot,qrdot,pigd
        real(dl) a,a2,z,clxc,clxb
        real(dl) clxde, qde,  E2,
        integer l,i,ind, ind2, of
        real(dl) dgs,sigmadot,dz
        real(dl) dgpi,dgrho_matte
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2084

```
      !non-flat vars
      real(dl) cothxor !1/tau i

  !ana simdata
  !    #SimDataAdd
      real(dl) phi
    !#SimDataAdd
  !end ana simdata


      k=EV%k_buf
      k2=EV%k2_buf

      a=ay(1)
      a2=a*a

      etak=ay(2)

      !   CDM variables
      clxc=ay(3)

      !   Baryon variables
      clxb=ay(4)
      vb=ay(5)

      !   Compute expansion rate

      grhob_t=grhob/a
      grhoc_t=grhoc/a
      grhor_t=grhornomass/a2
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1781

```
      !non-flat vars
      real(dl) cothxor !1/tau i
      !Variables for source cal
      real(dl) diff_rhopi, pido
      real(dl) E(2:3), Edot(2:3
      real(dl) phidot, polterdo
      real(dl) ddopacity, visib
      real(dl) ISW, quadrupole_

      k=EV%k_buf
      k2=EV%k2_buf

      a=ay(1)
      a2=a*a

      etak=ay(2)

      !   CDM variables
      clxc=ay(3)

      !   Baryon variables
      clxb=ay(4)
      vb=ay(5)

      !   Compute expansion rate

      grhob_t=grhob/a
      grhoc_t=grhoc/a
      grhor_t=grhornomass/a2
```

/Users/lplopa/Compare/camb_simdata/equations.f90, Top line: 2114

```
2114        grhog_t=grhog/a2
2115
2116    !#SimDataReplace
2117         grhov_t=grho_de(a)/a2
2118
2119    !    if (w_lam==-1._dl) then
2120    !        grhov_t=grhov*a2
2121    !    else
2122    !        grhov_t=grhov*a**(-1
2123    !    end if
2124    !#SimDataReplace
2125
2126        !  Get sound speed and io
2127        if (EV%TightCoupling) the
2128            call thermo(tau,cs2,o
2129        else
2130            call thermo(tau,cs2,o
2131        end if
2132
2133        gpres=0
2134        grho_matter=grhob_t+grhoc
2135
2136        !total perturbations: mat
2137        !  8*pi*a*a*SUM[rho_i*clx
2138        dgrho_matter=grhob_t*clxb
2139        !  8*pi*a*a*SUM[(rho_i+p_
2140        dgq=grhob_t*vb
2141
2142        if (CP%Num_Nu_Massive > 0
2143            call MassiveNuVars(EV
```

/Users/lplopa/Compare/camb_des/equations.f90, Top line: 1810

```
1810        grhog_t=grhog/a2
1811        if (w_lam==-1._dl) then
1812            grhov_t=grhov*a2
1813        else
1814            grhov_t=grhov*a**(-1-
1815        end if
1816
1817        !  Get sound speed and io
1818        if (EV%TightCoupling) the
1819            call thermo(tau,cs2,o
1820        else
1821            call thermo(tau,cs2,o
1822        end if
1823
1824        gpres_nu=0
1825        grhonu_t=0
1826
1827        !total perturbations: mat
1828        !  8*pi*a*a*SUM[rho_i*clx
1829        dgrho_matter=grhob_t*clxb
1830        !  8*pi*a*a*SUM[(rho_i+p_
1831        dgq=grhob_t*vb
1832
1833        if (CP%Num_Nu_Massive > 0
1834            call MassiveNuVars(EV
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2144

```
2144        end if
2145

2146        grho = grho_matter+grhor_
2147
2148        if (CP%flat) then
2149            adotoa=sqrt(grho/3)
2150            cothxor=1._dl/tau
2151        else
2152            adotoa=sqrt((grho+grh
2153            cothxor=1._dl/tanfunc
2154        end if
2155
2156        dgrho = dgrho_matter
2157
2158    !#SimDataReplace
2159    !    if (w_lam /= -1 .and. w_
2160     if (.not. is_cosmological_co
2161            clxq=ay(EV%w_ix)
2162            vq=ay(EV%w_ix+1)
2163            dgrho=dgrho + clxq*gr
2164    !        dgq = dgq + vq*grhov
2165
2166     dgq = dgq + vq*grhov_t*(1+w_
2167        end if
2168    !print*, 'GAINA: ', dgrho, gr
2169    !#SimDataReplace
2170
2171        if (EV%no_nu_multpoles) t
2172            !RSA approximation of
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1835

```
1835        end if
1836
1837        grho_matter=grhonu_t+grho
1838        grho = grho_matter+grhor_
1839
1840        if (CP%flat) then
1841            adotoa=sqrt(grho/3)
1842            cothxor=1._dl/tau
1843        else
1844            adotoa=sqrt((grho+grh
1845            cothxor=1._dl/tanfunc
1846        end if
1847
1848        dgrho = dgrho_matter
1849
1850        if (w_lam /= -1 .and. w_P
1851            clxde=ay(EV%w_ix)
1852            qde=ay(EV%w_ix+1)*(1+
1853            dgrho=dgrho + clxde*g
1854            dgq = dgq + qde*grhov



1855        end if


1856
1857        if (EV%no_nu_multpoles) t
1858            !RSA approximation of
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2173

```
                !Approximate total de
                z=(0.5_dl*dgrho/k + e
                dz= -adotoa*z - 0.5_d
                clxr=-4*dz/k
                qr=-4._dl/3*z
                pir=0
            else
                !  Massless neutrinos
                clxr=ay(EV%r_ix)
                qr  =ay(EV%r_ix+1)
                pir =ay(EV%r_ix+2)
            endif

            if (EV%no_phot_multpoles)
                if (.not. EV%no_nu_mu
                    z=(0.5_dl*dgrho/k
                    dz= -adotoa*z - 0
                    clxg=-4*dz/k-4/k*
                    qg=-4._dl/3*z
                else
                    clxg=clxr-4/k*opa
                    qg=qr
                end if
                pig=0
            else
                !  Photons
                clxg=ay(EV%g_ix)
                qg=ay(EV%g_ix+1)
                if (.not. EV%TightCou
            end if
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1859

```
                !Approximate total de
                z=(0.5_dl*dgrho/k + e
                dz= -adotoa*z - 0.5_d
                clxr=-4*dz/k
                qr=-4._dl/3*z
                pir=0
            else
                !  Massless neutrinos
                clxr=ay(EV%r_ix)
                qr  =ay(EV%r_ix+1)
                pir =ay(EV%r_ix+2)
            endif

            if (EV%no_phot_multpoles)
                if (.not. EV%no_nu_mu
                    z=(0.5_dl*dgrho/k
                    dz= -adotoa*z - 0
                    clxg=-4*dz/k-4/k*
                    qg=-4._dl/3*z
                else
                    clxg=clxr-4/k*opa
                    qg=qr
                end if
                pig=0
            else
                !  Photons
                clxg=ay(EV%g_ix)
                qg=ay(EV%g_ix+1)
                if (.not. EV%TightCou
            end if
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2203

```

    !   8*pi*a*a*SUM[rho_i*clx
    dgrho=dgrho + grhog_t*clx

    !   8*pi*a*a*SUM[(rho_i+p_
    dgq=dgq + grhog_t*qg+grho

    !   Photon mass density ov
    photbar=grhog_t/grhob_t
    pb43=4._dl/3*photbar

    ayprime(1)=adotoa*a


    !   Get sigma (shear) and
    ! have to get z from eta
    z=(0.5_dl*dgrho/k + etak)
    if (CP%flat) then
        !eta*k equation
        sigma=(z+1.5_dl*dgq/k
        ayprime(2)=0.5_dl*dgq
    else
        sigma=(z+1.5_dl*dgq/k
        ayprime(2)=0.5_dl*dgq
    end if

!#SimDataReplace
!    if (w_lam /= -1 .and. w_
     if (.not. is_cosmologica
  ayprime(EV%w_ix)= 0.d0 !-3*
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1889

```

    !   8*pi*a*a*SUM[rho_i*clx
    dgrho=dgrho + grhog_t*clx

    !   8*pi*a*a*SUM[(rho_i+p_
    dgq=dgq + grhog_t*qg+grho

    !   Photon mass density ov
    photbar=grhog_t/grhob_t
    pb43=4._dl/3*photbar

    ayprime(1)=adotoa*a


    !   Get sigma (shear) and
    ! have to get z from eta
    z=(0.5_dl*dgrho/k + etak)
    if (CP%flat) then
        !eta*k equation
        sigma=(z+1.5_dl*dgq/k
        ayprime(2)=0.5_dl*dgq
    else
        sigma=(z+1.5_dl*dgq/k
        ayprime(2)=0.5_dl*dgq
    end if

    if (w_lam /= -1 .and. w_P
        ayprime(EV%w_ix)= -3*
            - k*qde -(1+w_lam

!        ayprime(EV%w_ix)= -3
!        -(1+w_lam)*k*vq -(1+

!        ayprime(EV%w_ix+1) =

        ayprime(EV%w_ix+1) =
        !-adotoa*a*CP%wa/(1.d
    end if

!#SimDataReplace

    if (associated(EV%OutputT
        EV%OutputTransfer(Tra
        EV%OutputTransfer(Tra
        EV%OutputTransfer(Tra
        EV%OutputTransfer(Tra
        EV%OutputTransfer(Tra
        clxnu_all=0
        dgpi  = grhor_t*pir +
        if (CP%Num_Nu_Massive
            call MassiveNuVar
        end if
        EV%OutputTransfer(Tra
        EV%OutputTransfer(Tra
        EV%OutputTransfer(Tra
        EV%OutputTransfer(Tra
        !Transfer_Weyl is k^2
        EV%OutputTransfer(Tra
        EV%OutputTransfer(Tra
        EV%OutputTransfer(Tra
        EV%OutputTransfer(Tra
!ana simdata

        !print*,'EV%OutputTra
        !print*, 'EV%OutputTr
        !print*,' EV%OutputTr
        !print*,' EV%OutputTr
        !print*, 'EV%OutputTr
        !print*, 'EV%OutputTr
        !print*, 'EV%OutputTr
        !print*, 'EV%OutputTr
        !print*, 'EV%OutputTr
        !Transfer_Weyl is k^2
        !print*,'EV%OutputTra
        !print*, 'EV%OutputTr
        !print*,  'EV%OutputT
        !print*, 'EV%OutputTr

 a2 = a*a

    vb = ay(5)

    grhob_t = grhob/a
    grhoc_t = grhoc/a
    grhor_t = grhornomass/a2
    grhog_t = grhog/a2
    grhov_t = grho_de(a)/a2

    grho = grhob_t+grhoc_t+gr
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2293

```
2293
2294         adotoa = sqrt(grho/3.d0)
2295
2296         dgq = grhob_t*vb
2297
2298         phi = -(dgrho + 3.d0*dgq*
2299
2300         EV%OutputTransfer(Transfe
2301
2302         !print*, 'GAINA dgrho, dq
2303
2304          !print*, 'GAINA equation
2305
2306          !print*, 'k2: ', k2
2307 !end ana simdata
2308         end if
2309
2310         !  CDM equation of motion
2311         clxcdot=-k*z
2312         ayprime(3)=clxcdot
2313
2314         !  Baryon equation of mot
2315         clxbdot=-k*(z+vb)
2316         ayprime(4)=clxbdot
2317         !  Photon equation of mot
2318         clxgdot=-k*(4._dl/3._dl*z
2319
2320         ! old comment:Small k: po
2321         ! Easy to see instability
2322
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1918

```
1918
1919             ayprime(EV%w_ix+1) =
1920         end if
1921
1922         !  CDM equation of motion
1923         clxcdot=-k*z
1924         ayprime(3)=clxcdot
1925
1926         !  Baryon equation of mot
1927         clxbdot=-k*(z+vb)
1928         ayprime(4)=clxbdot
1929         !  Photon equation of mot
1930         clxgdot=-k*(4._dl/3._dl*z
1931
1932         ! old comment:Small k: po
1933         ! Easy to see instability
1934
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2323

```
    !  Use explicit equation

    if (EV%TightCoupling) the
        !  ddota/a

!#SimDataReplace
!        gpres=gpres+ (grhog_

        gpres=gpres+ (grhog_t
!#SimDataReplace


        adotdota=(adotoa*adot

        pig = 32._dl/45/opaci

        !  First-order approx
        slip = - (2*adotoa/(1
        +(-adotdota*vb-k/2*ad

        if (second_order_tigh
            ! by Francis-Yan
            !AL: First order

            !  8*pi*G*a*a*SUM
            dgs = grhog_t*pig

            ! Define shear de
            sigmadot = -2*ado

```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1935

```
    !  Use explicit equation

    if (EV%TightCoupling) the
        !  ddota/a
        gpres=gpres_nu+ (grho
        adotdota=(adotoa*adot

        pig = 32._dl/45/opaci

        !  First-order approx
        slip = - (2*adotoa/(1
            +(-adotdota*vb-k/

        if (second_order_tigh
            ! by Francis-Yan
            !AL: First order

            !  8*pi*G*a*a*SUM
            dgs = grhog_t*pig

            ! Define shear de
            sigmadot = -2*ado

```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2353

```
2353            !Once know slip,
2354            qgdot = k*(clxg/4
2355
2356            pig = 32._dl/45/o
2357            + (32._dl/45._dl/
2358
2359            pigdot = -(32._dl
2360            dopacity*11._dl/6
2361            + (32._dl/45._dl/
2362            *(dopacity/opacit
2363
2364            EV%pigdot = pigdo

2365          end if
2366
2367          !  Use tight-coupling
2368          !  zeroth order appro
2369          vbdot=(-adotoa*vb+cs2
2370          +k/4*pb43*(clxg-2*EV%
2371
2372          vbdot=vbdot+pb43/(1+p
2373
2374          EV%pig = pig

2375        else
2376          vbdot=-adotoa*vb+cs2*
2377        end if
2378
2379        ayprime(5)=vbdot
2380
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1958

```
1958            !Once know slip,
1959            qgdot = k*(clxg/4
1960
1961            pig = 32._dl/45/o
1962              + (32._dl/45.
1963
1964            pigdot = -(32._dl
1965              dopacity*11._
1966              + (32._dl/45.
1967              *(dopacity/op
1968
1969            EV%pigdot = pigdo
1970
1971          end if
1972
1973          !  Use tight-coupling
1974          !  zeroth order appro
1975          vbdot=(-adotoa*vb+cs2
1976            +k/4*pb43*(clxg-2
1977
1978          vbdot=vbdot+pb43/(1+p

1979          EV%pig = pig
1980
1981        else
1982          vbdot=-adotoa*vb+cs2*
1983        end if
1984
1985        ayprime(5)=vbdot
1986
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2381

```
2381       if (.not. EV%no_phot_mult
2382           !  Photon equations o
2383           ayprime(EV%g_ix)=clxg
2384           qgdot=4._dl/3*(-vbdot
2385           +EV%denlk(1)*clxg-EV%
2386           ayprime(EV%g_ix+1)=qg
2387
2388           !  Use explicit equat
2389           if (.not. EV%tightcou
2390               E2=ay(EV%polind+2
2391               polter = pig/10+9
2392               ix= EV%g_ix+2
2393               if (EV%lmaxg>2) t
2394                   pigdot=EV%den
2395                   +8._dl/15._dl
2396                   ayprime(ix)=p
2397                   do  l=3,EV%lm
2398                       ix=ix+1
2399                       ayprime(i
2400                   end do
2401                   ix=ix+1
2402                   !  Truncate t
2403                   ayprime(ix)=k
2404               else !closed case
2405                   pigdot=EV%den
2406                   ayprime(ix)=p
2407               endif
2408               !  Polarization
2409               !l=2
2410               ix=EV%polind+2
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1987

```
1987       if (.not. EV%no_phot_mult
1988           !  Photon equations o
1989           ayprime(EV%g_ix)=clxg
1990           qgdot=4._dl/3*(-vbdot
1991               +EV%denlk(1)*clxg
1992           ayprime(EV%g_ix+1)=qg
1993
1994           !  Use explicit equat
1995           if (.not. EV%tightcou
1996               E2=ay(EV%polind+2
1997               polter = pig/10+9
1998               ix= EV%g_ix+2
1999               if (EV%lmaxg>2) t
2000                   pigdot=EV%den
2001                       +8._dl/15
2002                   ayprime(ix)=p
2003                   do  l=3,EV%lm
2004                       ix=ix+1
2005                       ayprime(i
2006                   end do
2007                   ix=ix+1
2008                   !  Truncate t
2009                   ayprime(ix)=k
2010               else !closed case
2011                   pigdot=EV%den
2012                   ayprime(ix)=p
2013               endif
2014               !  Polarization
2015               !l=2
2016               ix=EV%polind+2
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2411

```
                if (EV%lmaxgpol>2
                    ayprime(ix) =
                    do l=3,EV%lma
                        ix=ix+1
                        ayprime(i
                    end do
                    ix=ix+1
                    !truncate
                    ayprime(ix)=-
                    k*EV%poltrunc
                else !closed case
                    ayprime(ix) =
                endif
            end if
        end if

        if (.not. EV%no_nu_multpo
            !  Massless neutrino
            clxrdot=-k*(4._dl/3._
            ayprime(EV%r_ix)=clxr
            qrdot=EV%denlk(1)*clx
            ayprime(EV%r_ix+1)=qr
            if (EV%high_ktau_neut
                !ufa approximatio
                !Method from arXi
                !
                !
                !
                !
                !
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2017

```
                if (EV%lmaxgpol>2
                    ayprime(ix) =
                    do l=3,EV%lma
                        ix=ix+1
                        ayprime(i
                    end do
                    ix=ix+1
                    !truncate
                    ayprime(ix)=-
                        k*EV%polt
                else !closed case
                    ayprime(ix) =
                endif
            end if
        end if

        if (.not. EV%no_nu_multpo
            !  Massless neutrino
            clxrdot=-k*(4._dl/3._
            ayprime(EV%r_ix)=clxr
            qrdot=EV%denlk(1)*clx
            ayprime(EV%r_ix+1)=qr
            if (EV%high_ktau_neut
                !ufa approximatio
                !Method from arXi
                !
                !
                !
                !
                !
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2441

```
              !
              !
              !
              !
              pirdot= -3*pir*co
              ayprime(EV%r_ix+2

              !
              !
              !
              !
              !               a
          else
              ix=EV%r_ix+2
              if (EV%lmaxnr>2)
                  pirdot=EV%den
                  ayprime(ix)=p
                  do l=3,EV%lma
                      ix=ix+1
                      ayprime(i
                  end do
                  !  Truncate t
                  ix=ix+1
                  ayprime(ix)=k
              else
                  pirdot=EV%den
                  ayprime(ix)=p
              end if
          end if
      end if ! no_nu_multpoles
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2047

```
              !
              !
              !
              !
              pirdot= -3*pir*co
              ayprime(EV%r_ix+2

              !
              !
              !
              !
              !               a
          else
              ix=EV%r_ix+2
              if (EV%lmaxnr>2)
                  pirdot=EV%den
                  ayprime(ix)=p
                  do l=3,EV%lma
                      ix=ix+1
                      ayprime(i
                  end do
                  !  Truncate t
                  ix=ix+1
                  ayprime(ix)=k
              else
                  pirdot=EV%den
                  ayprime(ix)=p
              end if
          end if
      end if ! no_nu_multpoles
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2471

```
2471
2472          !  Massive neutrino equat
2473          if (CP%Num_Nu_massive ==
2474
2475          do nu_i = 1, CP%Nu_mass_e
2476              if (EV%MassiveNuAppro
2477                  !Now EV%iq0 = clx
2478                  !see astro-ph/020
2479                  G11_t=EV%G11(nu_i
2480                  G30_t=EV%G30(nu_i
2481                  off_ix = EV%nu_ix
2482                  w=wnu_arr(nu_i)
2483                  ayprime(off_ix)=-
2484                  ayprime(off_ix+1)
2485                  ayprime(off_ix+2)
2486                  ayprime(off_ix+3)
2487              else
2488                  ind=EV%nu_ix(nu_i

2489                  do i=1,EV%nq(nu_i
2490                      q=nu_q(i)
2491                      aq=a*nu_masse
2492                      v=1._dl/sqrt(
2493
2494                      ayprime(ind)=
2495                      ind=ind+1
2496                      ayprime(ind)=
2497                      ind=ind+1
2498                      if (EV%lmaxnu
2499                          ayprime(i
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2077

```
2077
2078          !  Massive neutrino equat
2079          if (CP%Num_Nu_massive >0)
2080              !DIR$ LOOP COUNT MIN(
2081              do nu_i = 1, CP%Nu_ma
2082                  if (EV%MassiveNuA
2083                      !Now EV%iq0 =
2084                      !see astro-ph
2085                      G11_t=EV%G11(
2086                      G30_t=EV%G30(
2087                      off_ix = EV%n
2088                      w=wnu_arr(nu_
2089                      ayprime(off_i
2090                      ayprime(off_i
2091                      ayprime(off_i
2092                      ayprime(off_i
2093                  else
2094                      ind=EV%nu_ix(
2095                      !DIR$ LOOP CO
2096                      do i=1,EV%nq(
2097                          q=nu_q(i)
2098                          aq=a*nu_m
2099                          v=1._dl/s
2100
2101                          ayprime(i
2102                          ind=ind+1
2103                          ayprime(i
2104                          ind=ind+1
2105                          if (EV%lm
2106                              aypri
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2500

```
                  else
                      ayprime(i
                      +k*8._dl/
                      do l=3,EV
                          ind=i
                          aypri
                      end do
                      !  Trunca
                      ind = ind
                      ayprime(i
                  end if
                  ind = ind+1
              end do
          end if
      end do

      if (EV%has_nu_relativisti
          ind=EV%nu_pert_ix
          ayprime(ind)=+k*a2*qr
          ind2= EV%r_ix
          do l=1,EV%lmaxnu_pert
              ind=ind+1
              ind2=ind2+1
              ayprime(ind)= -a2
              +    (EV%denlk(l)*
          end do
          ind=ind+1
          ind2=ind2+1
          ayprime(ind)= k*(ay(i
      end if
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2107

```
                      else
                          aypri
                              +
                          do l=
                              i
                              a
                          end d
                          !  Tr
                          ind =
                          aypri
                      end if
                      ind = ind
                  end do
              end if
          end do

          if (EV%has_nu_relativ
              ind=EV%nu_pert_ix
              ayprime(ind)=+k*a
              ind2= EV%r_ix
              do l=1,EV%lmaxnu_
                  ind=ind+1
                  ind2=ind2+1
                  ayprime(ind)=
                      +    (EV%d
              end do
              ind=ind+1
              ind2=ind2+1
              ayprime(ind)= k*(
          end if
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2530

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2137

```
2137  end if
2138
2139  if (associated(EV%OutputT
2140    if (EV%TightCoupling
2141      E=0
2142      Edot=0
2143    else
2144      E = ay(EV%polind+
2145      Edot = ayprime(EV
2146    end if
2147    if (EV%no_nu_multpole
2148      pirdot=0
2149      qrdot = -4*dz/3
2150    end if
2151    if (EV%no_phot_multpo
2152      pigdot=0
2153      octg=0
2154      octgdot=0
2155      qgdot = -4*dz/3
2156    else
2157      if (EV%TightCoupl
2158        if (second_or
2159          octg = (3
2160          E(2) = pi
2161          E(3) = (3
2162          Edot(2)=
2163        else
2164          pigdot =
2165            +etak
2166          Edot(2) =
                    E(2) = pi
                    octg=0
                end if
                octgdot=0
            else
                octg=ay(EV%g_
                octgdot=aypri
            end if
        end if

        dgpi  = grhor_t*pir +
        dgpi_diff = 0  !sum (
        pidot_sum = grhog_t*p
        clxnu =0
        if (CP%Num_Nu_Massive
            call MassiveNuVar
                dgpi_diff=dgp
        end if
        diff_rhopi = pidot_su
        gpres=gpres_nu+ (grho

        phi = -((dgrho +3*dgq

        if (associated(EV%Out
            EV%OutputTransfer
            EV%OutputTransfer
            EV%OutputTransfer
            EV%OutputTransfer
            EV%OutputTransfer
            EV%OutputTransfer
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2530

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2197

```
            EV%OutputTransfer
            EV%OutputTransfer
            EV%OutputTransfer
            !Transfer_Weyl is
            EV%OutputTransfer
            EV%OutputTransfer
            EV%OutputTransfer
            EV%OutputTransfer
        end if
        if (associated(EV%Out

            call IonizationFu
                visibility, d

            tau0 = CP%tau0
            phidot = (1.0d0/2
                diff_rhopi+ k
            !time derivative
            sigmadot = -adoto
            !quadrupole sourc
            polter = pig/10+9
            polterdot = (1.0d
            polterddot = -2.0
                k*sigma - 4.0
                50.0d0*k*octg
                *k*EV%Kf(2)*E
                polter - 3.0d
                + (7.0d0/10.0
            !Temperature sour

```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2530

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2227

```
            !2phi' term (\phi
            ISW = 2*phidot*ex
            monopole_source =
            doppler = ((sigma
            quadrupole_source
                + (k**2*polte

            EV%OutputSources(

            if (tau < tau0) t
                !E polarizati
                EV%OutputSour
                !factor of fo
            else
                EV%OutputSour
            end if

            if (size(EV%Outpu
                !Get lensing
                !Can modify t
                if (tau>tau_m
                    EV%Output
                    !We inclu
                else
                    EV%Output
                end if
            end if
            if (associated(EV
                call custom_s
                    grhob_t,g
```

```
2530
2531          end subroutine derivs
2532
2533
2534
2535          subroutine derivsv(EV,n,t
2536          !  Evaluate the time deri
2537          use ThermoData
2538          use MassiveNu
2539          implicit none
2540          type(EvolutionVars) EV
2541          integer n,l
2542          real(dl), target ::  yv(n
2543          real(dl) ep,tau,grho,rhop
2544          logical finished_tightcou
2545          real(dl), dimension(:),po
2546          real(dl)  grhob_t,grhor_t
2547          real(dl) sigma, qg,pig, q
2548          real(dl) k,k2,a,a2, adotd
2549          real(dl) pir,adotoa
```

```
2257                                    k, etak,
2258                                    dgrho, cl
2259                                    dgq, qg,
2260                                    dgpi, pig
2261                                    polter, p
2262                                    opacity,
2263                                    tau0, tau
2264                            end if
2265                        end if
2266                    end if
2267
2268          end subroutine derivs
2269
2270
2271
2272          subroutine derivsv(EV,n,t
2273          !  Evaluate the time deri
2274          use ThermoData
2275          use MassiveNu
2276          implicit none
2277          type(EvolutionVars) EV
2278          integer n,l
2279          real(dl), target ::  yv(n
2280          real(dl) ep,tau,grho,rhop
2281          logical finished_tightcou
2282          real(dl), dimension(:),po
2283          real(dl)  grhob_t,grhor_t
2284          real(dl) sigma, qg,pig, q
2285          real(dl) k,k2,a,a2, adotd
2286          real(dl) pir,adotoa
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2550

```

        k2=EV%k2_buf
        k=EV%k_buf

        !E and B start at l=2. Se
        E => yv(EV%lmaxv+3:)
        Eprime=> yvprime(EV%lmaxv
        B => E(EV%lmaxpolv:)
        Bprime => Eprime(EV%lmaxp
        neutprime => Bprime(EV%lm
        neut => B(EV%lmaxpolv+1:)

        a=yv(1)

        sigma=yv(2)

        a2=a*a

        !  Get sound speed and op

        call thermo(tau,cs2,opaci
        if (k > 0.06_dl*epsw) the
            ep=ep0
        else
            ep=0.2_dl*ep0
        end if

        finished_tightcoupling =
        ((k/opacity > ep).or.(1._

```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2287

```

        k2=EV%k2_buf
        k=EV%k_buf

        !E and B start at l=2. Se
        E => yv(EV%lmaxv+3:)
        Eprime=> yvprime(EV%lmaxv
        B => E(EV%lmaxpolv:)
        Bprime => Eprime(EV%lmaxp
        neutprime => Bprime(EV%lm
        neut => B(EV%lmaxpolv+1:)

        a=yv(1)

        sigma=yv(2)

        a2=a*a

        !  Get sound speed and op

        call thermo(tau,cs2,opaci
        if (k > 0.06_dl*epsw) the
            ep=ep0
        else
            ep=0.2_dl*ep0
        end if

        finished_tightcoupling =
            ((k/opacity > ep).or.


        ! Compute expansion rate
        ! Also calculate gpres: 8
        grhob_t=grhob/a
        grhoc_t=grhoc/a
        grhor_t=grhornomass/a2
        grhog_t=grhog/a2

    !#SimDataReplace
        grhov_t=grho_de(a)/a2
        !grhov_t=grhov*a**(-1-3*w
    !#SimDataReplace


        grho=grhob_t+grhoc_t+grho
    !#SimDataReplace
    !    gpres=(grhog_t+grhor_t)/

        gpres=(grhog_t+grhor_t)/3
    !#SimDataReplace
        adotoa=sqrt(grho/3._dl)
        adotdota=(adotoa*adotoa-g

        photbar=grhog_t/grhob_t
        pb43=4._dl/3*photbar

        yvprime(1)=adotoa*a

        vb = yv(3)
        qg = yv(4)
```

```

        ! Compute expansion rate
        ! Also calculate gpres: 8
        grhob_t=grhob/a
        grhoc_t=grhoc/a
        grhor_t=grhornomass/a2
        grhog_t=grhog/a2
        grhov_t=grhov*a**(-1-3*w_

        grho=grhob_t+grhoc_t+grho
        gpres=(grhog_t+grhor_t)/3

        adotoa=sqrt(grho/3._dl)
        adotdota=(adotoa*adotoa-g

        photbar=grhog_t/grhob_t
        pb43=4._dl/3*photbar

        yvprime(1)=adotoa*a

        vb = yv(3)
        qg = yv(4)
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2610

```
2610        qr = neut(1)
2611
2612        !   8*pi*a*a*SUM[(rho_i+p_
2613        rhoq=grhob_t*vb+grhog_t*q
2614        !   sigma = 2*rhoq/k**2
2615        !for non-large k this exp
2616        !so propagate sigma equat
2617        ! print *,yv(2),2*rhoq/k*
2618
2619        if (finished_tightcouplin
2620            !   Use explicit equat
2621
2622            pig = yv(5)
2623
2624            polter = 0.1_dl*pig +
2625
2626            vbdot = -adotoa*vb-ph
2627
2628            !   Equation for the p
2629
2630            yvprime(4)=-0.5_dl*k*
2631
2632            !   Equation for the p
2633            yvprime(5)=k*(2._dl/5
2634            -opacity*(pig - polte
2635            ! And for the moments
2636            do  l=3,EV%lmaxv-1
2637                yvprime(l+3)=k*de
2638                vecfac(l)*yv(l+4)
2639            end do
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2339

```
2339        qr = neut(1)
2340
2341        !   8*pi*a*a*SUM[(rho_i+p_
2342        rhoq=grhob_t*vb+grhog_t*q
2343        !   sigma = 2*rhoq/k**2
2344        !for non-large k this exp
2345        !so propagate sigma equat
2346        ! print *,yv(2),2*rhoq/k*
2347
2348        if (finished_tightcouplin
2349            !   Use explicit equat
2350
2351            pig = yv(5)
2352
2353            polter = 0.1_dl*pig +
2354
2355            vbdot = -adotoa*vb-ph
2356
2357            !   Equation for the p
2358
2359            yvprime(4)=-0.5_dl*k*
2360
2361            !   Equation for the p
2362            yvprime(5)=k*(2._dl/5
2363                -opacity*(pig - p
2364            ! And for the moments
2365            do  l=3,EV%lmaxv-1
2366                yvprime(l+3)=k*de
2367                    vecfac(l)*yv(
2368            end do
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2640

```
2640          !  Truncate the hiera
2641          yvprime(EV%lmaxv+3)=k
2642          (EV%lmaxv+2._dl)*yv(E
2643
2644          !E equations
2645
2646          Eprime(2) = - opacity
2647          do l=3,EV%lmaxpolv-1
2648              Eprime(l) =-opaci
2649              vecfacpol(l)*E(l+
2650          end do
2651          !truncate
2652          Eprime(EV%lmaxpolv)=0
2653
2654          !B-bar equations
2655
2656          do l=2,EV%lmaxpolv-1
2657              Bprime(l) =-opaci
2658              vecfacpol(l)*B(l+
2659          end do
2660          !truncate
2661          Bprime(EV%lmaxpolv)=0
2662      else
2663          !Tight coupling expan
2664
2665          pig = 32._dl/45._dl*k
2666
2667          EV%pig = pig
2668
2669          vbdot=(-adotoa*vb  -3
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2369

```
2369          !  Truncate the hiera
2370          yvprime(EV%lmaxv+3)=k
2371              (EV%lmaxv+2._dl)*
2372
2373          !E equations
2374
2375          Eprime(2) = - opacity
2376          do l=3,EV%lmaxpolv-1
2377              Eprime(l) =-opaci
2378                  vecfacpol(l)*
2379          end do
2380          !truncate
2381          Eprime(EV%lmaxpolv)=0
2382
2383          !B-bar equations
2384
2385          do l=2,EV%lmaxpolv-1
2386              Bprime(l) =-opaci
2387                  vecfacpol(l)*
2388          end do
2389          !truncate
2390          Bprime(EV%lmaxpolv)=0
2391      else
2392          !Tight coupling expan
2393
2394          pig = 32._dl/45._dl*k
2395
2396          EV%pig = pig
2397
2398          vbdot=(-adotoa*vb  -3
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2670

```
                - pb43/(1+pb43)/opaci
                ( 2*pb43*adotoa**2/(1
                )/(1+pb43)

                !  Equation for the p
                ! Get drag from vbdot
                yvprime(4)=-0.5_dl*k*
                (vbdot+adotoa*vb)/pho

                !  Set the derivative
                yvprime(5:n)=0._dl
                yv(5)=pig
                E(2)=  pig/4
            endif

            yvprime(3) = vbdot

            !  Neutrino equations:

            !  Massless neutrino anis
            pir=neut(2)
            neutprime(1)= -0.5_dl*k*p
            neutprime(2)=2._dl/5*k*qr
            !  And for the moments
            do  l=3,EV%lmaxnrv-1
                neutprime(l)=k*denl(l
            end do

            !  Truncate the hierarchy
            neutprime(EV%lmaxnrv)=k*E
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2399

```
                    - pb43/(1+pb43)/o
                    ( 2*pb43*adotoa**
                    )/(1+pb43)

                !  Equation for the p
                ! Get drag from vbdot
                yvprime(4)=-0.5_dl*k*
                    (vbdot+adotoa*vb)

                !  Set the derivative
                yvprime(5:n)=0._dl
                yv(5)=pig
                E(2)=  pig/4
            endif

            yvprime(3) = vbdot

            !  Neutrino equations:

            !  Massless neutrino anis
            pir=neut(2)
            neutprime(1)= -0.5_dl*k*p
            neutprime(2)=2._dl/5*k*qr
            !  And for the moments
            do  l=3,EV%lmaxnrv-1
                neutprime(l)=k*denl(l
            end do

            !  Truncate the hierarchy
            neutprime(EV%lmaxnrv)=k*E
```

```
      (EV%lmaxnrv+2._dl)*neut(E


      !  Get the propagation eq

      rhopi=grhog_t*pig+grhor_t

      yvprime(2)=-2*adotoa*sigm

      end subroutine derivsv




      !cccccccccccccccccccccccc
      subroutine derivst(EV,n,t
      !  Evaluate the time deri
      use ThermoData
      use MassiveNu
      implicit none
      type(EvolutionVars) EV
      integer n,l,i,ind, nu_i
      real(dl), target ::  ayt(
      real(dl) tau,grho,rhopi,c
      real(dl), dimension(:),po
      real(dl) q,aq,v
      real(dl)  grhob_t,grhor_t
      real(dl) Hchi,pinu, pig
      real(dl) k,k2,a,a2
      real(dl) pir, adotoa, rho

```

```
          (EV%lmaxnrv+2._dl)*ne


      !  Get the propagation eq

      rhopi=grhog_t*pig+grhor_t

      yvprime(2)=-2*adotoa*sigm

      end subroutine derivsv




      !cccccccccccccccccccccccc
      subroutine derivst(EV,n,t
      !  Evaluate the time deri
      use ThermoData
      use MassiveNu
      implicit none
      type(EvolutionVars) EV
      integer n,l,i,ind, nu_i
      real(dl), target ::  ayt(
      real(dl) tau,grho,rhopi,c
      real(dl), dimension(:),po
      real(dl) q,aq,v
      real(dl)  grhob_t,grhor_t
      real(dl) Hchi,pinu, pig
      real(dl) k,k2,a,a2
      real(dl) pir, adotoa, rho

```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2730

```
    real(dl) cothxor

    k2=EV%k2_buf
    k= EV%k_buf

    a=ayt(1)

    Hchi=ayt(2)

    shear=ayt(3)

    a2=a*a

    ! Compute expansion rate
    ! Also calculate gpres: 8
    grhob_t=grhob/a
    grhoc_t=grhoc/a
    grhor_t=grhornomass/a2
    grhog_t=grhog/a2

!#SimDataReplace
!    if (w_lam==-1._dl) then
!        grhov_t=grhov*a2
!    else
!        grhov_t=grhov*a**(-1
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2459

```
    real(dl) cothxor

    k2=EV%k2_buf
    k= EV%k_buf

    a=ayt(1)

    Hchi=ayt(2)

    shear=ayt(3)

    a2=a*a

    ! Compute expansion rate
    ! Also calculate gpres: 8
    grhob_t=grhob/a
    grhoc_t=grhoc/a
    grhor_t=grhornomass/a2
    grhog_t=grhog/a2
    if (w_lam==-1._dl) then
        grhov_t=grhov*a2
    else
        grhov_t=grhov*a**(-1-
    end if

```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2755

```
!    end if
 grhov_t=grho_de(a)/a2
!#SimDataReplace
    grho=grhob_t+grhoc_t+grho

    !Do massive neutrinos
    if (CP%Num_Nu_Massive >0)
        do nu_i=1,CP%Nu_mass_
            call Nu_rho(a*nu_
            grho=grho+grhorma
        end do
    end if

    if (CP%flat) then
        cothxor=1._dl/tau
        adotoa=sqrt(grho/3._d
    else
        cothxor=1._dl/tanfunc
        adotoa=sqrt((grho+grh
    end if

    aytprime(1)=adotoa*a

    call thermo(tau,cs2,opaci

    if (.not. EV%TensTightCou
        !  Don't use tight co
        !  Equation for the p


```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2484

```
    grho=grhob_t+grhoc_t+grho

    !Do massive neutrinos
    if (CP%Num_Nu_Massive >0)
        do nu_i=1,CP%Nu_mass_
            call Nu_rho(a*nu_
            grho=grho+grhorma
        end do
    end if

    if (CP%flat) then
        cothxor=1._dl/tau
        adotoa=sqrt(grho/3._d
    else
        cothxor=1._dl/tanfunc
        adotoa=sqrt((grho+grh
    end if

    aytprime(1)=adotoa*a

    call thermo(tau,cs2,opaci

    if (.not. EV%TensTightCou
        !  Don't use tight co
        !  Equation for the p


```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2785

```
!E and B start at l=2
E => ayt(EV%E_ix+1:)
B => ayt(EV%B_ix+1:)
Eprime=> aytprime(EV%
Bprime => aytprime(EV

ind = EV%g_ix+2

!  Photon anisotropic
pig=ayt(ind)
polter = 0.1_dl*pig +

if (EV%lmaxt > 2) the
    aytprime(ind)=-EV
    -opacity*(pig - p

    do l=3, EV%lmaxt
        ind = ind+1
        aytprime(ind)
    end do

    !Truncate the hie
    ind=ind+1
    aytprime(ind)=k*E
    (EV%lmaxt+3._dl)*

    !E and B-bar equa

    Eprime(2) = - opa
    EV%denlkt(3,2)*E(
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2511

```
!E and B start at l=2
E => ayt(EV%E_ix+1:)
B => ayt(EV%B_ix+1:)
Eprime=> aytprime(EV%
Bprime => aytprime(EV

ind = EV%g_ix+2

!  Photon anisotropic
pig=ayt(ind)
polter = 0.1_dl*pig +

if (EV%lmaxt > 2) the
    aytprime(ind)=-EV
        -opacity*(pig

    do l=3, EV%lmaxt
        ind = ind+1
        aytprime(ind)
    end do

    !Truncate the hie
    ind=ind+1
    aytprime(ind)=k*E
        (EV%lmaxt+3._

    !E and B-bar equa

    Eprime(2) = - opa
        EV%denlkt(3,2
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2815

```
2815
2816                  do l=3, EV%lmaxpo
2817                      Eprime(l) =(E
2818▨                     -opacity*E(l)
2819                  end do
2820                  l= EV%lmaxpolt
2821                  !truncate: diffic
2822                  Eprime(l) = (EV%d
2823
2824                  Bprime(2) =-EV%de
2825                  do l=3, EV%lmaxpo
2826                      Bprime(l) =(E
2827▨                     -opacity*B(l)
2828                  end do
2829                  l=EV%lmaxpolt
2830                  !truncate
2831                  Bprime(l) =(EV%de
2832
2833              else !lmax=2
2834
2835▨             aytprime(ind)=k*8._dl
2836▨             Eprime(2) = - opacity
2837▨             Bprime(2) = - EV%denl
2838              end if
2839
2840          else  !Tight coupling
2841              pig = 32._dl/45._dl*k
2842          endif
2843
2844          rhopi=grhog_t*pig
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2541

```
2541
2542                  do l=3, EV%lmaxpo
2543                      Eprime(l) =(E
2544▨                         -opacity*
2545                  end do
2546                  l= EV%lmaxpolt
2547                  !truncate: diffic
2548                  Eprime(l) = (EV%d
2549
2550                  Bprime(2) =-EV%de
2551                  do l=3, EV%lmaxpo
2552                      Bprime(l) =(E
2553▨                         -opacity*
2554                  end do
2555                  l=EV%lmaxpolt
2556                  !truncate
2557                  Bprime(l) =(EV%de
2558
2559              else !lmax=2
2560
2561▨                 aytprime(ind)=k*8
2562▨                 Eprime(2) = - opa
2563▨                 Bprime(2) = - EV%
2564              end if
2565
2566          else  !Tight coupling
2567              pig = 32._dl/45._dl*k
2568          endif
2569
2570          rhopi=grhog_t*pig
```

```


      !  Neutrino equations:
      !  Anisotropic stress
      if (DoTensorNeutrinos) th
          neutprime => aytprime
          neut => ayt(EV%r_ix+1

          !  Massless neutrino
          pir=neut(2)

          rhopi=rhopi+grhor_t*p

          if (EV%lmaxnrt>2) the
              pirdt=-EV%denlkt(
              neutprime(2)=pird
             !  And for the mom
              do  l=3, EV%lmaxn
                  neutprime(l)=
              end do

              !  Truncate the h
              neutprime(EV%lmax
              (EV%lmaxnrt+3._dl
          else
              pirdt= 8._dl/15._
              neutprime(2)=pird
          end if

          !  Massive neutrino e
```

```


      !  Neutrino equations:
      !  Anisotropic stress
      if (DoTensorNeutrinos) th
          neutprime => aytprime
          neut => ayt(EV%r_ix+1

          !  Massless neutrino
          pir=neut(2)

          rhopi=rhopi+grhor_t*p

          if (EV%lmaxnrt>2) the
              pirdt=-EV%denlkt(
              neutprime(2)=pird
              !  And for the mo
              do  l=3, EV%lmaxn
                  neutprime(l)=
              end do

              !  Truncate the h
              neutprime(EV%lmax
                  (EV%lmaxnrt+3
          else
              pirdt= 8._dl/15._
              neutprime(2)=pird
          end if

          !  Massive neutrino e
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2875

```
2875            if (CP%Num_Nu_massive
2876                do nu_i=1,CP%Nu_m
2877                    if (.not. EV%
2878                        rhopi=rho
2879                    else
2880                        ind=EV%nu
2881
2882                        pinu= Nu_
2883                        rhopi=rho
2884
2885                        do i=1,nq
2886                            q=nu_
2887                            aq=a*
2888                            v=1._
2889                            if (E
2890                                a
2891                                d
2892
2893
2894                                e
2895                                i
2896                                !
2897                                a
2898                            else
2899                                a
2900                            end i
2901                            ind=i
2902                        end do
2903                    end if
2904                end do
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2601

```
2601            if (CP%Num_Nu_massive
2602                do nu_i=1,CP%Nu_m
2603                    if (.not. EV%
2604                        rhopi=rho
2605                    else
2606                        ind=EV%nu
2607
2608                        pinu= Nu_
2609                        rhopi=rho
2610
2611                        do i=1,nq
2612                            q=nu_
2613                            aq=a*
2614                            v=1._
2615                            if (E
2616                                a
2617                                d
2618
2619
2620                                e
2621                                i
2622                                !
2623                                a
2624                            else
2625                                a
2626                            end i
2627                            ind=i
2628                        end do
2629                    end if
2630                end do
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2905

```
          end if
      end if

      !  Get the propagation eq

      if (CP%flat) then
          aytprime(3)=-2*adotoa
      else
          aytprime(3)=-2*adotoa
      endif

      aytprime(2)=-k*shear

      end subroutine derivst



      !ccccccccccccccccccccccc

      end module GaugeInterface

```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2631

```
          end if
      end if

      !  Get the propagation eq

      if (CP%flat) then
          aytprime(3)=-2*adotoa
      else
          aytprime(3)=-2*adotoa
      endif

      aytprime(2)=-k*shear

      end subroutine derivst



      !ccccccccccccccccccccccc

      end module GaugeInterface

```