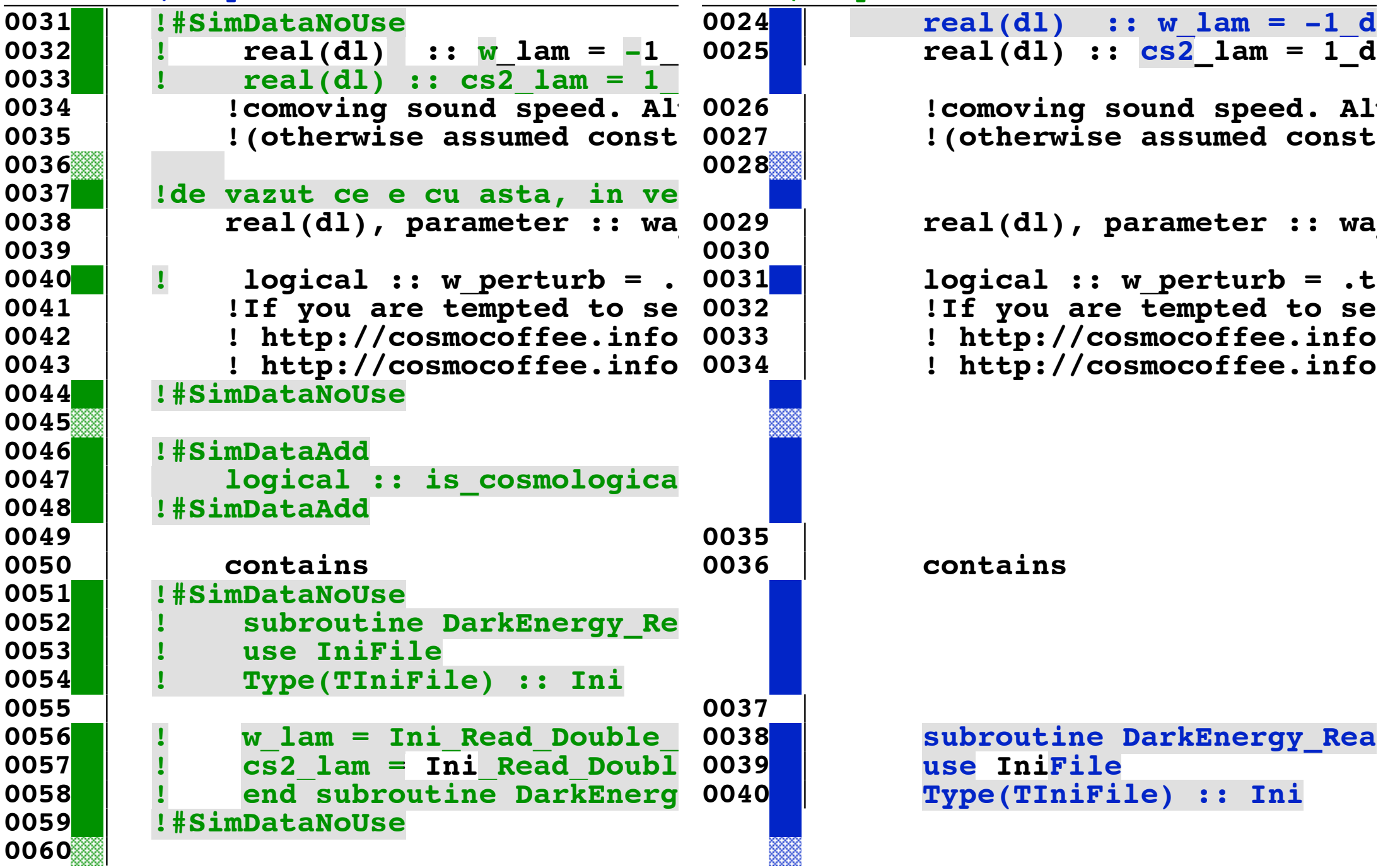
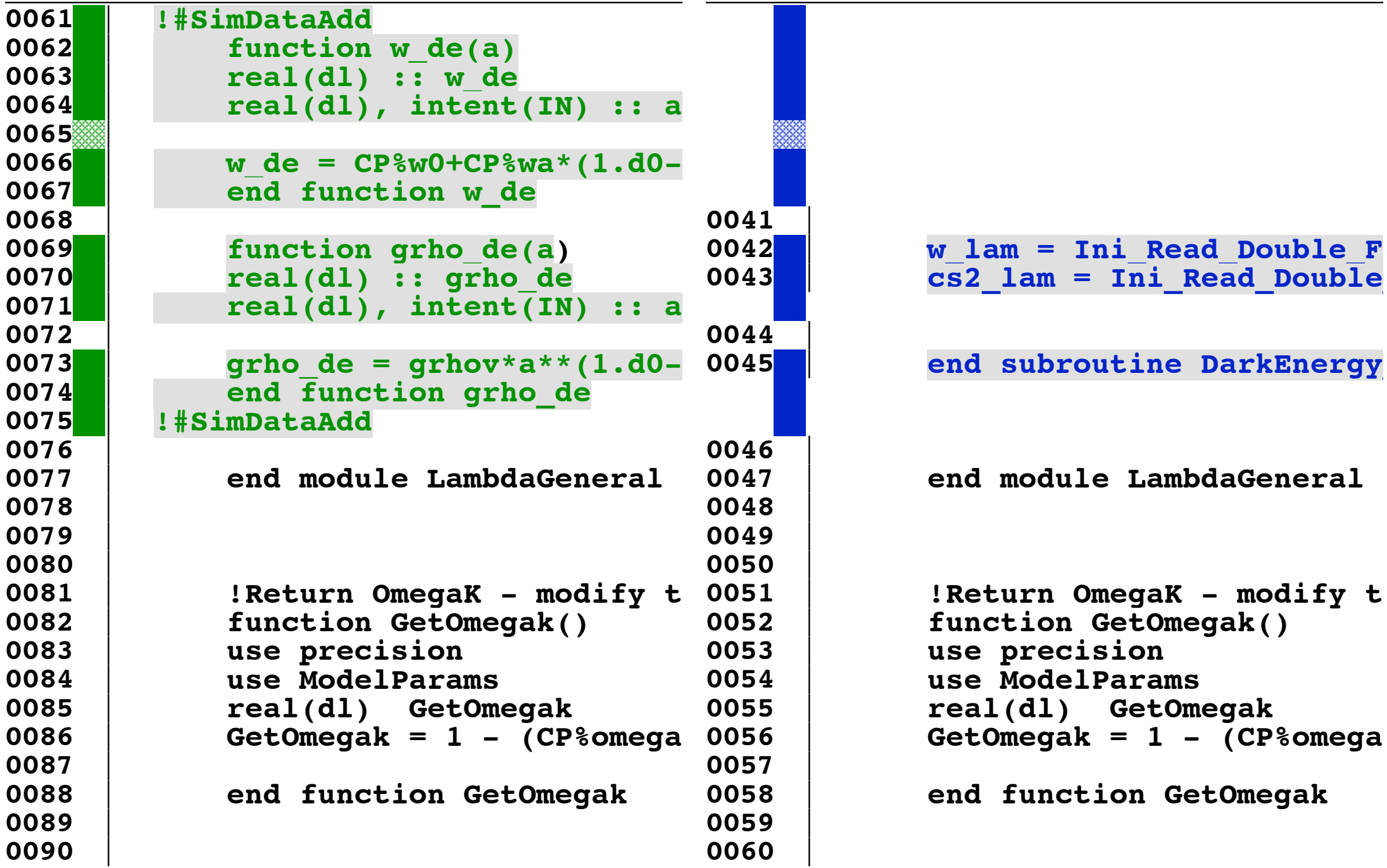
```
0001
0002
0003          ! Equations module for da
0004          ! allowing for perturbati
0005          ! by Antony Lewis (http:/
0006
0007          ! Dec 2003, fixed (fatal)
0008          ! Changes to tight coupli
0009          ! June 2004, fixed proble
0010          ! Generate vector modes o
0011          ! August 2004, fixed reio
0012          ! Nov 2004, change massiv
0013          ! Apr 2005, added DoLateR
0014          ! June 2006, added suppor
0015          ! Nov 2006, tweak to high
0016          ! June 2011, improved rad
0017          !             merged fderi
0018          !             optimized ne
0019          ! Feb 2013: fixed various
0020          ! Mar 2014: fixes for ten
0021
0022          module LambdaGeneral
0023          use precision
0024
0025      !#SimDataAdd
0026          use ModelParams
0027      !#SimDataAdd
0028
0029          implicit none
0030
```



```
0001          ! Equations module for da
0002          ! allowing for perturbati
0003          ! by Antony Lewis (http:/
0004
0005          ! Dec 2003, fixed (fatal)
0006          ! Changes to tight coupli
0007          ! June 2004, fixed proble
0008          ! Generate vector modes o
0009          ! August 2004, fixed reio
0010          ! Nov 2004, change massiv
0011          ! Apr 2005, added DoLateR
0012          ! June 2006, added suppor
0013          ! Nov 2006, tweak to high
0014          ! June 2011, improved rad
0015          !             merged fderi
0016          !             optimized ne
0017          ! Feb 2013: fixed various
0018          ! Mar 2014: fixes for ten
0019
0020          module LambdaGeneral
0021          use precision
0022          implicit none
0023
```


/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 31
/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 24
!#SimDataNoUse
! real(dl) :: w_lam = -1_
! real(dl) :: cs2_lam = 1_
!comoving sound speed. Al
!(otherwise assumed const
!de vazut ce e cu asta, in ve
real(dl), parameter :: wa
! logical :: w_perturb = .
!If you are tempted to se
! http://cosmocoffee.info
! http://cosmocoffee.info
!#SimDataNoUse
!#SimDataAdd
logical :: is_cosmologica
!#SimDataAdd
contains
!#SimDataNoUse
! subroutine DarkEnergy_Re
! use IniFile
! Type(TIniFile) :: Ini
! w_lam = Ini_Read_Double_
! cs2_lam = Ini_Read_Doubl
! end subroutine DarkEnerg
!#SimDataNoUse
real(dl) :: w_lam = -1_d
real(dl) :: cs2_lam = 1_d
!comoving sound speed. Al
!(otherwise assumed const
real(dl), parameter :: wa
logical :: w_perturb = .t
!If you are tempted to se
! http://cosmocoffee.info
! http://cosmocoffee.info
contains
subroutine DarkEnergy_Rea
use IniFile
Type(TIniFile) :: Ini


/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 61
/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 41
!#SimDataAdd
function w_de(a)
real(dl) :: w_de
real(dl), intent(IN) :: a
w_de = CP%w0+CP%wa*(1.d0-
end function w_de
function grho_de(a)
real(dl) :: grho_de
real(dl), intent(IN) :: a
grho_de = grhov*a**(1.d0-
end function grho_de
!#SimDataAdd
end module LambdaGeneral
!Return OmegaK - modify t
function GetOmegak()
use precision
use ModelParams
real(dl) GetOmegak
GetOmegak = 1 - (CP%omega
end function GetOmegak
w_lam = Ini_Read_Double_F
cs2_lam = Ini_Read_Double
end subroutine DarkEnergy
end module LambdaGeneral
!Return OmegaK - modify t
function GetOmegak()
use precision
use ModelParams
real(dl) GetOmegak
GetOmegak = 1 - (CP%omega
end function GetOmegak

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

```
0091        subroutine init_backgroun
0092        !This is only called once
0093        !It is called before firs
0094        !massive neutrinos are in
0095
0096
0097    !#SimDataAdd
0098        use LambdaGeneral
0099        is_cosmological_constant
0100    !#SimDataAdd
0101
0102        end  subroutine init_bac
0103
0104
0105        !Background evolution
0106        function dtauda(a)
0107        !get d tau / d a
0108        use precision
0109        use ModelParams
0110        use MassiveNu
0111        use LambdaGeneral
0112        implicit none
0113        real(dl) dtauda
0114        real(dl), intent(IN) :: a
0115        real(dl) rhonu,grhoa2, a2
0116        integer nu_i
0117
0118        a2=a**2
0119
0120        !  8*pi*G*rho*a**4.
```

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

```
0061        subroutine init_backgroun
0062        !This is only called once
0063        !It is called before firs
0064        !massive neutrinos are in
0065        end  subroutine init_back
0066
0067
0068        !Background evolution
0069        function dtauda(a)
0070        !get d tau / d a
0071        use precision
0072        use ModelParams
0073        use MassiveNu
0074        use LambdaGeneral
0075        implicit none
0076        real(dl) dtauda
0077        real(dl), intent(IN) :: a
0078        real(dl) rhonu,grhoa2, a2
0079        integer nu_i
0080
0081        a2=a**2
0082
0083        !  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         !ccccccccccccccccccccccc
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         !ccccccccccccccccccccccc
0103
0104         !Gauge-dependent perturba
```

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

```
0151
0152        module GaugeInterface
0153        use precision
0154        use ModelParams
0155        use MassiveNu
0156        use LambdaGeneral
0157        use Errors
0158        use Transfer
0159        implicit none
0160        public
0161
0162        !Description of this file
0163
0164    !#SimDataReplace
0165    !    character(LEN=*), parame
0166
0167        character(LEN=*), paramet
0168    !#SimDataReplace
0169
0170        integer, parameter :: bas
0171
0172        logical :: DoTensorNeutri
0173
0174        logical :: DoLateRadTrunc
0175        !if true, use smooth appr
0176        !small scales, saving evo
0177
0178        logical, parameter :: sec
0179
0180        real(dl) :: Magnetic = 0.
```

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

```
0105
0106        module GaugeInterface
0107        use precision
0108        use ModelParams
0109        use MassiveNu
0110        use LambdaGeneral
0111        use Errors
0112        use Transfer
0113        implicit none
0114        public
0115
0116        !Description of this file
0117        character(LEN=*), paramet
0118
0119        integer, parameter :: bas
0120
0121        logical :: DoTensorNeutri
0122
0123        logical :: DoLateRadTrunc
0124        !if true, use smooth appr
0125        !small scales, saving evo
0126
0127        logical, parameter :: sec
0128
0129        real(dl) :: Magnetic = 0.
```

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

```
0181        !Vector mode anisotropic
0182        real(dl) :: vec_sig0 = 1.
0183        !Vector mode shear
0184        integer, parameter :: max
0185        !Note higher values incre
0186
0187        !Supported scalar initial
0188        integer, parameter :: ini
0189        initial_iso_baryon=3,  in
0190        integer, parameter :: ini
0191
0192        type EvolutionVars
0193            real(dl) q, q2
0194            real(dl) k_buf,k2_buf
0195
0196            integer w_ix !Index o
0197            integer r_ix !Index o
0198            integer g_ix !Index o
0199
0200            integer q_ix !index i
0201            logical TransferOnly
0202
0203            !       nvar  - numbe
0204            integer nvar,nvart, n
0205
0206            !Max_l for the variou
0207            integer lmaxg,lmaxnr,
0208            integer lmaxnrt, lmax
0209            logical EvolveTensorM
0210            integer lmaxnrv, lmax
```

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

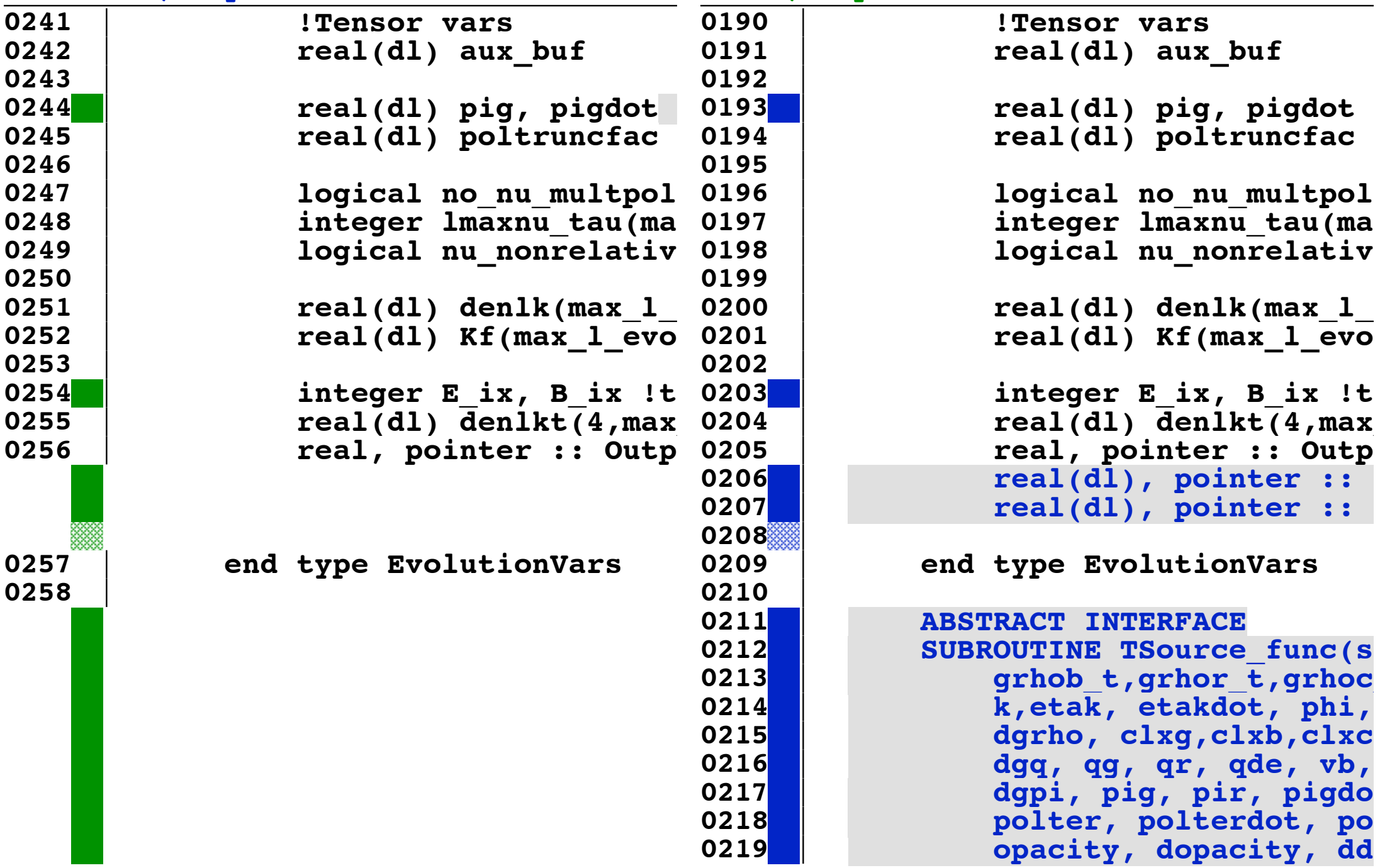
```
0130        !Vector mode anisotropic
0131        real(dl) :: vec_sig0 = 1.
0132        !Vector mode shear
0133        integer, parameter :: max
0134        !Note higher values incre
0135
0136        !Supported scalar initial
0137        integer, parameter :: ini
0138            initial_iso_baryon=3,
0139        integer, parameter :: ini
0140
0141        type EvolutionVars
0142            real(dl) q, q2
0143            real(dl) k_buf,k2_buf
0144
0145            integer w_ix !Index o
0146            integer r_ix !Index o
0147            integer g_ix !Index o
0148
0149            integer q_ix !index i
0150            logical TransferOnly
0151
0152            !       nvar  - numbe
0153            integer nvar,nvart, n
0154
0155            !Max_l for the variou
0156            integer lmaxg,lmaxnr,
0157            integer lmaxnrt, lmax
0158            logical EvolveTensorM
0159            integer lmaxnrv, lmax
```

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

```
0211 |
0212 |             integer polind  !inde
0213 |
0214 |             !array indices for ma
0215 |             integer nu_ix(max_nu)
0216 |             integer nq(max_nu), l
0217 |             logical has_nu_relati
0218 |
0219 |             !Initial values for m
0220 |             !to non-relativistic
0221 |             real(dl) G11(max_nu),
0222 |             !True when using non-
0223 |             logical MassiveNuAppr
0224 |             real(dl) MassiveNuApp
0225 |
0226 |             !True when truncating
0227 |             logical high_ktau_neu
0228 |
0229 |             !Massive neutrino sch
0230 |             integer NuMethod
0231 |
0232 |             !True when using tigh
0233 |             logical TightCoupling
0234 |             real(dl) TightSwitcho
0235 |
0236 |             !Numer of scalar equa
0237 |             integer ScalEqsToProp
0238 |             integer TensEqsToProp
0239 |             !beta > 1 for closed
0240 |             integer FirstZerolFor
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 160

```
0160 |
0161 |             integer polind  !inde
0162 |
0163 |             !array indices for ma
0164 |             integer nu_ix(max_nu)
0165 |             integer nq(max_nu), l
0166 |             logical has_nu_relati
0167 |
0168 |             !Initial values for m
0169 |             !to non-relativistic
0170 |             real(dl) G11(max_nu),
0171 |             !True when using non-
0172 |             logical MassiveNuAppr
0173 |             real(dl) MassiveNuApp
0174 |
0175 |             !True when truncating
0176 |             logical high_ktau_neu
0177 |
0178 |             !Massive neutrino sch
0179 |             integer NuMethod
0180 |
0181 |             !True when using tigh
0182 |             logical TightCoupling
0183 |             real(dl) TightSwitcho
0184 |
0185 |             !Numer of scalar equa
0186 |             integer ScalEqsToProp
0187 |             integer TensEqsToProp
0188 |             !beta > 1 for closed
0189 |             integer FirstZerolFor
```


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

```
0259          !precalculated arrays
0260          real(dl) polfac(max_l_evo
0261
0262          real(dl), parameter :: ep
0263          integer, parameter :: lma
0264
0265          real(dl) epsw
0266          real(dl) nu_tau_notmassle
0267          contains
0268
0269
0270          subroutine GaugeInterface
```

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

```
0220              tau0, tau_maxvis, Kf,
0221          real*8, intent(out) :: so
0222          real*8, intent(in) :: tau
0223              grhob_t,grhor_t,grhoc
0224              k,etak, etakdot, phi,
0225              dgrho, clxg,clxb,clxc
0226              dgq, qg, qr, qde, vb,
0227              dgpi, pig, pir, pigdo
0228              polter, polterdot, po
0229              opacity, dopacity, dd
0230              tau0, tau_maxvis
0231          REAL*8, intent(in) :: Kf(
0232          real*8, external :: f_K
0233          END SUBROUTINE TSource_fu
0234          END INTERFACE
0235
0236          procedure(TSource_func),
0237
0238          !precalculated arrays
0239          real(dl) polfac(max_l_evo
0240
0241          real(dl), parameter :: ep
0242          integer, parameter :: lma
0243
0244          real(dl) epsw
0245          real(dl) nu_tau_notmassle
0246          contains
0247
0248
0249          subroutine GaugeInterface
```

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

```
0271        type(EvolutionVars) EV
0272        real(dl) c(24),w(EV%nvar,
0273        integer ind
0274
0275        call dverk(EV,EV%ScalEqsT
0276        if (ind==-3) then
0277            call GlobalError('Dve
0278            //'requirement  with
0279            //'equal to hmin, whi
0280            //'--- but most likel
0281            //'compiling with bou
0282        end if
0283        end subroutine GaugeInter
0284
0285        function next_nu_nq(nq) r
0286        integer, intent(in) :: nq
0287        integer q, next_nq
0288
0289        if (nq==0) then
0290            next_nq=1
0291        else
0292            q = nu_q(nq)
0293            if (q>=10) then
0294                next_nq = nqmax
0295            else
0296                next_nq = nq+1
0297            end if
0298        end if
0299
0300        end function next_nu_nq
```

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

```
0250        type(EvolutionVars) EV
0251        real(dl) c(24),w(EV%nvar,
0252        integer ind
0253
0254        call dverk(EV,EV%ScalEqsT
0255        if (ind==-3) then
0256            call GlobalError('Dve
0257                //'requirement  w
0258                //'equal to hmin,
0259                //'--- but most l
0260                //'compiling with
0261        end if
0262        end subroutine GaugeInter
0263
0264        function next_nu_nq(nq) r
0265        integer, intent(in) :: nq
0266        integer q, next_nq
0267
0268        if (nq==0) then
0269            next_nq=1
0270        else
0271            q = nu_q(nq)
0272            if (q>=10) then
0273                next_nq = nqmax
0274            else
0275                next_nq = nq+1
0276            end if
0277        end if
0278
0279        end function next_nu_nq
```

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

```
0301 |
0302 |       recursive subroutine Gaug
0303 |       use ThermoData
0304 |       type(EvolutionVars) EV, E
0305 |       real(dl) c(24),w(EV%nvar,
0306 |       integer ind, nu_i
0307 |       real(dl) cs2, opacity, do
0308 |       real(dl) tau_switch_ktau,
0309 |       real(dl) tau_switch_no_nu
0310 |       real(dl) noSwitch, smallT
0311 |
0312 |       noSwitch= CP%tau0+1
0313 |       smallTime =  min(tau, 1/E
0314 |
0315 |       tau_switch_ktau = noSwitc
0316 |       tau_switch_no_nu_multpole
0317 |       tau_switch_no_phot_multpo
0318 |
0319 |       !Massive neutrino switche
0320 |       tau_switch_nu_massless =
0321 |       tau_switch_nu_nonrel = no
0322 |       tau_switch_nu_massive= no
0323 |
0324 |       !Evolve equations from ta
0325 |
0326 |       if (.not. EV%high_ktau_ne
0327 |           tau_switch_ktau=  max
0328 |       end if
0329 |
0330 |       if (CP%Num_Nu_massive /=
```

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

```
0280 |
0281 |       recursive subroutine Gaug
0282 |       use ThermoData
0283 |       type(EvolutionVars) EV, E
0284 |       real(dl) c(24),w(EV%nvar,
0285 |       integer ind, nu_i
0286 |       real(dl) cs2, opacity, do
0287 |       real(dl) tau_switch_ktau,
0288 |       real(dl) tau_switch_no_nu
0289 |       real(dl) noSwitch, smallT
0290 |
0291 |       noSwitch= CP%tau0+1
0292 |       smallTime =  min(tau, 1/E
0293 |
0294 |       tau_switch_ktau = noSwitc
0295 |       tau_switch_no_nu_multpole
0296 |       tau_switch_no_phot_multpo
0297 |
0298 |       !Massive neutrino switche
0299 |       tau_switch_nu_massless =
0300 |       tau_switch_nu_nonrel = no
0301 |       tau_switch_nu_massive= no
0302 |
0303 |       !Evolve equations from ta
0304 |
0305 |       if (.not. EV%high_ktau_ne
0306 |           tau_switch_ktau=  max
0307 |       end if
0308 |
0309 |       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

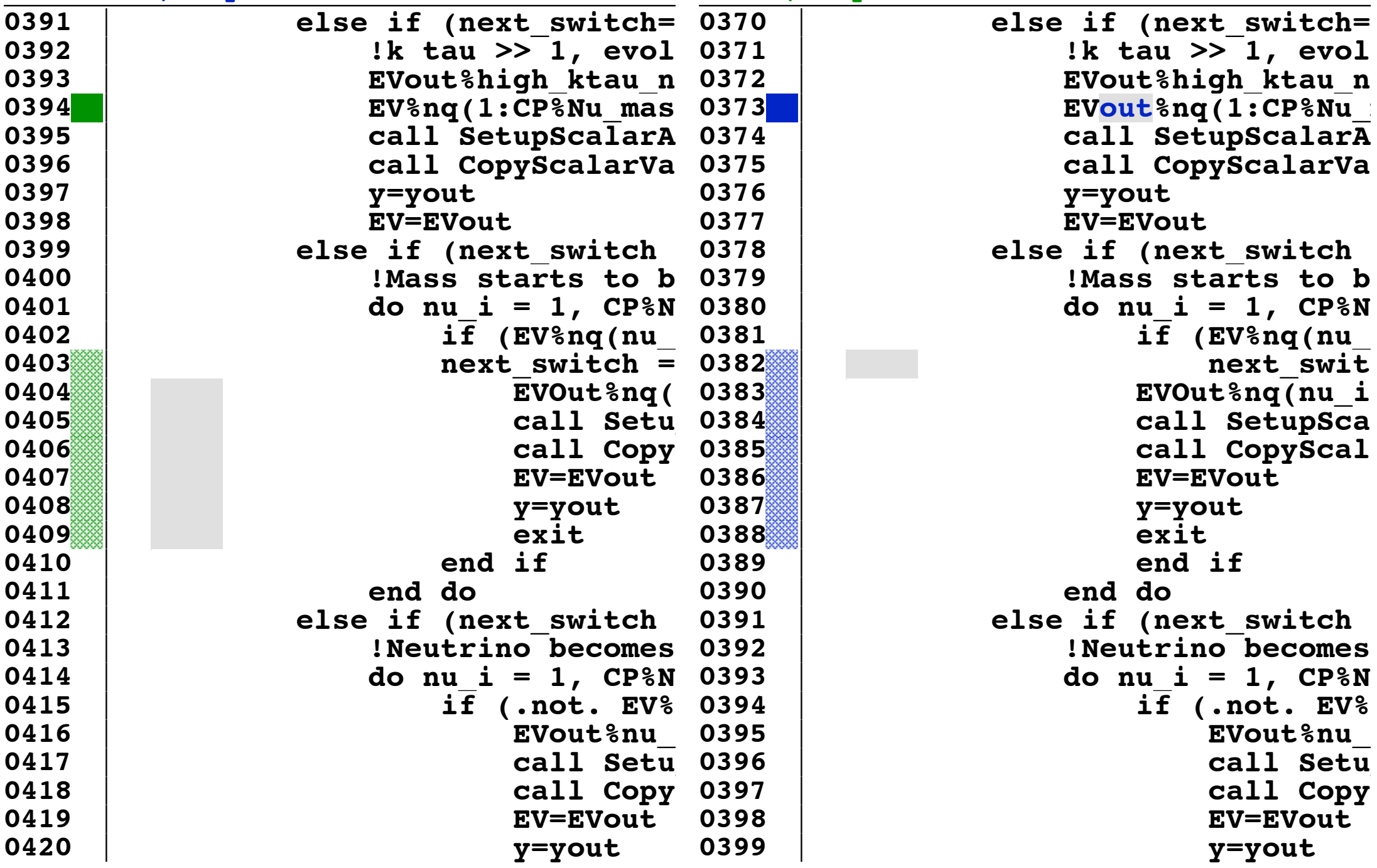
```
0310            do nu_i = 1, CP%Nu_ma
0311                if (EV%nq(nu_i) /
0312                    tau_switch_nu
0313                else if (.not. EV
0314                    tau_switch_nu
0315                else if (EV%NuMet
0316                    tau_switch_nu
0317                end if
0318            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

```
0361            if (next_switch == EV
0362                !TightCoupling
0363                EVout%TightCoupli
0364                EVout%TightSwitch
0365                call SetupScalarA
0366                call CopyScalarVa
0367                EV=EVout
0368                y=yout
0369                ind=1
0370                !Set up variables
0371                y(EV%g_ix+2) = EV
0372                call thermo(tau,c
0373
0374                if (second_order_
0375                    ! Francis-Yan
0376
0377                    y(EV%g_ix+3)
0378                    (3._dl/7._dl)
0379
0380                    y(EV%polind+2
0381                    (25._dl/16._d
0382                    EV%pig*(EV%k_
0383                    y(EV%polind+3
0384                    dopacity/opac
0385                    (1._dl+(5._dl
0386                else
0387                    y(EV%g_ix+3)
0388                    y(EV%polind+2
0389                    y(EV%polind+3
0390                end if
```

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

```
0340            if (next_switch == EV
0341                !TightCoupling
0342                EVout%TightCoupli
0343                EVout%TightSwitch
0344                call SetupScalarA
0345                call CopyScalarVa
0346                EV=EVout
0347                y=yout
0348                ind=1
0349                !Set up variables
0350                y(EV%g_ix+2) = EV
0351                call thermo(tau,c
0352
0353                if (second_order_
0354                    ! Francis-Yan
0355
0356                    y(EV%g_ix+3)
0357                        (3._dl/7.
0358
0359                    y(EV%polind+2
0360                        (25._dl/1
0361                        EV%pig*(E
0362                    y(EV%polind+3
0363                        dopacity/
0364                        (1._dl+(5
0365                else
0366                    y(EV%g_ix+3)
0367                    y(EV%polind+2
0368                    y(EV%polind+3
0369                end if
```


/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 391
/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 370

/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

```
0451          call GaugeInterface_E
0452          return
0453       end if
0454
0455       call GaugeInterface_ScalE
0456
0457       end subroutine GaugeInter
0458
0459       subroutine GaugeInterface
0460       use ThermoData
0461       type(EvolutionVars) EV, E
0462       real(dl) c(24),w(EV%nvart
0463       integer ind
0464       real(dl) opacity, cs2
0465
0466       if (EV%TensTightCoupling
0467          if (EV%TightSwitchoff
0468             call dverk(EV,EV%
0469          end if
0470          EVOut=EV
0471          EVOut%TensTightCoupli
0472          call SetupTensorArray
0473          call CopyTensorVariab
0474          Ev = EvOut
0475          y=yout
0476          call thermo(tau,cs2,o
0477          y(EV%g_ix+2)= 32._dl/
0478          y(EV%E_ix+2) = y(EV%g
0479       end if
0480
```

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

```
0430          call GaugeInterface_E
0431          return
0432       end if
0433
0434       call GaugeInterface_ScalE
0435
0436       end subroutine GaugeInter
0437
0438       subroutine GaugeInterface
0439       use ThermoData
0440       type(EvolutionVars) EV, E
0441       real(dl) c(24),w(EV%nvart
0442       integer ind
0443       real(dl) opacity, cs2
0444
0445       if (EV%TensTightCoupling
0446          if (EV%TightSwitchoff
0447             call dverk(EV,EV%
0448          end if
0449          EVOut=EV
0450          EVOut%TensTightCoupli
0451          call SetupTensorArray
0452          call CopyTensorVariab
0453          Ev = EvOut
0454          y=yout
0455          call thermo(tau,cs2,o
0456          y(EV%g_ix+2)= 32._dl/
0457          y(EV%E_ix+2) = y(EV%g
0458       end if
0459
```

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

```
0481        call dverk(EV,EV%TensEqsT
0482
0483
0484        end subroutine GaugeInter
0485
0486        function DeltaTimeMaxed(a
0487        real(dl) a1,a2,t
0488        real(dl), optional :: tol
0489        if (a1>1._dl) then
0490            t=0
0491        elseif (a2 > 1._dl) then
0492            t = DeltaTime(a1,1.01
0493        else
0494            t= DeltaTime(a1,a2, t
0495        end if
0496        end function DeltaTimeMax
0497
0498        subroutine GaugeInterface
0499        !Precompute various array
0500        integer j, nu_i
0501        real(dl) a_nonrel, a_mass
0502
0503        epsw = 100/CP%tau0
0504
0505        if (CP%WantScalars) then
0506            do j=2,max_l_evolve
0507                polfac(j)=real((j
0508            end do
0509        end if
0510
```

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

```
0460        call dverk(EV,EV%TensEqsT
0461
0462
0463        end subroutine GaugeInter
0464
0465        function DeltaTimeMaxed(a
0466        real(dl) a1,a2,t
0467        real(dl), optional :: tol
0468        if (a1>1._dl) then
0469            t=0
0470        elseif (a2 > 1._dl) then
0471            t = DeltaTime(a1,1.01
0472        else
0473            t= DeltaTime(a1,a2, t
0474        end if
0475        end function DeltaTimeMax
0476
0477        subroutine GaugeInterface
0478        !Precompute various array
0479        integer j, nu_i
0480        real(dl) a_nonrel, a_mass
0481
0482        epsw = 100/CP%tau0
0483
0484        if (CP%WantScalars) then
0485            do j=2,max_l_evolve
0486                polfac(j)=real((j
0487            end do
0488        end if
0489
```

/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

```
0541 |
0542 |
0543 |       subroutine SetupScalarArr
0544 |       !Set up array indices aft
0545 |       use MassiveNu
0546 |       !Set the numer of equatio
0547 |       type(EvolutionVars) EV
0548 |       integer, intent(out), opt
0549 |       integer neq, maxeq, nu_i
0550 |
0551 |       neq=basic_num_eqns
0552 |       maxeq=neq
0553 |       if (.not. EV%no_phot_mult
0554 |           !Photon multipoles
0555 |           EV%g_ix=basic_num_eqn
0556 |           if (EV%TightCoupling)
0557 |               neq=neq+2
0558 |           else
0559 |               neq = neq+ (EV%lm
0560 |               !Polarization mul
0561 |               EV%polind = neq -
0562 |               neq=neq + EV%lmax
0563 |           end if
0564 |       end if
0565 |       if (.not. EV%no_nu_multpo
0566 |           !Massless neutrino mu
0567 |           EV%r_ix= neq+1
0568 |           if (EV%high_ktau_neut
0569 |               neq=neq + 3
0570 |           else
```

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

```
0520 |
0521 |
0522 |       subroutine SetupScalarArr
0523 |       !Set up array indices aft
0524 |       use MassiveNu
0525 |       !Set the numer of equatio
0526 |       type(EvolutionVars) EV
0527 |       integer, intent(out), opt
0528 |       integer neq, maxeq, nu_i
0529 |
0530 |       neq=basic_num_eqns
0531 |       maxeq=neq
0532 |       if (.not. EV%no_phot_mult
0533 |           !Photon multipoles
0534 |           EV%g_ix=basic_num_eqn
0535 |           if (EV%TightCoupling)
0536 |               neq=neq+2
0537 |           else
0538 |               neq = neq+ (EV%lm
0539 |               !Polarization mul
0540 |               EV%polind = neq -
0541 |               neq=neq + EV%lmax
0542 |           end if
0543 |       end if
0544 |       if (.not. EV%no_nu_multpo
0545 |           !Massless neutrino mu
0546 |           EV%r_ix= neq+1
0547 |           if (EV%high_ktau_neut
0548 |               neq=neq + 3
0549 |           else
```

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

```
0571                neq=neq + (EV%lma
0572            end if
0573        end if
0574        maxeq = maxeq +  (EV%lmax
0575
0576    !#SimDataReplace
0577    !Dark energy
0578    !    if (w_lam /= -1 .and. w_
0579     if (.not. is_cosmological_co
0580            EV%w_ix = neq+1
0581            neq=neq+2
0582            maxeq=maxeq+2
0583        else
0584            EV%w_ix=0
0585        end if
0586    !#SimDataReplace
0587
0588        !Massive neutrinos
0589        if (CP%Num_Nu_massive /=
0590            EV%has_nu_relativisti
0591            if (EV%has_nu_relativ
0592                EV%lmaxnu_pert=EV
0593                EV%nu_pert_ix=neq
0594                neq = neq+ EV%lma
0595                maxeq=maxeq+ EV%l
0596            else
0597                EV%lmaxnu_pert=0
0598            end if
0599
0600            do nu_i=1, CP%Nu_Mass
```

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

```
0550                neq=neq + (EV%lma
0551            end if
0552        end if
0553        maxeq = maxeq +  (EV%lmax
0554
0555        !Dark energy
0556        if (w_lam /= -1 .and. w_P
0557            EV%w_ix = neq+1
0558            neq=neq+2
0559            maxeq=maxeq+2
0560        else
0561            EV%w_ix=0
0562        end if
0563
0564        !Massive neutrinos
0565        if (CP%Num_Nu_massive /=
0566            EV%has_nu_relativisti
0567            if (EV%has_nu_relativ
0568                EV%lmaxnu_pert=EV
0569                EV%nu_pert_ix=neq
0570                neq = neq+ EV%lma
0571                maxeq=maxeq+ EV%l
0572            else
0573                EV%lmaxnu_pert=0
0574            end if
0575
0576            do nu_i=1, CP%Nu_Mass
```

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

```
0601 |                 if (EV%high_ktau_
0602 |                     if (HighAccur
0603 |                         EV%lmaxnu
0604 |                     else
0605 |                         EV%lmaxnu
0606 |                     end if
0607 |                 else
0608 |                     EV%lmaxnu_tau
0609 |                     !!!Feb13tweak
0610 |                     if (EV%nu_non
0611 |                 end if
     |
0612 |                 EV%lmaxnu_tau(nu_
0613 |
0614 |                 EV%nu_ix(nu_i)=ne
0615 |                 if (EV%MassiveNuA
0616 |                     neq = neq+4
0617 |                 else
0618 |                     neq = neq+ EV
0619 |                 endif
0620 |                 maxeq = maxeq + n
0621 |             end do
0622 |         else
0623 |             EV%has_nu_relativisti
0624 |         end if
0625 |
0626 |         EV%ScalEqsToPropagate = n
0627 |         if (present(max_num_eqns)
0628 |             max_num_eqns=maxeq
0629 |         end if
```

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

```
0577 |                 if (EV%high_ktau_
0578 |                     EV%lmaxnu_tau
0579 |                     if (CP%Transf
     |
     |
     |
0580 |                 else
0581 |                     EV%lmaxnu_tau
0582 |                     !!!Feb13tweak
0583 |                     if (EV%nu_non
0584 |                 end if
0585 |                 if (nu_masses(nu_
0586 |                 EV%lmaxnu_tau(nu_
0587 |
0588 |                 EV%nu_ix(nu_i)=ne
0589 |                 if (EV%MassiveNuA
0590 |                     neq = neq+4
0591 |                 else
0592 |                     neq = neq+ EV
0593 |                 endif
0594 |                 maxeq = maxeq + n
0595 |             end do
0596 |         else
0597 |             EV%has_nu_relativisti
0598 |         end if
0599 |
0600 |         EV%ScalEqsToPropagate = n
0601 |         if (present(max_num_eqns)
0602 |             max_num_eqns=maxeq
0603 |         end if
```

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

```
0630
0631        end subroutine SetupScala
0632
0633        subroutine CopyScalarVari
0634        type(EvolutionVars) EV, E
0635        real(dl), intent(in) :: y
0636        real(dl), intent(out) ::
0637        integer lmax,i, nq
0638        integer nnueq,nu_i, ix_of
0639        real(dl) q, pert_scale
0640
0641        yout=0
0642        yout(1:basic_num_eqns) =
0643
0644    !#SimDataReplace
0645        if (.not. is_cosmological
0646    !    if (w_lam /= -1 .and. w_
0647            yout(EVout%w_ix)=y(EV
0648            yout(EVout%w_ix+1)=y(
0649        end if
0650    !#SimDataReplace
0651
0652        if (.not. EV%no_phot_mult
0653            if (EV%TightCoupling
0654                lmax=1
0655            else
0656                lmax = min(EV%lma
0657            end if
0658            yout(EVout%g_ix:EVout
0659            if (.not. EV%TightCou
```

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

```
0604
0605        end subroutine SetupScala
0606
0607        subroutine CopyScalarVari
0608        type(EvolutionVars) EV, E
0609        real(dl), intent(in) :: y
0610        real(dl), intent(out) ::
0611        integer lmax,i, nq
0612        integer nnueq,nu_i, ix_of
0613        real(dl) q, pert_scale
0614
0615        yout=0
0616        yout(1:basic_num_eqns) =
0617        if (w_lam /= -1 .and. w_P
0618            yout(EVout%w_ix)=y(EV
0619            yout(EVout%w_ix+1)=y(
0620        end if
0621
0622        if (.not. EV%no_phot_mult
0623            if (EV%TightCoupling
0624                lmax=1
0625            else
0626                lmax = min(EV%lma
0627            end if
0628            yout(EVout%g_ix:EVout
0629            if (.not. EV%TightCou
```

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

```
0660 |                 lmax = min(EV%lma
0661 |                 yout(EVout%polind
0662 |             end if
0663 |         end if
0664 |
0665 |         if (.not. EV%no_nu_multpo
0666 |             if (EV%high_ktau_neut
0667 |                 lmax=2
0668 |             else
0669 |                 lmax = min(EV%lma
0670 |             end if
0671 |             yout(EVout%r_ix:EVout
0672 |         end if
0673 |
0674 |         if (CP%Num_Nu_massive /=
0675 |             do nu_i=1,CP%Nu_mass_
0676 |                 ix_off=EV%nu_ix(n
0677 |                 ix_off2=EVOut%nu_
0678 |                 if (EV%MassiveNuA
0679 |                     nnueq=4
0680 |                     yout(ix_off2:
0681 |                 else if (.not. EV
0682 |                     lmax=min(EV%l
0683 |                     nq = min(EV%n
0684 |                     do i=1,nq
0685 |                         ind= ix_o
0686 |                         ind2=ix_o
0687 |                         yout(ind2
0688 |                     end do
0689 |                     do i=nq+1, EV
```

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

```
0630 |                 lmax = min(EV%lma
0631 |                 yout(EVout%polind
0632 |             end if
0633 |         end if
0634 |
0635 |         if (.not. EV%no_nu_multpo
0636 |             if (EV%high_ktau_neut
0637 |                 lmax=2
0638 |             else
0639 |                 lmax = min(EV%lma
0640 |             end if
0641 |             yout(EVout%r_ix:EVout
0642 |         end if
0643 |
0644 |         if (CP%Num_Nu_massive /=
0645 |             do nu_i=1,CP%Nu_mass_
0646 |                 ix_off=EV%nu_ix(n
0647 |                 ix_off2=EVOut%nu_
0648 |                 if (EV%MassiveNuA
0649 |                     nnueq=4
0650 |                     yout(ix_off2:
0651 |                 else if (.not. EV
0652 |                     lmax=min(EV%l
0653 |                     nq = min(EV%n
0654 |                     do i=1,nq
0655 |                         ind= ix_o
0656 |                         ind2=ix_o
0657 |                         yout(ind2
0658 |                     end do
0659 |                     do i=nq+1, EV
```

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

```
0690                                lmax = mi
0691                                ind2=ix_o
0692                                yout(ind2
0693
0694                                !Add lead
0695                                q=nu_q(i)
0696                                pert_scal
0697                                lmax = mi
0698                                yout(ind2
0699                                + y(EV%nu
0700                            end do
0701                        end if
0702                    end do
0703
0704                    if (EVOut%has_nu_rela
0705                        lmax = min(EVOut%
0706                        yout(EVout%nu_per
0707                    end if
0708                end if
0709
0710                end subroutine CopyScalar
0711
0712
0713                subroutine SetupTensorArr
0714                type(EvolutionVars) EV
0715                integer nu_i, neq
0716                integer, optional, intent
0717                neq=3
0718                EV%g_ix = neq-1 !EV%g_ix+
0719                if (.not. EV%TensTightCou
```

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

```
0660                                lmax = mi
0661                                ind2=ix_o
0662                                yout(ind2
0663
0664                                !Add lead
0665                                q=nu_q(i)
0666                                pert_scal
0667                                lmax = mi
0668                                yout(ind2
0669                                    + y(E
0670                            end do
0671                        end if
0672                    end do
0673
0674                    if (EVOut%has_nu_rela
0675                        lmax = min(EVOut%
0676                        yout(EVout%nu_per
0677                    end if
0678                end if
0679
0680                end subroutine CopyScalar
0681
0682
0683                subroutine SetupTensorArr
0684                type(EvolutionVars) EV
0685                integer nu_i, neq
0686                integer, optional, intent
0687                neq=3
0688                EV%g_ix = neq-1 !EV%g_ix+
0689                if (.not. EV%TensTightCou
```

```
0720            EV%E_ix = EV%g_ix + (
0721            EV%B_ix = EV%E_ix + (
0722            neq = neq+ (EV%lmaxt-
0723        end if
0724        if (present(maxeq)) then
0725            maxeq =3 + (EV%lmaxt-
0726        end if
0727        EV%r_ix = neq -1
0728        if (DoTensorNeutrinos) th
0729            neq = neq + EV%lmaxnr
0730            if (present(maxeq)) m
0731            if (CP%Num_Nu_massive
0732                do nu_i=1, CP%nu_
0733                    EV%EvolveTens
0734                    if (EV%Evolve
0735                        EV%nu_ix(
0736                        neq = neq
0737                        if (prese
0738                    end if
0739                end do
0740            end if
0741        end if
0742
0743        EV%TensEqsToPropagate = n
0744
0745        end  subroutine SetupTens
0746
0747        subroutine CopyTensorVari
0748        type(EvolutionVars) EV, E
0749        real(dl), intent(in) :: y
```



```
0690            EV%E_ix = EV%g_ix + (
0691            EV%B_ix = EV%E_ix + (
0692            neq = neq+ (EV%lmaxt-
0693        end if
0694        if (present(maxeq)) then
0695            maxeq =3 + (EV%lmaxt-
0696        end if
0697        EV%r_ix = neq -1
0698        if (DoTensorNeutrinos) th
0699            neq = neq + EV%lmaxnr
0700            if (present(maxeq)) m
0701            if (CP%Num_Nu_massive
0702                do nu_i=1, CP%nu_
0703                    EV%EvolveTens
0704                    if (EV%Evolve
0705                        EV%nu_ix(
0706                        neq = neq
0707                        if (prese
0708                    end if
0709                end do
0710            end if
0711        end if
0712
0713        EV%TensEqsToPropagate = n
0714
0715        end  subroutine SetupTens
0716
0717        subroutine CopyTensorVari
0718        type(EvolutionVars) EV, E
0719        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

```
0780        use MassiveNu
0781        !Set the numer of equatio
0782        type(EvolutionVars) EV
0783        real(dl) scal, max_nu_mas
0784        integer nu_i,q_rel,j
0785
0786        if (CP%Num_Nu_massive ==
0787            EV%lmaxnu=0
0788            max_nu_mass=0
0789        else
0790            max_nu_mass = maxval(
0791            do nu_i = 1, CP%Nu_ma
0792                !Start with momen
0793                q_rel=0
0794                do j=1, nqmax
0795                    !two differen
0796                    if (nu_q(j) >
0797                    q_rel = q_rel
0798                end do
0799
0800                if (q_rel>= nqmax
0801                    EV%nq(nu_i)=n
0802                else
0803                    EV%nq(nu_i)=q
0804                end if
0805                !q_rel = nint(nu_
0806                !EV%nq(nu_i)=max(
0807                EV%nu_nonrelativi
0808            end do
0809
```

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

```
0750        use MassiveNu
0751        !Set the numer of equatio
0752        type(EvolutionVars) EV
0753        real(dl) scal, max_nu_mas
0754        integer nu_i,q_rel,j
0755
0756        if (CP%Num_Nu_massive ==
0757            EV%lmaxnu=0
0758            max_nu_mass=0
0759        else
0760            max_nu_mass = maxval(
0761            do nu_i = 1, CP%Nu_ma
0762                !Start with momen
0763                q_rel=0
0764                do j=1, nqmax
0765                    !two differen
0766                    if (nu_q(j) >
0767                    q_rel = q_rel
0768                end do
0769
0770                if (q_rel>= nqmax
0771                    EV%nq(nu_i)=n
0772                else
0773                    EV%nq(nu_i)=q
0774                end if
0775                !q_rel = nint(nu_
0776                !EV%nq(nu_i)=max(
0777                EV%nu_nonrelativi
0778            end do
0779
```

/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(
```

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

```
0840 |             if (max_nu_mass>700 .
0841 |
0842 |             EV%lmaxgpol = EV%lmax
0843 |             if (.not.CP%AccurateP
0844 |
0845 |             if (EV%q < 0.05) then
0846 |                 !Large scales nee
0847 |                 scal  = 1
0848 |                 if (CP%AccuratePo
0849 |                 EV%lmaxgpol=max(3
0850 |                 EV%lmaxnr=max(3,n
0851 |                 EV%lmaxg=max(3,ni
0852 |                 if (CP%AccurateRe
0853 |                     EV%lmaxg=EV%l
0854 |                     EV%lmaxgpol=E
0855 |                 end if
0856 |             end if
0857 |
0858 |             if (EV%TransferOnly)
0859 |                 EV%lmaxgpol = min
0860 |                 EV%lmaxg = min(EV
0861 |             end if
0862 |             if (CP%Transfer%high_
0863 |                 if (HighAccuracyD
0864 |                     EV%lmaxnr=max
0865 |                 else
0866 |                     EV%lmaxnr=max
0867 |                 endif
0868 |                 if (EV%q > 0.04 .
0869 |                     EV%lmaxg=max(
```

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

```
0810 |             if (max_nu_mass>700 .
0811 |
0812 |             EV%lmaxgpol = EV%lmax
0813 |             if (.not.CP%AccurateP
0814 |
0815 |             if (EV%q < 0.05) then
0816 |                 !Large scales nee
0817 |                 scal  = 1
0818 |                 if (CP%AccuratePo
0819 |                 EV%lmaxgpol=max(3
0820 |                 EV%lmaxnr=max(3,n
0821 |                 EV%lmaxg=max(3,ni
0822 |                 if (CP%AccurateRe
0823 |                     EV%lmaxg=EV%l
0824 |                     EV%lmaxgpol=E
0825 |                 end if
0826 |             end if
0827 |
0828 |             if (EV%TransferOnly)
0829 |                 EV%lmaxgpol = min
0830 |                 EV%lmaxg = min(EV
0831 |             end if
0832 |             if (CP%Transfer%high_
0833 |                 if (HighAccuracyD
0834 |                     EV%lmaxnr=max
0835 |                 else
0836 |                     EV%lmaxnr=max
0837 |                 endif
0838 |                 if (EV%q > 0.04 .
0839 |                     EV%lmaxg=max(
```

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

```
0870                end if
0871             end if
0872
0873             if (CP%closed) then
0874                EV%lmaxnu=min(EV%
0875                EV%lmaxnr=min(EV%
0876                EV%lmaxg=min(EV%l
0877                EV%lmaxgpol=min(E
0878             end if
0879
0880             EV%poltruncfac=real(E
0881             EV%MaxlNeeded=max(EV%
0882             if (EV%MaxlNeeded > m
0883             call SetupScalarArray
0884             if (CP%closed) EV%nva
0885             EV%lmaxt=0
0886          else
0887             EV%nvar=0
0888          end if
0889
0890          if (CP%WantTensors) then
0891             EV%TensTightCoupling
0892             EV%lmaxt=max(3,nint(8
0893             EV%lmaxpolt = max(3,n
0894             ! if (EV%q < 1e-3) EV
0895             if (DoTensorNeutrinos
0896                EV%lmaxnrt=nint(6
0897                EV%lmaxnut=EV%lma
0898             else
0899                EV%lmaxnut=0
```

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

```
0840                end if
0841             end if
0842
0843             if (CP%closed) then
0844                EV%lmaxnu=min(EV%
0845                EV%lmaxnr=min(EV%
0846                EV%lmaxg=min(EV%l
0847                EV%lmaxgpol=min(E
0848             end if
0849
0850             EV%poltruncfac=real(E
0851             EV%MaxlNeeded=max(EV%
0852             if (EV%MaxlNeeded > m
0853             call SetupScalarArray
0854             if (CP%closed) EV%nva
0855             EV%lmaxt=0
0856          else
0857             EV%nvar=0
0858          end if
0859
0860          if (CP%WantTensors) then
0861             EV%TensTightCoupling
0862             EV%lmaxt=max(3,nint(8
0863             EV%lmaxpolt = max(3,n
0864             ! if (EV%q < 1e-3) EV
0865             if (DoTensorNeutrinos
0866                EV%lmaxnrt=nint(6
0867                EV%lmaxnut=EV%lma
0868             else
0869                EV%lmaxnut=0
```

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

```
0900                 EV%lmaxnrt=0
0901             end if
0902             if (CP%closed) then
0903                 EV%lmaxt=min(EV%F
0904                 EV%lmaxpolt=min(E
0905                 EV%lmaxnrt=min(EV
0906                 EV%lmaxnut=min(EV
0907             end if
0908             EV%MaxlNeededt=max(EV
0909             if (EV%MaxlNeededt >
0910             call SetupTensorArray
0911         else
0912             EV%nvart=0
0913         end if
0914
0915
0916         if (CP%WantVectors) then
0917             EV%lmaxv=max(10,nint(
0918             EV%lmaxpolv = max(5,n
0919
0920             EV%nvarv=(EV%lmaxv)+(
0921
0922             EV%lmaxnrv=nint(30*lA
0923
0924             EV%nvarv=EV%nvarv+EV%
0925             if (CP%Num_Nu_massive
0926                 stop 'massive neu
0927             end if
0928         else
0929             EV%nvarv=0
```

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

```
0870                 EV%lmaxnrt=0
0871             end if
0872             if (CP%closed) then
0873                 EV%lmaxt=min(EV%F
0874                 EV%lmaxpolt=min(E
0875                 EV%lmaxnrt=min(EV
0876                 EV%lmaxnut=min(EV
0877             end if
0878             EV%MaxlNeededt=max(EV
0879             if (EV%MaxlNeededt >
0880             call SetupTensorArray
0881         else
0882             EV%nvart=0
0883         end if
0884
0885
0886         if (CP%WantVectors) then
0887             EV%lmaxv=max(10,nint(
0888             EV%lmaxpolv = max(5,n
0889
0890             EV%nvarv=(EV%lmaxv)+(
0891
0892             EV%lmaxnrv=nint(30*lA
0893
0894             EV%nvarv=EV%nvarv+EV%
0895             if (CP%Num_Nu_massive
0896                 call MpiStop('mas
0897             end if
0898         else
0899             EV%nvarv=0
```

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

```
0930        end if
0931
0932        end subroutine GetNumEqns
0933
0934        !ccccccccccccccccccccccc
0935        subroutine SwitchToMassiv
0936        !When the neutrinos are n
0937        !energy-integrated hierar
0938        type(EvolutionVars) EV, E
0939        integer, intent(in) :: nu
0940
0941        real(dl) a,a2,pnu,clxnu,d
0942        real(dl) qnu
0943        real(dl) y(EV%nvar), yout
0944
0945        a=y(1)
0946        a2=a*a
0947        EVout=EV
0948        EVout%MassiveNuApprox(nu_
0949        call SetupScalarArrayIndi
0950        call CopyScalarVariableAr
0951
0952        !Get density and pressure
0953        call Nu_background(a*nu_m
0954
0955        !Integrate over q
0956        call Nu_Integrate_L012(EV
0957        !clxnu_here  = rhonu*clxn
0958        dpnu=dpnu/rhonu
0959        qnu=qnu/rhonu
```

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

```
0900        end if
0901
0902        end subroutine GetNumEqns
0903
0904        !ccccccccccccccccccccccc
0905        subroutine SwitchToMassiv
0906        !When the neutrinos are n
0907        !energy-integrated hierar
0908        type(EvolutionVars) EV, E
0909        integer, intent(in) :: nu
0910
0911        real(dl) a,a2,pnu,clxnu,d
0912        real(dl) qnu
0913        real(dl) y(EV%nvar), yout
0914
0915        a=y(1)
0916        a2=a*a
0917        EVout=EV
0918        EVout%MassiveNuApprox(nu_
0919        call SetupScalarArrayIndi
0920        call CopyScalarVariableAr
0921
0922        !Get density and pressure
0923        call Nu_background(a*nu_m
0924
0925        !Integrate over q
0926        call Nu_Integrate_L012(EV
0927        !clxnu_here  = rhonu*clxn
0928        dpnu=dpnu/rhonu
0929        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

```
0990 |
0991 |       integer nu_i
0992 |       real(dl) pinudot,grhormas
0993 |       real(dl) adotoa, grhonu_t
0994 |       real(dl) clxnu, qnu, pinu
0995 |       real(dl) dtauda
0996 |
0997 |       grhonu=0
0998 |       dgrhonu=0
0999 |       do nu_i = 1, CP%Nu_mass_e
1000 |           grhormass_t=grhormass
1001 |
1002 |           !Get density and pres
1003 |           call Nu_background(a*
1004 |
1005 |           if (EV%MassiveNuAppro
1006 |               clxnu=y(EV%nu_ix(
1007 |               !dpnu = y(EV%iq0+
1008 |               qnu=y(EV%nu_ix(nu
1009 |               pinu=y(EV%nu_ix(n
1010 |               pinudot=yprime(EV
1011 |           else
1012 |               !Integrate over q
1013 |               call Nu_Integrate
1014 |               !clxnu_here  = rh
1015 |               !dpnu=dpnu/rhonu
1016 |               qnu=qnu/rhonu
1017 |               clxnu = clxnu/rho
1018 |               pinu=pinu/rhonu
1019 |               adotoa = 1/(a*dta
```

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

```
0960 |
0961 |       integer nu_i
0962 |       real(dl) pinudot,grhormas
0963 |       real(dl) adotoa, grhonu_t
0964 |       real(dl) clxnu, qnu, pinu
0965 |       real(dl) dtauda
0966 |
0967 |       grhonu=0
0968 |       dgrhonu=0
0969 |       do nu_i = 1, CP%Nu_mass_e
0970 |           grhormass_t=grhormass
0971 |
0972 |           !Get density and pres
0973 |           call Nu_background(a*
0974 |
0975 |           if (EV%MassiveNuAppro
0976 |               clxnu=y(EV%nu_ix(
0977 |               !dpnu = y(EV%iq0+
0978 |               qnu=y(EV%nu_ix(nu
0979 |               pinu=y(EV%nu_ix(n
0980 |               pinudot=yprime(EV
0981 |           else
0982 |               !Integrate over q
0983 |               call Nu_Integrate
0984 |               !clxnu_here  = rh
0985 |               !dpnu=dpnu/rhonu
0986 |               qnu=qnu/rhonu
0987 |               clxnu = clxnu/rho
0988 |               pinu=pinu/rhonu
0989 |               adotoa = 1/(a*dta
```

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

```
1020                rhonudot = Nu_drh
1021
1022                call Nu_pinudot(E
1023                pinudot=pinudot/r
1024            endif
1025
1026            grhonu_t=grhormass_t*
1027            gpnu_t=grhormass_t*pn
1028
1029            grhonu = grhonu  + gr
1030            if (present(gpres)) g
1031
1032            dgrhonu= dgrhonu + gr
1033            if (present(dgq)) dgq
1034            if (present(dgpi)) dg
1035            if (present(gdpi_diff
1036            if (present(pidot_sum
1037        end do
1038        if (present(grho)) grho =
1039        if (present(dgrho)) dgrho
1040        if (present(clxnu_all)) c
1041
1042        end subroutine MassiveNuV
1043
1044        subroutine Nu_Integrate_L
1045        type(EvolutionVars) EV
1046        !  Compute the perturbati
1047        !  of one eigenstate of m
1048        !  density of one eigenst
1049        !  momentum.
```

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

```
0990                rhonudot = Nu_drh
0991
0992                call Nu_pinudot(E
0993                pinudot=pinudot/r
0994            endif
0995
0996            grhonu_t=grhormass_t*
0997            gpnu_t=grhormass_t*pn
0998
0999            grhonu = grhonu  + gr
1000            if (present(gpres)) g
1001
1002            dgrhonu= dgrhonu + gr
1003            if (present(dgq)) dgq
1004            if (present(dgpi)) dg
1005            if (present(dgpi_diff
1006            if (present(pidot_sum
1007        end do
1008        if (present(grho)) grho =
1009        if (present(dgrho)) dgrho
1010        if (present(clxnu_all)) c
1011
1012        end subroutine MassiveNuV
1013
1014        subroutine Nu_Integrate_L
1015        type(EvolutionVars) EV
1016        !  Compute the perturbati
1017        !  of one eigenstate of m
1018        !  density of one eigenst
1019        !  momentum.
```

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

```
1050        integer, intent(in) :: nu
1051        real(dl), intent(in) :: a
1052        real(dl), intent(OUT) ::
1053        real(dl), optional, inten
1054        real(dl) tmp, am, aq,v, p
1055        integer iq, ind
1056
1057        !  q is the comoving mome
1058
1059        drhonu=0
1060        fnu=0
1061        if (present(dpnu)) then
1062            dpnu=0
1063            pinu=0
1064        end if
1065        am=a*nu_masses(nu_i)
1066        ind=EV%nu_ix(nu_i)
1067        do iq=1,EV%nq(nu_i)
1068            aq=am/nu_q(iq)
1069            v=1._dl/sqrt(1._dl+aq
1070            drhonu=drhonu+ nu_int
1071            fnu=fnu+nu_int_kernel
1072            if (present(dpnu)) th
1073                dpnu=dpnu+  nu_in
1074                pinu=pinu+ nu_int
1075            end if
1076            ind=ind+EV%lmaxnu_tau
1077        end do
1078        ind = EV%nu_pert_ix
1079        do iq=EV%nq(nu_i)+1,nqmax
```

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

```
1020        integer, intent(in) :: nu
1021        real(dl), intent(in) :: a
1022        real(dl), intent(OUT) ::
1023        real(dl), optional, inten
1024        real(dl) tmp, am, aq,v, p
1025        integer iq, ind
1026
1027        !  q is the comoving mome
1028
1029        drhonu=0
1030        fnu=0
1031        if (present(dpnu)) then
1032            dpnu=0
1033            pinu=0
1034        end if
1035        am=a*nu_masses(nu_i)
1036        ind=EV%nu_ix(nu_i)
1037        do iq=1,EV%nq(nu_i)
1038            aq=am/nu_q(iq)
1039            v=1._dl/sqrt(1._dl+aq
1040            drhonu=drhonu+ nu_int
1041            fnu=fnu+nu_int_kernel
1042            if (present(dpnu)) th
1043                dpnu=dpnu+  nu_in
1044                pinu=pinu+ nu_int
1045            end if
1046            ind=ind+EV%lmaxnu_tau
1047        end do
1048        ind = EV%nu_pert_ix
1049        do iq=EV%nq(nu_i)+1,nqmax
```

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

```
1080 |             !Get the rest from pe
1081 |             aq=am/nu_q(iq)
1082 |             v=1._dl/sqrt(1._dl+aq
1083 |             pert_scale=(nu_masses
1084 |             tmp = nu_int_kernel(i
1085 |             drhonu=drhonu+ tmp/v
1086 |             fnu=fnu+nu_int_kernel
1087 |             if (present(dpnu)) th
1088 |                 dpnu=dpnu+ tmp*v
1089 |                 pinu = pinu+ nu_i
1090 |             end if
1091 |         end do
1092 |
1093 |         if (present(dpnu)) then
1094 |             dpnu = dpnu/3
1095 |         end if
1096 |
1097 |         end subroutine Nu_Integra
1098 |
1099 |         subroutine Nu_pinudot(EV,
1100 |         type(EvolutionVars) EV
1101 |         integer, intent(in) :: nu
1102 |         real(dl), intent(in) :: a
1103 |
1104 |         !  Compute the time deriv
1105 |         !  and the shear perturba
1106 |         real(dl) pinudot
1107 |         real(dl) aq,q,v,aqdot,vdo
1108 |         real(dl) psi2,psi2dot
1109 |         real(dl) am, pert_scale
```

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

```
1050 |             !Get the rest from pe
1051 |             aq=am/nu_q(iq)
1052 |             v=1._dl/sqrt(1._dl+aq
1053 |             pert_scale=(nu_masses
1054 |             tmp = nu_int_kernel(i
1055 |             drhonu=drhonu+ tmp/v
1056 |             fnu=fnu+nu_int_kernel
1057 |             if (present(dpnu)) th
1058 |                 dpnu=dpnu+ tmp*v
1059 |                 pinu = pinu+ nu_i
1060 |             end if
1061 |         end do
1062 |
1063 |         if (present(dpnu)) then
1064 |             dpnu = dpnu/3
1065 |         end if
1066 |
1067 |         end subroutine Nu_Integra
1068 |
1069 |         subroutine Nu_pinudot(EV,
1070 |         type(EvolutionVars) EV
1071 |         integer, intent(in) :: nu
1072 |         real(dl), intent(in) :: a
1073 |
1074 |         !  Compute the time deriv
1075 |         !  and the shear perturba
1076 |         real(dl) pinudot
1077 |         real(dl) aq,q,v,aqdot,vdo
1078 |         real(dl) psi2,psi2dot
1079 |         real(dl) am, pert_scale
```

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

```
1110        integer iq,ind
1111
1112        !  q is the comoving mome
1113        pinudot=0._dl
1114        ind=EV%nu_ix(nu_i)+2
1115        am=a*nu_masses(nu_i)
1116        do iq=1,EV%nq(nu_i)
1117            q=nu_q(iq)
1118            aq=am/q
1119            aqdot=aq*adotoa
1120            v=1._dl/sqrt(1._dl+aq
1121            vdot=-aq*aqdot/(1._dl
1122            pinudot=pinudot+nu_in
1123            ind=ind+EV%lmaxnu_tau
1124        end do
1125        ind = EV%nu_pert_ix+2
1126        do iq=EV%nq(nu_i)+1,nqmax
1127            q=nu_q(iq)
1128            aq=am/q
1129            aqdot=aq*adotoa
1130            pert_scale=(nu_masses
1131            v=1._dl/sqrt(1._dl+aq
1132            vdot=-aq*aqdot/(1._dl
1133            psi2dot=ydot(EV%r_ix+
1134            psi2=y(EV%r_ix+2)  +
1135            pinudot=pinudot+nu_in
1136        end do
1137
1138        end subroutine Nu_pinudot
1139
```

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

```
1080        integer iq,ind
1081
1082        !  q is the comoving mome
1083        pinudot=0._dl
1084        ind=EV%nu_ix(nu_i)+2
1085        am=a*nu_masses(nu_i)
1086        do iq=1,EV%nq(nu_i)
1087            q=nu_q(iq)
1088            aq=am/q
1089            aqdot=aq*adotoa
1090            v=1._dl/sqrt(1._dl+aq
1091            vdot=-aq*aqdot/(1._dl
1092            pinudot=pinudot+nu_in
1093            ind=ind+EV%lmaxnu_tau
1094        end do
1095        ind = EV%nu_pert_ix+2
1096        do iq=EV%nq(nu_i)+1,nqmax
1097            q=nu_q(iq)
1098            aq=am/q
1099            aqdot=aq*adotoa
1100            pert_scale=(nu_masses
1101            v=1._dl/sqrt(1._dl+aq
1102            vdot=-aq*aqdot/(1._dl
1103            psi2dot=ydot(EV%r_ix+
1104            psi2=y(EV%r_ix+2)  +
1105            pinudot=pinudot+nu_in
1106        end do
1107
1108        end subroutine Nu_pinudot
1109
```

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

```
1140        !ccccccccccccccccccccccc
1141        function Nu_pi(EV, y, a,
1142        type(EvolutionVars) EV
1143        integer, intent(in) :: nu
1144        real(dl), intent(in) :: a
1145        real(dl) :: am
1146        real(dl) pinu,q,aq,v
1147        integer iq, ind
1148
1149        if (EV%nq(nu_i)/=nqmax) s
1150        pinu=0
1151        ind=EV%nu_ix(nu_i)+2
1152        am=a*nu_masses(nu_i)
1153        do iq=1, EV%nq(nu_i)
1154            q=nu_q(iq)
1155            aq=am/q
1156            v=1._dl/sqrt(1._dl+aq
1157            pinu=pinu+nu_int_kern
1158            ind =ind+EV%lmaxnut+1
1159        end do
1160
1161        end function Nu_pi
1162
1163        !ccccccccccccccccccccccc
1164        subroutine Nu_Intvsq(EV,y
1165        type(EvolutionVars) EV
1166        integer, intent(in) :: nu
1167        real(dl), intent(in) :: a
1168        real(dl), intent(OUT) ::
1169
```

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

```
1110        !ccccccccccccccccccccccc
1111        function Nu_pi(EV, y, a,
1112        type(EvolutionVars) EV
1113        integer, intent(in) :: nu
1114        real(dl), intent(in) :: a
1115        real(dl) :: am
1116        real(dl) pinu,q,aq,v
1117        integer iq, ind
1118
1119        if (EV%nq(nu_i)/=nqmax) c
1120        pinu=0
1121        ind=EV%nu_ix(nu_i)+2
1122        am=a*nu_masses(nu_i)
1123        do iq=1, EV%nq(nu_i)
1124            q=nu_q(iq)
1125            aq=am/q
1126            v=1._dl/sqrt(1._dl+aq
1127            pinu=pinu+nu_int_kern
1128            ind =ind+EV%lmaxnut+1
1129        end do
1130
1131        end function Nu_pi
1132
1133        !ccccccccccccccccccccccc
1134        subroutine Nu_Intvsq(EV,y
1135        type(EvolutionVars) EV
1136        integer, intent(in) :: nu
1137        real(dl), intent(in) :: a
1138        real(dl), intent(OUT) ::
1139
```

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

```
1170        !  Compute the third orde
1171        !by integrating over mome
1172        real(dl) aq,q,v, am
1173        integer iq, ind
1174
1175        !  q is the comoving mome
1176        am=a*nu_masses(nu_i)
1177        ind=EV%nu_ix(nu_i)
1178        G11=0._dl
1179        G30=0._dl
1180        if (EV%nq(nu_i)/=nqmax) s
1181        do iq=1, EV%nq(nu_i)
1182            q=nu_q(iq)
1183            aq=am/q
1184            v=1._dl/sqrt(1._dl+aq
1185            G11=G11+nu_int_kernel
1186            if (EV%lmaxnu_tau(nu_
1187                G30=G30+nu_int_ke
1188            end if
1189            ind = ind+EV%lmaxnu_t
1190        end do
1191
1192        end subroutine Nu_Intvsq
1193
1194
1195        subroutine MassiveNuVars(
1196        implicit none
1197        type(EvolutionVars) EV
1198        real(dl) :: y(EV%nvar), a
1199        real(dl), intent(out), op
```

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

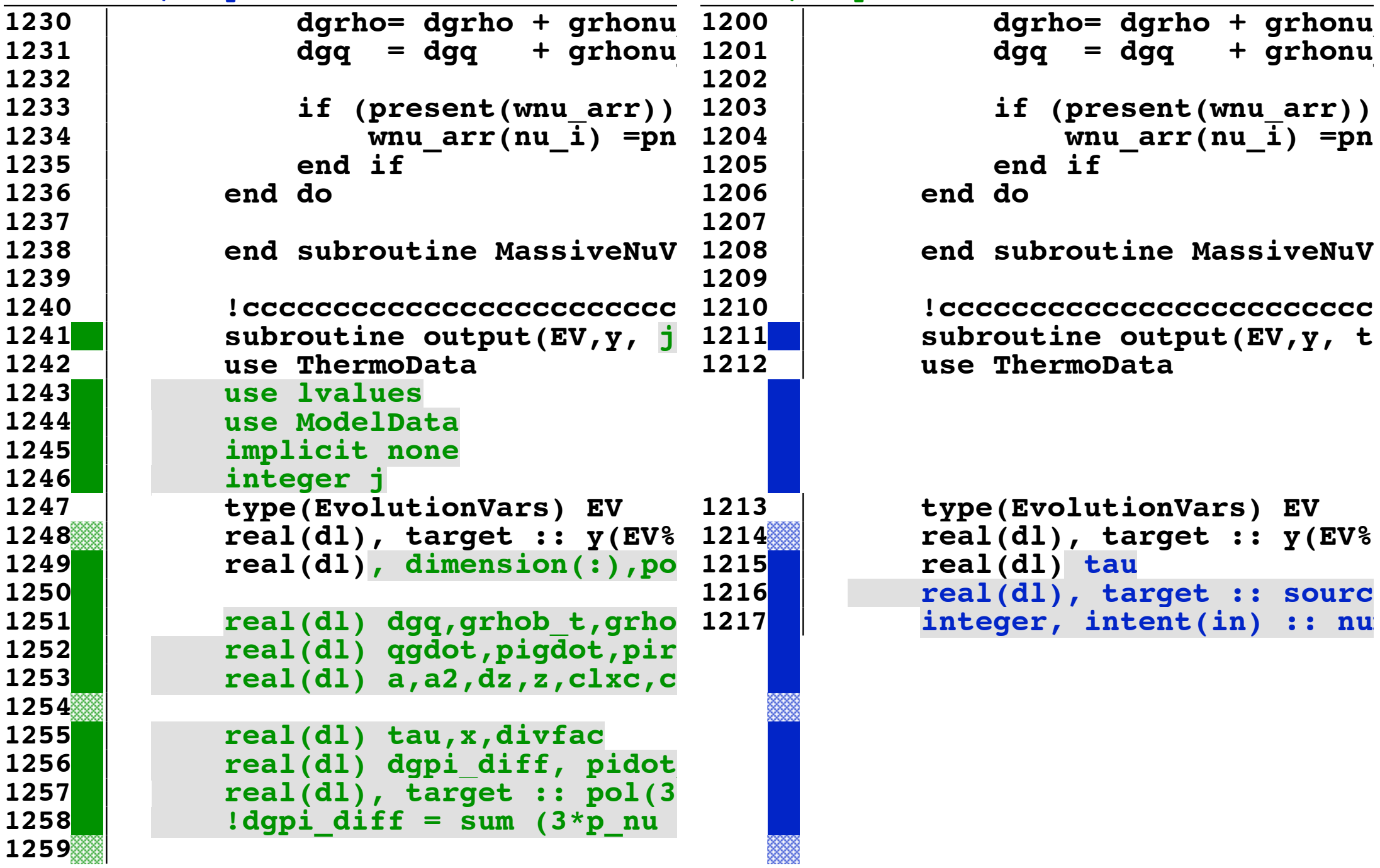
```
1140        !  Compute the third orde
1141        !by integrating over mome
1142        real(dl) aq,q,v, am
1143        integer iq, ind
1144
1145        !  q is the comoving mome
1146        am=a*nu_masses(nu_i)
1147        ind=EV%nu_ix(nu_i)
1148        G11=0._dl
1149        G30=0._dl
1150        if (EV%nq(nu_i)/=nqmax) c
1151        do iq=1, EV%nq(nu_i)
1152            q=nu_q(iq)
1153            aq=am/q
1154            v=1._dl/sqrt(1._dl+aq
1155            G11=G11+nu_int_kernel
1156            if (EV%lmaxnu_tau(nu_
1157                G30=G30+nu_int_ke
1158            end if
1159            ind = ind+EV%lmaxnu_t
1160        end do
1161
1162        end subroutine Nu_Intvsq
1163
1164
1165        subroutine MassiveNuVars(
1166        implicit none
1167        type(EvolutionVars) EV
1168        real(dl) :: y(EV%nvar), a
1169        real(dl), intent(out), op
```

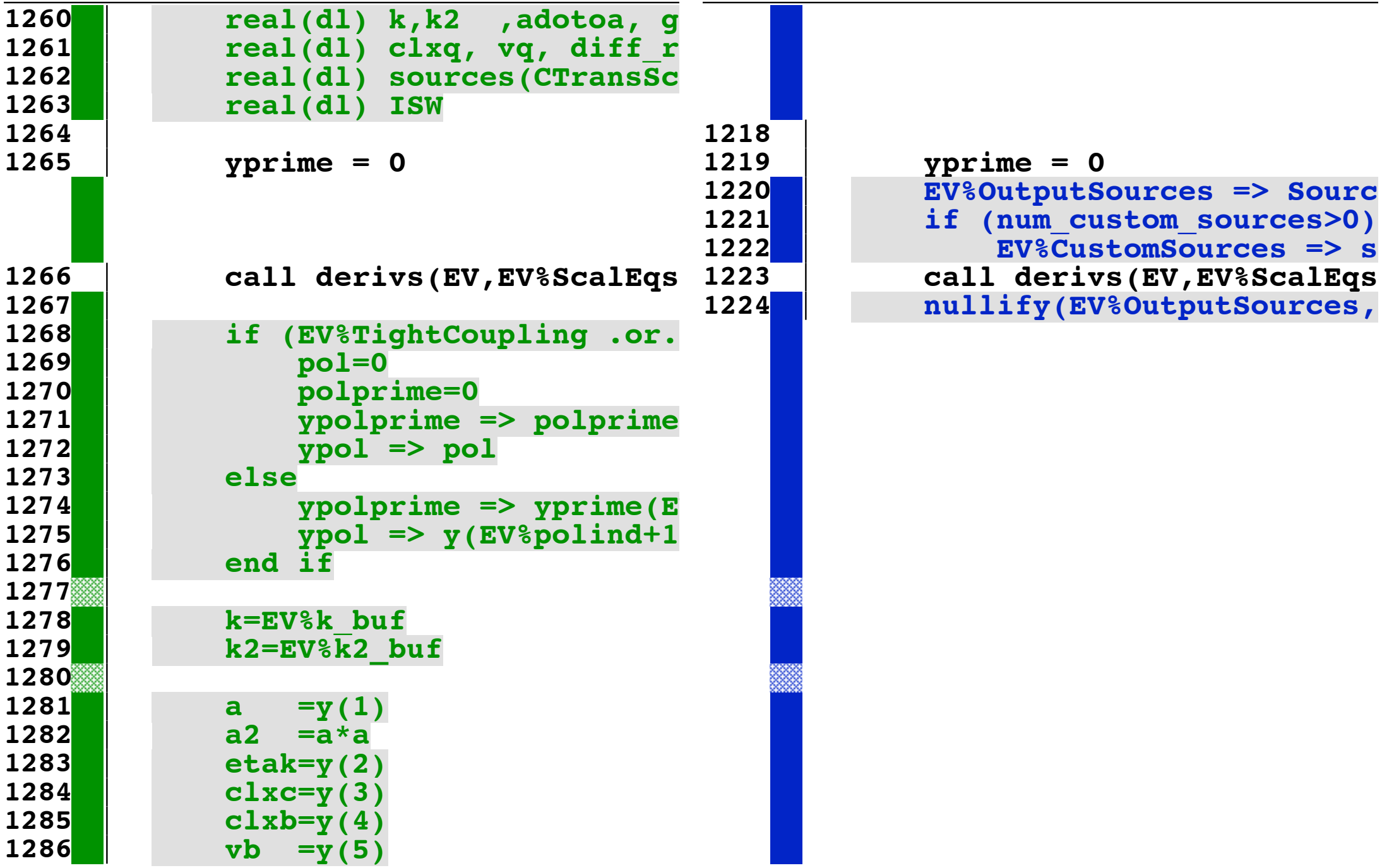
/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1200

```
1200          !grho = a^2 kappa rho
1201          !gpres = a^2 kappa p
1202          !dgrho = a^2 kappa \delta
1203          !dgp =  a^2 kappa \delta
1204          !dgq = a^2 kappa q (heat
1205          integer nu_i
1206          real(dl) grhormass_t, rho
1207
1208          do nu_i = 1, CP%Nu_mass_e
1209              grhormass_t=grhormass
1210
1211              !Get density and pres
1212              call Nu_background(a*
1213
1214              if (EV%MassiveNuAppro
1215                  clxnu=y(EV%nu_ix(
1216                  qnu=y(EV%nu_ix(nu
1217              else
1218                  !Integrate over q
1219                  call Nu_Integrate
1220                  !clxnu_here  = rh
1221                  qnu=qnu/rhonu
1222                  clxnu = clxnu/rho
1223              endif
1224
1225              grhonu_t=grhormass_t*
1226              gpnu_t=grhormass_t*pn
1227
1228              grho = grho  + grhonu
1229              gpres= gpres + gpnu_t
```

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

```
1170          !grho = a^2 kappa rho
1171          !gpres = a^2 kappa p
1172          !dgrho = a^2 kappa \delta
1173          !dgp =  a^2 kappa \delta
1174          !dgq = a^2 kappa q (heat
1175          integer nu_i
1176          real(dl) grhormass_t, rho
1177
1178          do nu_i = 1, CP%Nu_mass_e
1179              grhormass_t=grhormass
1180
1181              !Get density and pres
1182              call Nu_background(a*
1183
1184              if (EV%MassiveNuAppro
1185                  clxnu=y(EV%nu_ix(
1186                  qnu=y(EV%nu_ix(nu
1187              else
1188                  !Integrate over q
1189                  call Nu_Integrate
1190                  !clxnu_here  = rh
1191                  qnu=qnu/rhonu
1192                  clxnu = clxnu/rho
1193              endif
1194
1195              grhonu_t=grhormass_t*
1196              gpnu_t=grhormass_t*pn
1197
1198              grho = grho  + grhonu
1199              gpres= gpres + gpnu_t
```


/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1230
/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1200
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 !ccccccccccccccccccccccc
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
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 !ccccccccccccccccccccccc
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
1260 real(dl) k,k2 ,adotoa, g
1261 real(dl) clxq, vq, diff_r
1262 real(dl) sources(CTransSc
1263 real(dl) ISW
1264
1265 yprime = 0
1266 call derivs(EV,EV%ScalEqs
1267
1268 if (EV%TightCoupling .or.
1269 pol=0
1270 polprime=0
1271 ypolprime => polprime
1272 ypol => pol
1273 else
1274 ypolprime => yprime(E
1275 ypol => y(EV%polind+1
1276 end if
1277
1278 k=EV%k_buf
1279 k2=EV%k2_buf
1280
1281 a =y(1)
1282 a2 =a*a
1283 etak=y(2)
1284 clxc=y(3)
1285 clxb=y(4)
1286 vb =y(5)
/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1218
1218
1219 yprime = 0
1220 EV%OutputSources => Sourc
1221 if (num_custom_sources>0)
1222 EV%CustomSources => s
1223 call derivs(EV,EV%ScalEqs
1224 nullify(EV%OutputSources,

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

```
1287      vbdot =yprime(5)
1288
1289      !  Compute expansion rate
1290
1291      grhob_t=grhob/a
1292      grhoc_t=grhoc/a
1293      grhor_t=grhornomass/a2
1294      grhog_t=grhog/a2
1295
1296    !#SimDataReplace
1297    !     grhov_t=grhov*a**(-1-3*w
1298
1299          grhov_t=grho_de(a)/a2
1300    !#SimDataReplace
1301
1302      grho=grhob_t+grhoc_t+grho
1303
1304    !#SimDataReplace
1305    !     gpres=(grhog_t+grhor_t)/
1306          gpres=(grhog_t+grhor_t)/
1307    !#SimDataReplace
1308
1309      !  8*pi*a*a*SUM[rho_i*clx
1310      dgrho=grhob_t*clxb+grhoc_
1311
1312      !  8*pi*a*a*SUM[(rho_i+p_
1313      dgq=grhob_t*vb
1314
1315      dgpi=0
1316      dgpi_diff = 0
```

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

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

```
1317        pidot_sum = 0
1318
1319        if (CP%Num_Nu_Massive /=
1320            call MassiveNuVarsOut
1321        end if
1322
1323    !#SimDataReplace
1324    !     if (w_lam /= -1 .and. w_
1325          if (.not. is_cosmologica
1326             clxq=y(EV%w_ix)
1327             vq=y(EV%w_ix+1)
1328             dgrho=dgrho + clxq*gr
1329    !        dgq = dgq + vq*grhov_
1330             dgq = dgq + vq*grhov_
1331        end if
1332    !#SimDataReplace
1333        adotoa=sqrt((grho+grhok)/
1334
1335        if (EV%no_nu_multpoles) t
1336            z=(0.5_dl*dgrho/k + e
1337            dz= -adotoa*z - 0.5_d
1338            clxr=-4*dz/k
1339            qr=-4._dl/3*z
1340            pir=0
1341            pirdot=0
1342        else
1343            clxr=y(EV%r_ix)
1344            qr  =y(EV%r_ix+1)
1345            pir =y(EV%r_ix+2)
1346            pirdot=yprime(EV%r_ix
```

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

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

```
1347        end if
1348
1349        if (EV%no_phot_multpoles)
1350            z=(0.5_dl*dgrho/k + e
1351            dz= -adotoa*z - 0.5_d
1352            clxg=-4*dz/k -4/k*opa
1353            qg=-4._dl/3*z
1354            pig=0
1355            pigdot=0
1356            octg=0
1357            octgprime=0
1358            qgdot = -4*dz/3
1359        else
1360            if (EV%TightCoupling)
1361                pig = EV%pig
1362                !pigdot=EV%pigdot
1363                if (second_order_
1364                    octg = (3._dl
1365                    ypol(2) = EV%
1366                    ypol(3) = (3.
1367                else
1368                    ypol(2) = EV%
1369                    octg=0
1370                end if
1371                octgprime=0
1372            else
1373                pig =y(EV%g_ix+2)
1374                pigdot=yprime(EV%
1375                octg=y(EV%g_ix+3)
1376                octgprime=yprime(
```

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

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1377

```
1377            end if
1378            clxg=y(EV%g_ix)
1379            qg  =y(EV%g_ix+1)
1380            qgdot =yprime(EV%g_ix
1381        end if
1382
1383        dgrho = dgrho + grhog_t*c
1384        dgq   = dgq   + grhog_t*q
1385        dgpi  = dgpi  + grhor_t*p
1386
1387
1388        !  Get sigma (shear) and
1389        !  have to get z from eta
1390        z=(0.5_dl*dgrho/k + etak)
1391        sigma=(z+1.5_dl*dgq/k2)/E
1392
1393        polter = 0.1_dl*pig+9._dl
1394
1395        if (CP%flat) then
1396            x=k*(CP%tau0-tau)
1397            divfac=x*x
1398        else
1399            x=(CP%tau0-tau)/CP%r
1400            divfac=(CP%r*rofChi(x
1401        end if
1402
1403
1404        if (EV%TightCoupling) the
1405            if (second_order_tigh
1406                pigdot = EV%pigdo
```

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

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1407

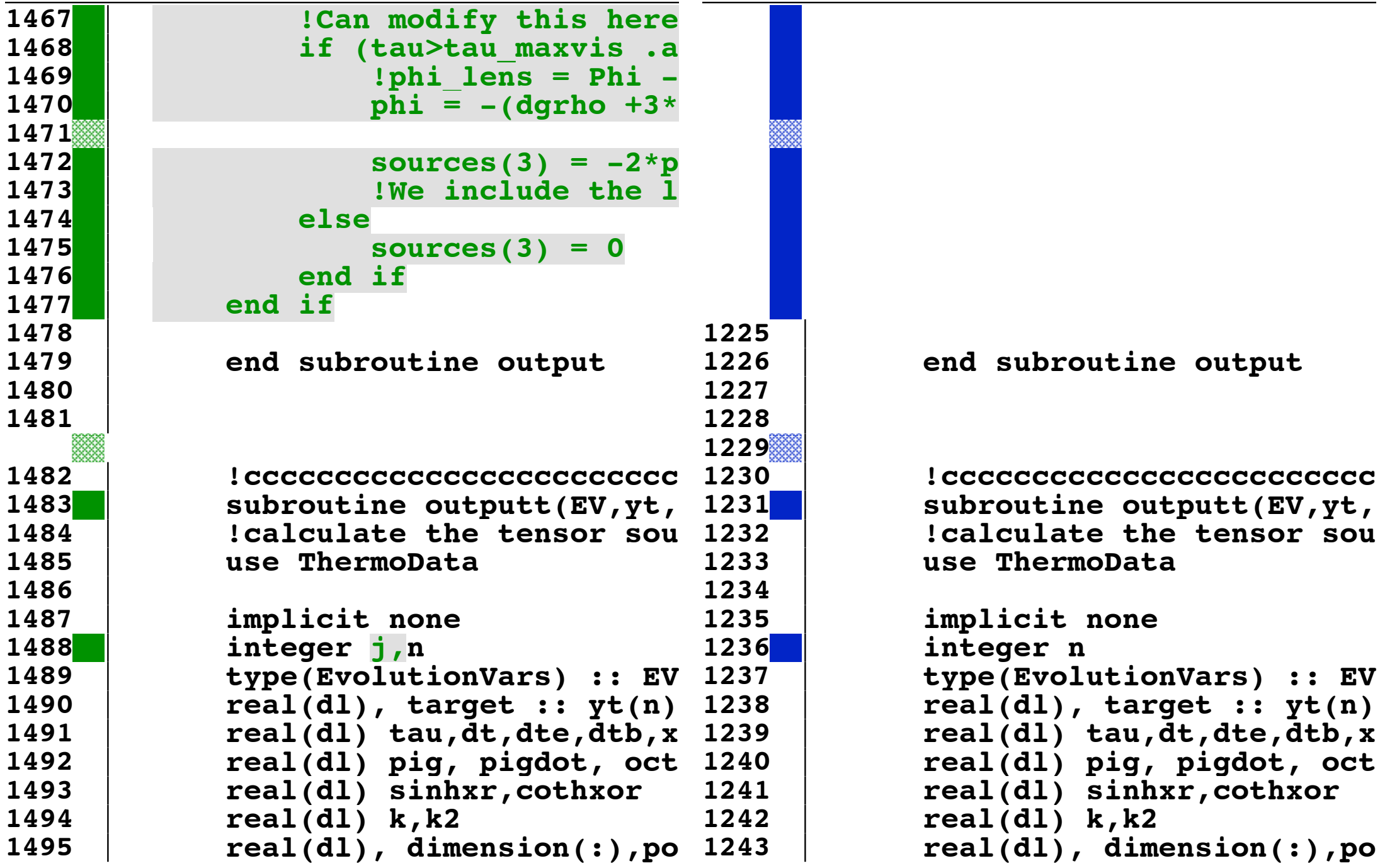
```
1407                ypolprime(2)= (pi
1408            else
1409                pigdot = -dopac(j
1410                +etak/EV%Kf(1)-
1411                ypolprime(2)= pig
1412            end if
1413        end if
1414
1415        pidot_sum =  pidot_sum +
1416        diff_rhopi = pidot_sum -
1417
1418        !Maple's fortran output -
1419        !2phi' term (\phi' + \psi
1420        ISW = (4.D0/3.D0*k*EV%Kf(
1421        -diff_rhopi/k**2-1.D0/ado
1422        -2.D0/k*adotoa/EV%Kf(1)*e
1423
1424        !e.g. to get only late-ti
1425        !  if (1/a-1 < 30) ISW=0
1426
1427        !The rest, note y(9)->oct
1428        sources(1)= ISW +  ((-9.D
1429        (11.D0/10.D0*sigma- 3.D0/
1430        (-180.D0*ypolprime(2)-30.
1431        (-(9.D0*pigdot+ 54.D0*ypo
1432        (-21.D0/5.D0*adotoa*sigma
1433        vbdot+3.D0/40.D0*qgdot- 9
1434        (-9.D0/160.D0*dopac(j)*pi
1435        (3.D0/16.D0*ddvis(j)*pig+
1436
```

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

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1437

```
1437        ! Doppler term
1438        !   sources(1)=  (sigma+v
1439        !         +1.D0/k/EV%Kf(1
1440
1441        !Equivalent full result
1442        !    t4 = 1.D0/adotoa
1443        !    t92 = k**2
1444        !   sources(1) = (4.D0/3.
1445        !       (3.D0/8.D0*ypol(2
1446        !    sources(1) = sources
1447        !         3.D0/8.D0*EV%Kf
1448        !        gpres)*sigma*exp
1449        !        EV%Kf(1)+(vbdot-
1450        !        5.D0*sigma*adoto
1451        !        27.D0/80.D0*ypol
1452        !        -9.D0/160.D0*dop
1453        !        160.D0*pig)*opac
1454        !        8.D0*ddvis(j)*yp
1455
1456
1457        if (x > 0._dl) then
1458            !E polarization sourc
1459            sources(2)=vis(j)*pol
1460            !factor of four becau
1461        else
1462            sources(2)=0
1463        end if
1464
1465        if (CTransScal%NumSources
1466            !Get lensing sources
```

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


/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1467
/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1225
!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
end subroutine output
!ccccccccccccccccccccccc
!ccccccccccccccccccccccc
subroutine outputt(EV,yt,
subroutine outputt(EV,yt,
!calculate the tensor sou
!calculate the tensor sou
use ThermoData
use ThermoData
implicit none
implicit none
integer j,n
integer n
type(EvolutionVars) :: EV
type(EvolutionVars) :: EV
real(dl), target :: yt(n)
real(dl), target :: yt(n)
real(dl) tau,dt,dte,dtb,x
real(dl) tau,dt,dte,dtb,x
real(dl) pig, pigdot, oct
real(dl) pig, pigdot, oct
real(dl) sinhxr,cothxor
real(dl) sinhxr,cothxor
real(dl) k,k2
real(dl) k,k2
real(dl), dimension(:),po
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

```
1552            vis(j)*polter*(k2 -6*
1553
1554            dtb=15._dl/4._dl*EV%q
1555        else
1556            dt=0._dl
1557            dte=0._dl
1558            dtb=0._dl
1559        end if
1560
1561        end subroutine outputt
1562
1563        !ccccccccccccccccccccccc
1564        subroutine outputv(EV,yv,
1565        !calculate the vector sou
1566        use ThermoData
1567
1568        implicit none
1569        integer j,n
1570        type(EvolutionVars) :: EV
1571        real(dl), target :: yv(n)
1572        real(dl) tau,dt,dte,dtb,x
1573        real(dl) vb,qg, pig, polt
1574        real(dl) k,k2
1575        real(dl), dimension(:),po
1576
1577        call derivsv(EV,EV%nvarv,
1578
```

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

```
1304            visibility*polter
1305
1306            dtb=15._dl/4._dl*EV%q
1307        else
1308            dt=0._dl
1309            dte=0._dl
1310            dtb=0._dl
1311        end if
1312
1313        end subroutine outputt
1314
1315        !ccccccccccccccccccccccc
1316        subroutine outputv(EV,yv,
1317        !calculate the vector sou
1318        use ThermoData
1319
1320        implicit none
1321        integer n
1322        type(EvolutionVars) :: EV
1323        real(dl), target :: yv(n)
1324        real(dl) tau,dt,dte,dtb,x
1325        real(dl) vb,qg, pig, polt
1326        real(dl) k,k2
1327        real(dl), dimension(:),po
1328        real(dl) opacity, dopacit
1329            visibility, dvisibili
1330
1331
1332        call derivsv(EV,EV%nvarv,
1333
```

/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

```
1606            dtb= -15._dl/2*polter
1607        else
1608            dt=0
1609            dte=0
1610            dtb=0
1611        end if
1612
1613        end subroutine outputv
1614
1615
1616        !cccccccccccccccccccccccc
1617        subroutine initial(EV,y,
1618        !  Initial conditions.
1619        use ThermoData
1620        implicit none
1621
1622        type(EvolutionVars) EV
1623        real(dl) y(EV%nvar)
1624        real(dl) Rp15,tau,x,x2,x3
1625        Rc,Rb,Rv,Rg,grhonu,chi
1626        real(dl) k,k2
1627        real(dl) a,a2, iqg, rhoma
1628        integer l,i, nu_i, j, ind
1629        integer, parameter :: i_c
1630        i_qg=5,i_qr=6,i_vb=7,i_pi
1631        integer, parameter :: i_m
1632        real(dl) initv(6,1:i_max)
1633
1634        nullify(EV%OutputTransfer
```

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

```
1364            dtb= -15._dl/2*polter
1365        else
1366            dt=0
1367            dte=0
1368            dtb=0
1369        end if
1370
1371        end subroutine outputv
1372
1373
1374        !cccccccccccccccccccccccc
1375        subroutine initial(EV,y,
1376        !  Initial conditions.
1377        use ThermoData
1378        implicit none
1379
1380        type(EvolutionVars) EV
1381        real(dl) y(EV%nvar)
1382        real(dl) Rp15,tau,x,x2,x3
1383            Rc,Rb,Rv,Rg,grhonu,ch
1384        real(dl) k,k2
1385        real(dl) a,a2, iqg, rhoma
1386        integer l,i, nu_i, j, ind
1387        integer, parameter :: i_c
1388            i_qg=5,i_qr=6,i_vb=7,
1389        integer, parameter :: i_m
1390        real(dl) initv(6,1:i_max)
1391
1392        nullify(EV%OutputTransfer
1393        nullify(EV%OutputSources)
```

```
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
```



```
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

```
1664 |             if (EV%k_buf > epsw*5
1665 |                 ep=ep0*5/Accuracy
1666 |                 if (HighAccuracyD
1667 |             else
1668 |                 ep=ep0
1669 |             end if
1670 |         else
1671 |             ep=ep0
1672 |         end if
1673 |         if (second_order_tightcou
1674 |         EV%TightSwitchoffTime = m
1675 |
1676 |
1677 |         y=0
1678 |
1679 |         !  k*tau, (k*tau)**2, (k*
1680 |         x=k*tau
1681 |         x2=x*x
1682 |         x3=x2*x
1683 |         rhomass =  sum(grhormass(
1684 |         grhonu=rhomass+grhornomas
1685 |
1686 |         om = (grhob+grhoc)/sqrt(3
1687 |         omtau=om*tau
1688 |         Rv=grhonu/(grhonu+grhog)
1689 |
1690 |         Rg = 1-Rv
1691 |         Rc=CP%omegac/(CP%omegac+C
1692 |         Rb=1-Rc
1693 |         Rp15=4*Rv+15
```

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

```
1424 |             if (EV%k_buf > epsw*5
1425 |                 ep=ep0*5/Accuracy
1426 |                 if (HighAccuracyD
1427 |             else
1428 |                 ep=ep0
1429 |             end if
1430 |         else
1431 |             ep=ep0
1432 |         end if
1433 |         if (second_order_tightcou
1434 |         EV%TightSwitchoffTime = m
1435 |
1436 |
1437 |         y=0
1438 |
1439 |         !  k*tau, (k*tau)**2, (k*
1440 |         x=k*tau
1441 |         x2=x*x
1442 |         x3=x2*x
1443 |         rhomass =  sum(grhormass(
1444 |         grhonu=rhomass+grhornomas
1445 |
1446 |         om = (grhob+grhoc)/sqrt(3
1447 |         omtau=om*tau
1448 |         Rv=grhonu/(grhonu+grhog)
1449 |
1450 |         Rg = 1-Rv
1451 |         Rc=CP%omegac/(CP%omegac+C
1452 |         Rb=1-Rc
1453 |         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

```
1724            initv(2,i_qg)=-Rc/9*o
1725            initv(2,i_qr)=initv(2
1726            initv(2,i_vb)=0.75_dl
1727            initv(2,i_pir)=-Rc*om
1728            initv(2,i_eta)= Rc*om
1729            initv(2,i_aj3r)=0
1730            !Baryon isocurvature
1731            if (Rc==0) stop 'Isoc
1732
1733            initv(3,:) = initv(2,
1734            initv(3,i_clxc) = ini
1735            initv(3,i_clxb)= init
1736
1737            !neutrino isocurvatur
1738
1739            initv(4,i_clxg)=Rv/Rg
1740            initv(4,i_clxr)=1-x2/
1741            initv(4,i_clxc)=-omta
1742            initv(4,i_clxb)= Rv/R
1743            iqg = - Rv/Rg*(x/3 -
1744            initv(4,i_qg) =iqg
1745            initv(4,i_qr) = x/3
1746            initv(4,i_vb)=0.75_dl
1747            initv(4,i_pir)=x2/Rp1
1748            initv(4,i_eta)=EV%Kf(
1749
1750            !neutrino isocurvatur
1751
1752            initv(5,i_clxg)=Rv/Rg
1753            initv(5,i_clxr)=-x -3
```

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

```
1484            initv(2,i_qg)=-Rc/9*o
1485            initv(2,i_qr)=initv(2
1486            initv(2,i_vb)=0.75_dl
1487            initv(2,i_pir)=-Rc*om
1488            initv(2,i_eta)= Rc*om
1489            initv(2,i_aj3r)=0
1490            !Baryon isocurvature
1491            if (Rc==0) call MpiSt
1492
1493            initv(3,:) = initv(2,
1494            initv(3,i_clxc) = ini
1495            initv(3,i_clxb)= init
1496
1497            !neutrino isocurvatur
1498
1499            initv(4,i_clxg)=Rv/Rg
1500            initv(4,i_clxr)=1-x2/
1501            initv(4,i_clxc)=-omta
1502            initv(4,i_clxb)= Rv/R
1503            iqg = - Rv/Rg*(x/3 -
1504            initv(4,i_qg) =iqg
1505            initv(4,i_qr) = x/3
1506            initv(4,i_vb)=0.75_dl
1507            initv(4,i_pir)=x2/Rp1
1508            initv(4,i_eta)=EV%Kf(
1509
1510            !neutrino isocurvatur
1511
1512            initv(5,i_clxg)=Rv/Rg
1513            initv(5,i_clxr)=-x -3
```

```
1754 |             initv(5,i_clxc)=-9*om
1755 |             initv(5,i_clxb)= 3*Rv
1756 |             iqg = Rv/Rg*(-1 + 3*R
1757 |             initv(5,i_qg) =iqg
1758 |             initv(5,i_qr) = 1 - x
1759 |             initv(5,i_vb)=0.75_dl
1760 |             initv(5,i_pir)=2*x/(4
1761 |             initv(5,i_eta)=2*EV%K
1762 |             initv(5,i_aj3r) = 3._
1763 |
1764 |             !quintessence isocurv
1765 |         end if
1766 |
1767 |         if (CP%Scalar_initial_con
1768 |             InitVec = 0
1769 |             do i=1,initial_nummod
1770 |                 InitVec = InitVec
1771 |             end do
1772 |         else
1773 |             InitVec = initv(CP%Sc
1774 |             if (CP%Scalar_initial
1775 |             !So we start with chi
1776 |         end if
1777 |
1778 |         y(1)=a
1779 |         y(2)= -InitVec(i_eta)*k/2
1780 |         !get eta_s*k, where eta_s
1781 |
1782 |         !  CDM
1783 |         y(3)=InitVec(i_clxc)
```



```
1514 |             initv(5,i_clxc)=-9*om
1515 |             initv(5,i_clxb)= 3*Rv
1516 |             iqg = Rv/Rg*(-1 + 3*R
1517 |             initv(5,i_qg) =iqg
1518 |             initv(5,i_qr) = 1 - x
1519 |             initv(5,i_vb)=0.75_dl
1520 |             initv(5,i_pir)=2*x/(4
1521 |             initv(5,i_eta)=2*EV%K
1522 |             initv(5,i_aj3r) = 3._
1523 |
1524 |             !quintessence isocurv
1525 |         end if
1526 |
1527 |         if (CP%Scalar_initial_con
1528 |             InitVec = 0
1529 |             do i=1,initial_nummod
1530 |                 InitVec = InitVec
1531 |             end do
1532 |         else
1533 |             InitVec = initv(CP%Sc
1534 |             if (CP%Scalar_initial
1535 |             !So we start with chi
1536 |         end if
1537 |
1538 |         y(1)=a
1539 |         y(2)= -InitVec(i_eta)*k/2
1540 |         !get eta_s*k, where eta_s
1541 |
1542 |         !  CDM
1543 |         y(3)=InitVec(i_clxc)
```

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

```
1784
1785          !  Baryons
1786          y(4)=InitVec(i_clxb)
1787          y(5)=InitVec(i_vb)
1788
1789          !  Photons
1790          y(EV%g_ix)=InitVec(i_clxg
1791          y(EV%g_ix+1)=InitVec(i_qg
1792
1793       !#SimDataReplace
1794       !     if (w_lam /= -1 .and. w_
1795             if (.not. is_cosmologica
1796       !#SimDataReplace
1797               y(EV%w_ix) = InitVec(
1798               y(EV%w_ix+1) = InitVe
1799          end if
1800
1801          !  Neutrinos
1802          y(EV%r_ix)=InitVec(i_clxr
1803          y(EV%r_ix+1)=InitVec(i_qr
1804          y(EV%r_ix+2)=InitVec(i_pi
1805
1806          if (EV%lmaxnr>2) then
1807              y(EV%r_ix+3)=InitVec(
1808          endif
1809
1810          if (CP%Num_Nu_massive ==
1811
1812          do nu_i = 1, CP%Nu_mass_e
1813              EV%MassiveNuApproxTim
```

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

```
1544
1545          !  Baryons
1546          y(4)=InitVec(i_clxb)
1547          y(5)=InitVec(i_vb)
1548
1549          !  Photons
1550          y(EV%g_ix)=InitVec(i_clxg
1551          y(EV%g_ix+1)=InitVec(i_qg
1552
1553          if (w_lam /= -1 .and. w_P
1554              y(EV%w_ix) = InitVec(
1555              y(EV%w_ix+1) = InitVe
1556          end if
1557
1558          !  Neutrinos
1559          y(EV%r_ix)=InitVec(i_clxr
1560          y(EV%r_ix+1)=InitVec(i_qr
1561          y(EV%r_ix+2)=InitVec(i_pi
1562
1563          if (EV%lmaxnr>2) then
1564              y(EV%r_ix+3)=InitVec(
1565          endif
1566
1567          if (CP%Num_Nu_massive ==
1568
1569          do nu_i = 1, CP%Nu_mass_e
1570              EV%MassiveNuApproxTim
```

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

```
1814 |            a_massive =  20000*k/
1815 |            if (a_massive >=0.99)
1816 |                EV%MassiveNuAppro
1817 |            else if (a_massive >
1818 |                EV%MassiveNuAppro
1819 |            end if
1820 |            ind = EV%nu_ix(nu_i)
1821 |            do  i=1,EV%nq(nu_i)
1822 |                y(ind:ind+2)=y(EV
1823 |                if (EV%lmaxnu_tau
1824 |                ind = ind + EV%lm
1825 |            end do
1826 |        end do
1827 |
1828 |        end subroutine initial
1829 |
1830 |
1831 |        !ccccccccccccccccccccccc
1832 |        subroutine initialt(EV,yt
1833 |        !  Initial conditions for
1834 |        use ThermoData
1835 |        implicit none
1836 |        real(dl) bigR,tau,x,aj3r,
1837 |        integer l
1838 |        type(EvolutionVars) EV
1839 |        real(dl) k,k2 ,a, omtau
1840 |        real(dl) yt(EV%nvart)
1841 |        real(dl) tens0, ep, tensf
1842 |
1843 |        if (CP%flat) then
```

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

```
1571 |            a_massive =  20000*k/
1572 |            if (a_massive >=0.99)
1573 |                EV%MassiveNuAppro
1574 |            else if (a_massive >
1575 |                EV%MassiveNuAppro
1576 |            end if
1577 |            ind = EV%nu_ix(nu_i)
1578 |            do  i=1,EV%nq(nu_i)
1579 |                y(ind:ind+2)=y(EV
1580 |                if (EV%lmaxnu_tau
1581 |                ind = ind + EV%lm
1582 |            end do
1583 |        end do
1584 |
1585 |        end subroutine initial
1586 |
1587 |
1588 |        !ccccccccccccccccccccccc
1589 |        subroutine initialt(EV,yt
1590 |        !  Initial conditions for
1591 |        use ThermoData
1592 |        implicit none
1593 |        real(dl) bigR,tau,x,aj3r,
1594 |        integer l
1595 |        type(EvolutionVars) EV
1596 |        real(dl) k,k2 ,a, omtau
1597 |        real(dl) yt(EV%nvart)
1598 |        real(dl) tens0, ep, tensf
1599 |
1600 |        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

```
1874 |
1875 |       a=tau*adotrad
1876 |       rhomass =  sum(grhormass(
1877 |       omtau = tau*(grhob+grhoc)
1878 |
1879 |       if (DoTensorNeutrinos) th
1880 |           bigR = (rhomass+grhor
1881 |       else
1882 |           bigR = 0._dl
1883 |       end if
1884 |
1885 |       x=k*tau
1886 |
1887 |       yt(1)=a
1888 |       tens0 = 1
1889 |
1890 |       yt(2)= tens0
1891 |       !commented things are for
1892 |       !-15/28._dl*x**2*(bigR-1)
1893 |
1894 |       elec=-tens0*(1+2*CP%curv/
1895 |
1896 |       !shear
1897 |       yt(3)=-5._dl/2/(bigR+5)*x
1898 |       !           + 15._dl/14*x*
1899 |
1900 |       yt(4:EV%nvart)=0._dl
1901 |
1902 |       !  Neutrinos
1903 |       if (DoTensorNeutrinos) th
```

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

```
1631 |
1632 |       a=tau*adotrad
1633 |       rhomass =  sum(grhormass(
1634 |       omtau = tau*(grhob+grhoc)
1635 |
1636 |       if (DoTensorNeutrinos) th
1637 |           bigR = (rhomass+grhor
1638 |       else
1639 |           bigR = 0._dl
1640 |       end if
1641 |
1642 |       x=k*tau
1643 |
1644 |       yt(1)=a
1645 |       tens0 = 1
1646 |
1647 |       yt(2)= tens0
1648 |       !commented things are for
1649 |       !-15/28._dl*x**2*(bigR-1)
1650 |
1651 |       elec=-tens0*(1+2*CP%curv/
1652 |
1653 |       !shear
1654 |       yt(3)=-5._dl/2/(bigR+5)*x
1655 |       !           + 15._dl/14*x*
1656 |
1657 |       yt(4:EV%nvart)=0._dl
1658 |
1659 |       !  Neutrinos
1660 |       if (DoTensorNeutrinos) th
```

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

```
1904            pir=-2._dl/3._dl/(big
1905            !            + (bigR-1
1906            aj3r=  -2._dl/21._dl/
1907            !            + 3._dl/7
1908            yt(EV%r_ix+2)=pir
1909            yt(EV%r_ix+3)=aj3r
1910            !Should set up massiv
1911        end if
1912
1913        end subroutine initialt
1914
1915        !cccccccccccccccccccccccc
1916        subroutine initialv(EV,yv
1917        !  Initial conditions for
1918
1919        implicit none
1920        real(dl) bigR,Rc,tau,x,pi
1921        type(EvolutionVars) EV
1922        real(dl) k,k2 ,a, omtau
1923        real(dl) yv(EV%nvarv)
1924
1925        if (CP%flat) then
1926            EV%k2_buf=EV%q2
1927            EV%k_buf=EV%q
1928        else
1929            stop 'Vectors not sup
1930        endif
1931
1932        k=EV%k_buf
1933        k2=EV%k2_buf
```

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

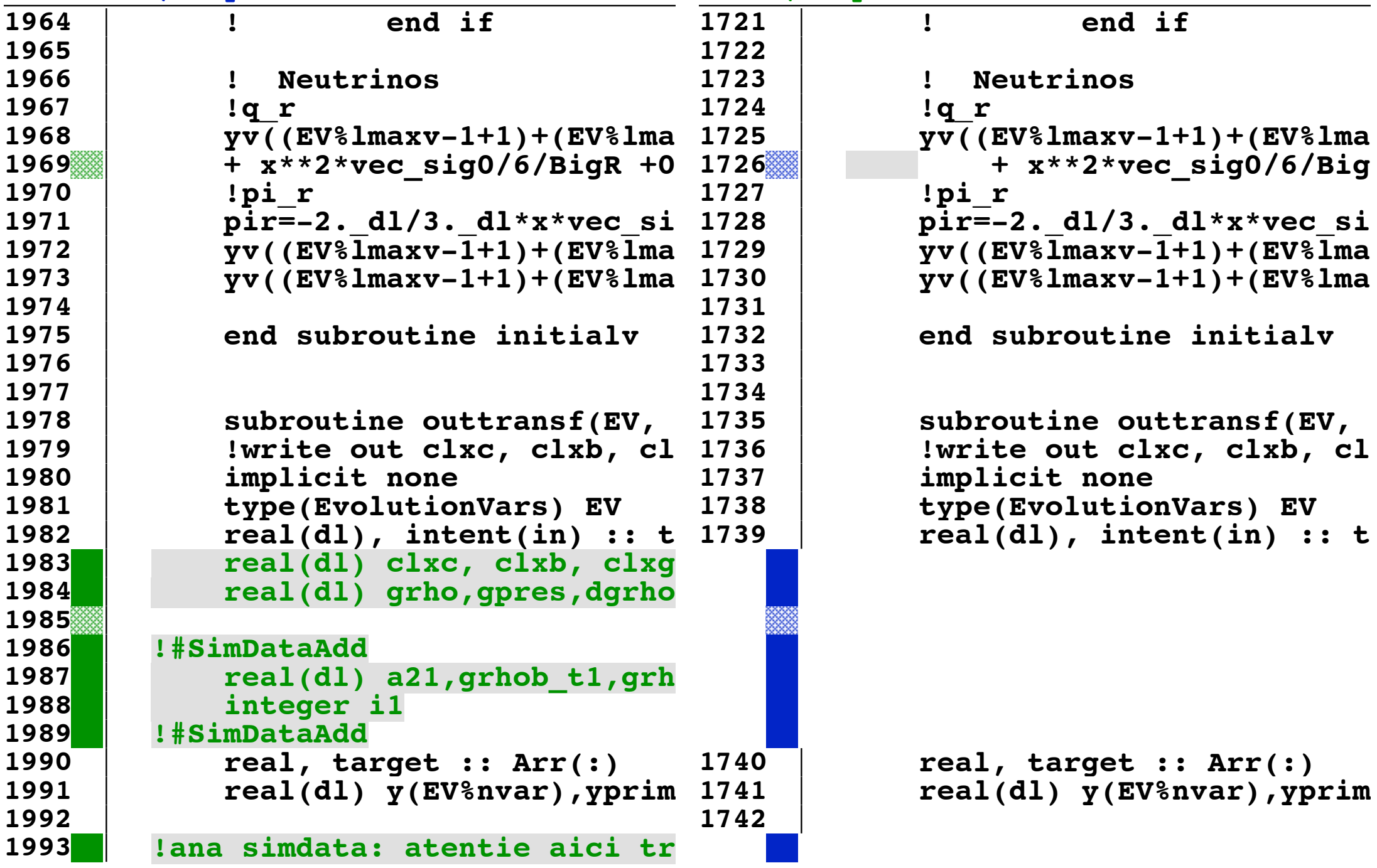
```
1661            pir=-2._dl/3._dl/(big
1662            !            + (bigR-1
1663            aj3r=  -2._dl/21._dl/
1664                !            + 3._
1665                yt(EV%r_ix+2)=pir
1666            yt(EV%r_ix+3)=aj3r
1667            !Should set up massiv
1668        end if
1669
1670        end subroutine initialt
1671
1672        !cccccccccccccccccccccccc
1673        subroutine initialv(EV,yv
1674        !  Initial conditions for
1675
1676        implicit none
1677        real(dl) bigR,Rc,tau,x,pi
1678        type(EvolutionVars) EV
1679        real(dl) k,k2 ,a, omtau
1680        real(dl) yv(EV%nvarv)
1681
1682        if (CP%flat) then
1683            EV%k2_buf=EV%q2
1684            EV%k_buf=EV%q
1685        else
1686            call MpiStop('Vectors
1687        endif
1688
1689        k=EV%k_buf
1690        k2=EV%k2_buf
```

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

```
1934
1935        omtau = tau*(grhob+grhoc)
1936
1937        a=tau*adotrad*(1+omtau/4)
1938
1939        x=k*tau
1940
1941        bigR = (grhornomass)/(grh
1942        Rc=CP%omegac/(CP%omegac+C
1943
1944        yv(1)=a
1945
1946
1947        yv(2)= vec_sig0*(1- 15._d
1948        !qg
1949        yv(4)= vec_sig0/3* (4*big
1950        (1 - 0.25_dl*omtau*(3*Rc-
1951        -x/2*Magnetic
1952        yv(3)= 3._dl/4*yv(4)
1953
1954        yv(5:EV%nvarv) = 0
1955
1956        !         if (.false.) the
1957        !          yv((EV%lmaxv-1+
1958        !          yv((EV%lmaxv-1+
1959        !          yv((EV%lmaxv-1+
1960        !          yv((EV%lmaxv-1+
1961        !          yv(4) = 0
1962        !          yv(3)= 3._dl/4*
1963        !           return
```

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

```
1691
1692        omtau = tau*(grhob+grhoc)
1693
1694        a=tau*adotrad*(1+omtau/4)
1695
1696        x=k*tau
1697
1698        bigR = (grhornomass)/(grh
1699        Rc=CP%omegac/(CP%omegac+C
1700
1701        yv(1)=a
1702
1703
1704        yv(2)= vec_sig0*(1- 15._d
1705        !qg
1706        yv(4)= vec_sig0/3* (4*big
1707             (1 - 0.25_dl*omtau*(3
1708             -x/2*Magnetic
1709        yv(3)= 3._dl/4*yv(4)
1710
1711        yv(5:EV%nvarv) = 0
1712
1713        !         if (.false.) the
1714        !          yv((EV%lmaxv-1+
1715        !          yv((EV%lmaxv-1+
1716        !          yv((EV%lmaxv-1+
1717        !          yv((EV%lmaxv-1+
1718        !          yv(4) = 0
1719        !          yv(3)= 3._dl/4*
1720        !           return
```


/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 1964
/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1721
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
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
/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 1743
1994
1995 yprime = 0
1996 EV%OutputTransfer => Arr
1997 call derivs(EV,EV%ScalEqs
1998 nullify(EV%OutputTransfer
1999
2000 Arr(Transfer_kh+1:Transfe
2001
2002 ! do i1=Transfer_kh, Tran
2003 !print*, 'equatio
2004
2005 ! end do
2006
2007 !#SimDataAdd +ana add
2008
2009 a = y(1)
2010 clxc = y(3)
2011 clxb = y(4)
2012
2013 k = EV%k_buf
2014 k2 = EV%k2_buf
2015
2016 dgrho = 0
2017 grho = 0
2018
2019 if (CP%Num_Nu_Massive > 0
2020 call MassiveNuVars(EV
2021 end if
2022
2023 dgrho = dgrho+(clxc*grhoc
1743 yprime = 0
1744 EV%OutputTransfer => Arr
1745 call derivs(EV,EV%ScalEqs
1746 nullify(EV%OutputTransfer
1747
1748 Arr(Transfer_kh+1:Transfe
1749

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

```
2024        grho =  grho+(grhoc+grhob
2025
2026        a21 = a*a
2027
2028
2029        vb1 = y(5)
2030
2031        grhob_t1 = grhob/a
2032        grhoc_t1 = grhoc/a
2033        grhor_t1 = grhornomass/a2
2034        grhog_t1 = grhog/a21
2035        grhov_t1 = grho_de(a)/a21
2036
2037        grho = grhob_t1+grhoc_t1+
2038
2039        adotoa1 = sqrt(grho/3.d0)
2040
2041        dgq = grhob_t1*vb1
2042
2043        phi1 = -(dgrho + 3.d0*dgq
2044
2045        Arr(Transfer_Psi) = -phi1
2046
2047     !print*, 'equations: i =', T
2048      !  print*, 'dgrho, dqq, k,
2049       ! print*, 'equations I: i
2050
2051    !#SimDataAdd
2052
2053        end subroutine outtransf
```

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

```
1750        end subroutine outtransf
```

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

```
2054
2055        !ccccccccccccccccccccccc
2056        subroutine derivs(EV,n,ta
2057        !  Evaluate the time deri
2058        !  ayprime is not necessa
2059        use ThermoData
2060        use MassiveNu
2061        implicit none
2062        type(EvolutionVars) EV
2063
2064        integer n,nu_i
2065        real(dl) ay(n),ayprime(n)
2066        real(dl) tau,w
2067        real(dl) k,k2
2068
2069        !  Internal variables.
2070
2071        real(dl) opacity
2072        real(dl) photbar,cs2,pb43
2073        clxcdot,clxbdot,adotdota,
2074        real(dl) q,aq,v
2075        real(dl) G11_t,G30_t, wnu
2076
2077        real(dl) dgq,grhob_t,grho
2078        real(dl) qgdot,qrdot,pigd
2079        real(dl) a,a2,z,clxc,clxb
2080        real(dl) clxq, vq,  E2, d
2081        integer l,i,ind, ind2, of
2082        real(dl) dgs,sigmadot,dz
2083        real(dl) dgpi,dgrho_matte
```

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

```
1751
1752        !ccccccccccccccccccccccc
1753        subroutine derivs(EV,n,ta
1754        !  Evaluate the time deri
1755        !  ayprime is not necessa
1756        use ThermoData
1757        use MassiveNu
1758        implicit none
1759        type(EvolutionVars) EV
1760
1761        integer n,nu_i
1762        real(dl) ay(n),ayprime(n)
1763        real(dl) tau,w
1764        real(dl) k,k2
1765
1766        !  Internal variables.
1767
1768        real(dl) opacity
1769        real(dl) photbar,cs2,pb43
1770            clxcdot,clxbdot,adotd
1771        real(dl) q,aq,v
1772        real(dl) G11_t,G30_t, wnu
1773
1774        real(dl) dgq,grhob_t,grho
1775        real(dl) qgdot,qrdot,pigd
1776        real(dl) a,a2,z,clxc,clxb
1777        real(dl) clxde, qde,  E2,
1778        integer l,i,ind, ind2, of
1779        real(dl) dgs,sigmadot,dz
1780        real(dl) dgpi,dgrho_matte
```

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

```
2084        !non-flat vars
2085        real(dl) cothxor !1/tau i
2086
2087     !ana simdata
2088     !     #SimDataAdd
2089        real(dl) phi
2090       !#SimDataAdd
2091     !end ana simdata
2092
2093
2094        k=EV%k_buf
2095        k2=EV%k2_buf
2096
2097        a=ay(1)
2098        a2=a*a
2099
2100        etak=ay(2)
2101
2102        !  CDM variables
2103        clxc=ay(3)
2104
2105        !  Baryon variables
2106        clxb=ay(4)
2107        vb=ay(5)
2108
2109        !  Compute expansion rate
2110
2111        grhob_t=grhob/a
2112        grhoc_t=grhoc/a
2113        grhor_t=grhornomass/a2
```

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

```
1781        !non-flat vars
1782        real(dl) cothxor !1/tau i
1783        !Variables for source cal
1784        real(dl) diff_rhopi, pido
1785        real(dl) E(2:3), Edot(2:3
1786        real(dl) phidot, polterdo
1787        real(dl) ddopacity, visib
1788        real(dl) ISW, quadrupole_
1789
1790        k=EV%k_buf
1791        k2=EV%k2_buf
1792
1793        a=ay(1)
1794        a2=a*a
1795
1796        etak=ay(2)
1797
1798        !  CDM variables
1799        clxc=ay(3)
1800
1801        !  Baryon variables
1802        clxb=ay(4)
1803        vb=ay(5)
1804
1805        !  Compute expansion rate
1806
1807        grhob_t=grhob/a
1808        grhoc_t=grhoc/a
1809        grhor_t=grhornomass/a2
```

/Users/lp1opa/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/lp1opa/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

```
2173 |             !Approximate total de
2174 |             z=(0.5_dl*dgrho/k + e
2175 |             dz= -adotoa*z - 0.5_d
2176 |             clxr=-4*dz/k
2177 |             qr=-4._dl/3*z
2178 |             pir=0
2179 |         else
2180 |             !  Massless neutrinos
2181 |             clxr=ay(EV%r_ix)
2182 |             qr  =ay(EV%r_ix+1)
2183 |             pir =ay(EV%r_ix+2)
2184 |         endif
2185 |
2186 |         if (EV%no_phot_multpoles)
2187 |             if (.not. EV%no_nu_mu
2188 |                 z=(0.5_dl*dgrho/k
2189 |                 dz= -adotoa*z - 0
2190 |                 clxg=-4*dz/k-4/k*
2191 |                 qg=-4._dl/3*z
2192 |             else
2193 |                 clxg=clxr-4/k*opa
2194 |                 qg=qr
2195 |             end if
2196 |             pig=0
2197 |         else
2198 |             !  Photons
2199 |             clxg=ay(EV%g_ix)
2200 |             qg=ay(EV%g_ix+1)
2201 |             if (.not. EV%TightCou
2202 |         end if
```

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

```
1859 |             !Approximate total de
1860 |             z=(0.5_dl*dgrho/k + e
1861 |             dz= -adotoa*z - 0.5_d
1862 |             clxr=-4*dz/k
1863 |             qr=-4._dl/3*z
1864 |             pir=0
1865 |         else
1866 |             !  Massless neutrinos
1867 |             clxr=ay(EV%r_ix)
1868 |             qr  =ay(EV%r_ix+1)
1869 |             pir =ay(EV%r_ix+2)
1870 |         endif
1871 |
1872 |         if (EV%no_phot_multpoles)
1873 |             if (.not. EV%no_nu_mu
1874 |                 z=(0.5_dl*dgrho/k
1875 |                 dz= -adotoa*z - 0
1876 |                 clxg=-4*dz/k-4/k*
1877 |                 qg=-4._dl/3*z
1878 |             else
1879 |                 clxg=clxr-4/k*opa
1880 |                 qg=qr
1881 |             end if
1882 |             pig=0
1883 |         else
1884 |             !  Photons
1885 |             clxg=ay(EV%g_ix)
1886 |             qg=ay(EV%g_ix+1)
1887 |             if (.not. EV%TightCou
1888 |         end if
```

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

```
2203
2204        !  8*pi*a*a*SUM[rho_i*clx
2205        dgrho=dgrho + grhog_t*clx
2206
2207        !  8*pi*a*a*SUM[(rho_i+p_
2208        dgq=dgq + grhog_t*qg+grho
2209
2210        !  Photon mass density ov
2211        photbar=grhog_t/grhob_t
2212        pb43=4._dl/3*photbar
2213
2214        ayprime(1)=adotoa*a
2215
2216
2217        !  Get sigma (shear) and
2218        ! have to get z from eta
2219        z=(0.5_dl*dgrho/k + etak)
2220        if (CP%flat) then
2221            !eta*k equation
2222            sigma=(z+1.5_dl*dgq/k
2223            ayprime(2)=0.5_dl*dgq
2224        else
2225            sigma=(z+1.5_dl*dgq/k
2226            ayprime(2)=0.5_dl*dgq
2227        end if
2228
2229    !#SimDataReplace
2230    !     if (w_lam /= -1 .and. w_
2231          if (.not. is_cosmologica
2232      ayprime(EV%w_ix)= 0.d0 !-3*
```

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

```
1889
1890        !  8*pi*a*a*SUM[rho_i*clx
1891        dgrho=dgrho + grhog_t*clx
1892
1893        !  8*pi*a*a*SUM[(rho_i+p_
1894        dgq=dgq + grhog_t*qg+grho
1895
1896        !  Photon mass density ov
1897        photbar=grhog_t/grhob_t
1898        pb43=4._dl/3*photbar
1899
1900        ayprime(1)=adotoa*a
1901
1902
1903        !  Get sigma (shear) and
1904        ! have to get z from eta
1905        z=(0.5_dl*dgrho/k + etak)
1906        if (CP%flat) then
1907            !eta*k equation
1908            sigma=(z+1.5_dl*dgq/k
1909            ayprime(2)=0.5_dl*dgq
1910        else
1911            sigma=(z+1.5_dl*dgq/k
1912            ayprime(2)=0.5_dl*dgq
1913        end if
1914
1915        if (w_lam /= -1 .and. w_P
1916            ayprime(EV%w_ix)= -3*
1917              - k*qde -(1+w_lam
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2233

```
2233
2234    !         ayprime(EV%w_ix)= -3
2235    !         -(1+w_lam)*k*vq -(1+
2236
2237    !         ayprime(EV%w_ix+1) =
2238
2239            ayprime(EV%w_ix+1) =
2240            !-adotoa*a*CP%wa/(1.d
2241        end if
2242
2243    !#SimDataReplace
2244
2245        if (associated(EV%OutputT
2246            EV%OutputTransfer(Tra
2247            EV%OutputTransfer(Tra
2248            EV%OutputTransfer(Tra
2249            EV%OutputTransfer(Tra
2250            EV%OutputTransfer(Tra
2251            clxnu_all=0
2252            dgpi  = grhor_t*pir +
2253            if (CP%Num_Nu_Massive
2254                call MassiveNuVar
2255            end if
2256            EV%OutputTransfer(Tra
2257            EV%OutputTransfer(Tra
2258            EV%OutputTransfer(Tra
2259            EV%OutputTransfer(Tra
2260            !Transfer_Weyl is k^2
2261            EV%OutputTransfer(Tra
2262            EV%OutputTransfer(Tra
```

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

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2263

```
2263            EV%OutputTransfer(Tra
2264            EV%OutputTransfer(Tra
2265        !ana simdata
2266
2267            !print*,'EV%OutputTra
2268            !print*, 'EV%OutputTr
2269            !print*,' EV%OutputTr
2270            !print*,' EV%OutputTr
2271            !print*, 'EV%OutputTr
2272            !print*, 'EV%OutputTr
2273            !print*, 'EV%OutputTr
2274            !print*, 'EV%OutputTr
2275            !print*, 'EV%OutputTr
2276            !Transfer_Weyl is k^2
2277            !print*,'EV%OutputTra
2278            !print*, 'EV%OutputTr
2279            !print*,  'EV%OutputT
2280            !print*, 'EV%OutputTr
2281
2282         a2 = a*a
2283
2284            vb = ay(5)
2285
2286            grhob_t = grhob/a
2287            grhoc_t = grhoc/a
2288            grhor_t = grhornomass/a2
2289            grhog_t = grhog/a2
2290            grhov_t = grho_de(a)/a2
2291
2292            grho = grhob_t+grhoc_t+gr
```

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

/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

```
2323          !  Use explicit equation
2324
2325          if (EV%TightCoupling) the
2326              !  ddota/a
2327
2328      !#SimDataReplace
2329      !         gpres=gpres+ (grhog_
2330
2331              gpres=gpres+ (grhog_t
2332      !#SimDataReplace
2333
2334
2335              adotdota=(adotoa*adot
2336
2337              pig = 32._dl/45/opaci
2338
2339              !  First-order approx
2340              slip = - (2*adotoa/(1
2341              +(-adotdota*vb-k/2*ad
2342
2343              if (second_order_tigh
2344                  ! by Francis-Yan
2345                  !AL: First order
2346
2347                  !  8*pi*G*a*a*SUM
2348                  dgs = grhog_t*pig
2349
2350                  ! Define shear de
2351                  sigmadot = -2*ado
2352
```

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

```
1935          !  Use explicit equation
1936
1937          if (EV%TightCoupling) the
1938              !  ddota/a
1939              gpres=gpres_nu+ (grho
1940              adotdota=(adotoa*adot
1941
1942              pig = 32._dl/45/opaci
1943
1944              !  First-order approx
1945              slip = - (2*adotoa/(1
1946                  +(-adotdota*vb-k/
1947
1948              if (second_order_tigh
1949                  ! by Francis-Yan
1950                  !AL: First order
1951
1952                  !  8*pi*G*a*a*SUM
1953                  dgs = grhog_t*pig
1954
1955                  ! Define shear de
1956                  sigmadot = -2*ado
1957
```

/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

```
2411                if (EV%lmaxgpol>2
2412                 ayprime(ix) =
2413                 do l=3,EV%lma
2414                  ix=ix+1
2415                  ayprime(i
2416                 end do
2417                 ix=ix+1
2418                 !truncate
2419                 ayprime(ix)=-
2420                 k*EV%poltrunc
2421                else !closed case
2422                 ayprime(ix) =
2423                endif
2424              end if
2425           end if
2426
2427           if (.not. EV%no_nu_multpo
2428            !  Massless neutrino
2429            clxrdot=-k*(4._dl/3._
2430            ayprime(EV%r_ix)=clxr
2431            qrdot=EV%denlk(1)*clx
2432            ayprime(EV%r_ix+1)=qr
2433            if (EV%high_ktau_neut
2434             !ufa approximatio
2435             !Method from arXi
2436             !
2437             !
2438             !
2439             !
2440             !
```

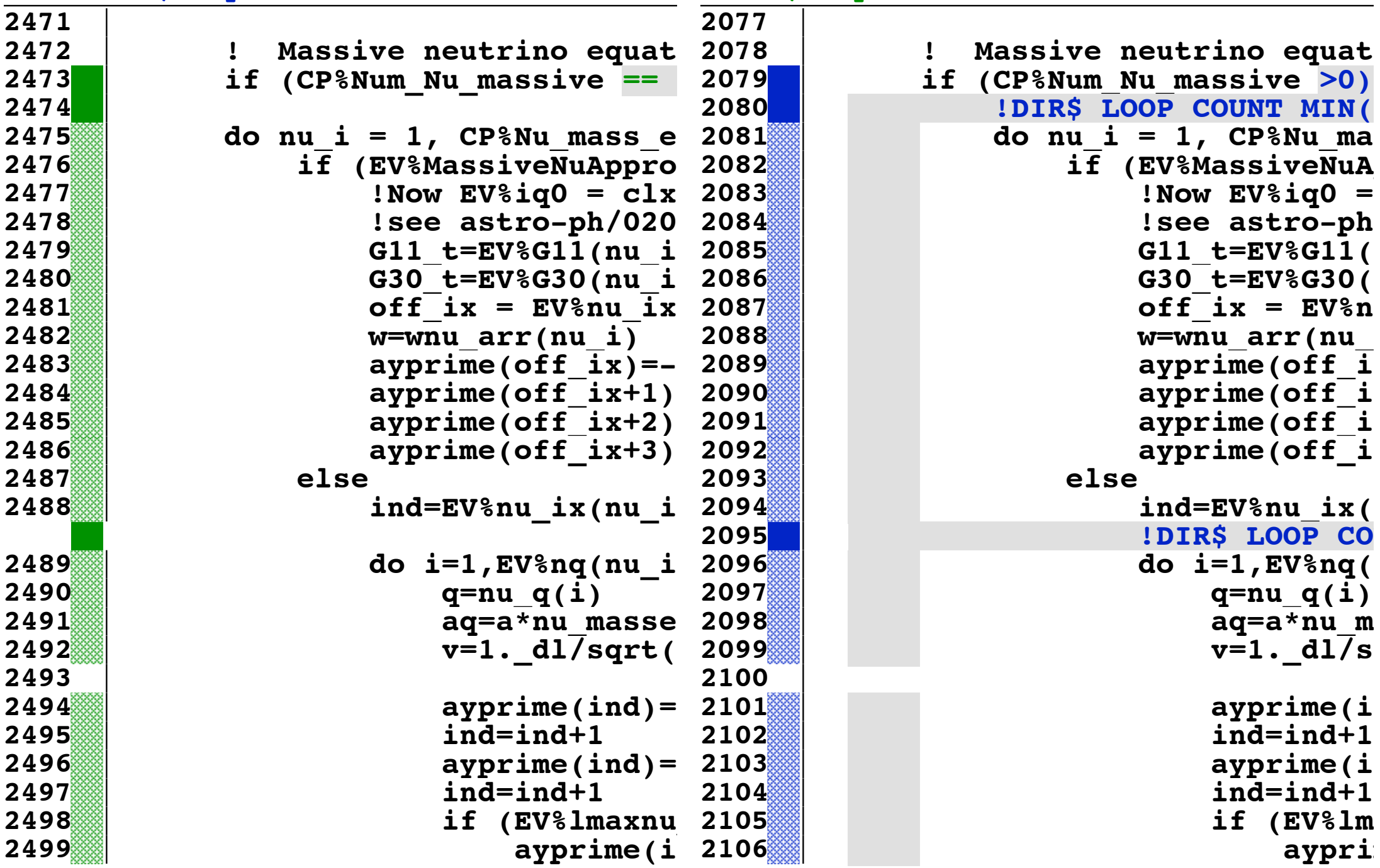
/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2017

```
2017                if (EV%lmaxgpol>2
2018                 ayprime(ix) =
2019                 do l=3,EV%lma
2020                  ix=ix+1
2021                  ayprime(i
2022                 end do
2023                 ix=ix+1
2024                 !truncate
2025                 ayprime(ix)=-
2026                  k*EV%polt
2027                else !closed case
2028                 ayprime(ix) =
2029                endif
2030              end if
2031           end if
2032
2033           if (.not. EV%no_nu_multpo
2034            !  Massless neutrino
2035            clxrdot=-k*(4._dl/3._
2036            ayprime(EV%r_ix)=clxr
2037            qrdot=EV%denlk(1)*clx
2038            ayprime(EV%r_ix+1)=qr
2039            if (EV%high_ktau_neut
2040             !ufa approximatio
2041             !Method from arXi
2042             !
2043             !
2044             !
2045             !
2046             !
```

```
2441 |                 !
2442 |                 !
2443 |                 !
2444 |                 !
2445 |                 pirdot= -3*pir*co
2446 |                 ayprime(EV%r_ix+2
2447 |
2448 |                 !
2449 |                 !
2450 |                 !
2451 |                 !
2452 |                 !                   a
2453 |             else
2454 |                 ix=EV%r_ix+2
2455 |                 if (EV%lmaxnr>2)
2456 |                     pirdot=EV%den
2457 |                     ayprime(ix)=p
2458 |                     do l=3,EV%lma
2459 |                         ix=ix+1
2460 |                         ayprime(i
2461 |                     end do
2462 |                     !  Truncate t
2463 |                     ix=ix+1
2464 |                     ayprime(ix)=k
2465 |                 else
2466 |                     pirdot=EV%den
2467 |                     ayprime(ix)=p
2468 |                 end if
2469 |             end if
2470 |         end if ! no_nu_multpoles
```



```
2047 |                 !
2048 |                 !
2049 |                 !
2050 |                 !
2051 |                 pirdot= -3*pir*co
2052 |                 ayprime(EV%r_ix+2
2053 |
2054 |                 !
2055 |                 !
2056 |                 !
2057 |                 !
2058 |                 !                   a
2059 |             else
2060 |                 ix=EV%r_ix+2
2061 |                 if (EV%lmaxnr>2)
2062 |                     pirdot=EV%den
2063 |                     ayprime(ix)=p
2064 |                     do l=3,EV%lma
2065 |                         ix=ix+1
2066 |                         ayprime(i
2067 |                     end do
2068 |                     !  Truncate t
2069 |                     ix=ix+1
2070 |                     ayprime(ix)=k
2071 |                 else
2072 |                     pirdot=EV%den
2073 |                     ayprime(ix)=p
2074 |                 end if
2075 |             end if
2076 |         end if ! no_nu_multpoles
```


/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2471
/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2077
! Massive neutrino equat
if (CP%Num_Nu_massive ==
do nu_i = 1, CP%Nu_mass_e
if (EV%MassiveNuAppro
!Now EV%iq0 = clx
!see astro-ph/020
G11_t=EV%G11(nu_i
G30_t=EV%G30(nu_i
off_ix = EV%nu_ix
w=wnu_arr(nu_i)
ayprime(off_ix)=-
ayprime(off_ix+1)
ayprime(off_ix+2)
ayprime(off_ix+3)
else
ind=EV%nu_ix(nu_i
do i=1,EV%nq(nu_i
q=nu_q(i)
aq=a*nu_masse
v=1._dl/sqrt(
ayprime(ind)=
ind=ind+1
ayprime(ind)=
ind=ind+1
if (EV%lmaxnu
ayprime(i
! Massive neutrino equat
if (CP%Num_Nu_massive >0)
!DIR$ LOOP COUNT MIN(
do nu_i = 1, CP%Nu_ma
if (EV%MassiveNuA
!Now EV%iq0 =
!see astro-ph
G11_t=EV%G11(
G30_t=EV%G30(
off_ix = EV%n
w=wnu_arr(nu_
ayprime(off_i
ayprime(off_i
ayprime(off_i
ayprime(off_i
else
ind=EV%nu_ix(
!DIR$ LOOP CO
do i=1,EV%nq(
q=nu_q(i)
aq=a*nu_m
v=1._dl/s
ayprime(i
ind=ind+1
ayprime(i
ind=ind+1
if (EV%lm
aypri

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

```
2500                        else
2501                            ayprime(i
2502                            +k*8._dl/
2503                            do l=3,EV
2504                                ind=i
2505                                aypri
2506                            end do
2507                            !  Trunca
2508                            ind = ind
2509                            ayprime(i
2510                        end if
2511                        ind = ind+1
2512                    end do
2513                end if
2514            end do
2515
2516            if (EV%has_nu_relativisti
2517                ind=EV%nu_pert_ix
2518                ayprime(ind)=+k*a2*qr
2519                ind2= EV%r_ix
2520                do l=1,EV%lmaxnu_pert
2521                    ind=ind+1
2522                    ind2=ind2+1
2523                    ayprime(ind)= -a2
2524                    +   (EV%denlk(l)*
2525                end do
2526                ind=ind+1
2527                ind2=ind2+1
2528                ayprime(ind)= k*(ay(i
2529            end if
```

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

```
2107                        else
2108                            aypri
2109                                +
2110                            do l=
2111                                i
2112                                a
2113                            end d
2114                            !  Tr
2115                            ind =
2116                            aypri
2117                        end if
2118                        ind = ind
2119                    end do
2120                end if
2121            end do
2122
2123            if (EV%has_nu_relativ
2124                ind=EV%nu_pert_ix
2125                ayprime(ind)=+k*a
2126                ind2= EV%r_ix
2127                do l=1,EV%lmaxnu_
2128                    ind=ind+1
2129                    ind2=ind2+1
2130                    ayprime(ind)=
2131                        +   (EV%d
2132                end do
2133                ind=ind+1
2134                ind2=ind2+1
2135                ayprime(ind)= k*(
2136            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) =
```

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

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2167

```
2167                                E(2) = pi
2168                                octg=0
2169                            end if
2170                            octgdot=0
2171                        else
2172                            octg=ay(EV%g_
2173                            octgdot=aypri
2174                        end if
2175                    end if
2176
2177                    dgpi  = grhor_t*pir +
2178                    dgpi_diff = 0  !sum (
2179                    pidot_sum = grhog_t*p
2180                    clxnu =0
2181                    if (CP%Num_Nu_Massive
2182                        call MassiveNuVar
2183                            dgpi_diff=dgp
2184                    end if
2185                    diff_rhopi = pidot_su
2186                    gpres=gpres_nu+ (grho
2187
2188                    phi = -((dgrho +3*dgq
2189
2190                    if (associated(EV%Out
2191                        EV%OutputTransfer
2192                        EV%OutputTransfer
2193                        EV%OutputTransfer
2194                        EV%OutputTransfer
2195                        EV%OutputTransfer
2196                        EV%OutputTransfer
```

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

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

```
2197                    EV%OutputTransfer
2198                    EV%OutputTransfer
2199                    EV%OutputTransfer
2200                    !Transfer_Weyl is
2201                    EV%OutputTransfer
2202                    EV%OutputTransfer
2203                    EV%OutputTransfer
2204                    EV%OutputTransfer
2205                end if
2206                if (associated(EV%Out
2207
2208                    call IonizationFu
2209                        visibility, d
2210
2211                    tau0 = CP%tau0
2212                    phidot = (1.0d0/2
2213                        diff_rhopi+ k
2214                    !time derivative
2215                    sigmadot = -adoto
2216                    !quadrupole sourc
2217                    polter = pig/10+9
2218                    polterdot = (1.0d
2219                    polterddot = -2.0
2220                        k*sigma - 4.0
2221                        50.0d0*k*octg
2222                        *k*EV%Kf(2)*E
2223                        polter - 3.0d
2224                        + (7.0d0/10.0
2225                    !Temperature sour
2226
```

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

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

```
2227                    !2phi' term (\phi
2228                    ISW = 2*phidot*ex
2229                    monopole_source =
2230                    doppler = ((sigma
2231                    quadrupole_source
2232                        + (k**2*polte
2233
2234                    EV%OutputSources(
2235
2236                    if (tau < tau0) t
2237                        !E polarizati
2238                        EV%OutputSour
2239                        !factor of fo
2240                    else
2241                        EV%OutputSour
2242                    end if
2243
2244                    if (size(EV%Outpu
2245                        !Get lensing
2246                        !Can modify t
2247                        if (tau>tau_m
2248                            EV%Output
2249                            !We inclu
2250                        else
2251                            EV%Output
2252                        end if
2253                    end if
2254                    if (associated(EV
2255                        call custom_s
2256                            grhob_t,g
```

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

```
      end subroutine derivs



      subroutine derivsv(EV,n,t
      !  Evaluate the time deri
      use ThermoData
      use MassiveNu
      implicit none
      type(EvolutionVars) EV
      integer n,l
      real(dl), target ::  yv(n
      real(dl) ep,tau,grho,rhop
      logical finished_tightcou
      real(dl), dimension(:),po
      real(dl)  grhob_t,grhor_t
      real(dl) sigma, qg,pig, q
      real(dl) k,k2,a,a2, adotd
      real(dl) pir,adotoa
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2257

```
                                      k, etak,
                                      dgrho, cl
                                      dgq, qg,
                                      dgpi, pig
                                      polter, p
                                      opacity,
                                      tau0, tau
                              end if
                          end if
                      end if

      end subroutine derivs



      subroutine derivsv(EV,n,t
      !  Evaluate the time deri
      use ThermoData
      use MassiveNu
      implicit none
      type(EvolutionVars) EV
      integer n,l
      real(dl), target ::  yv(n
      real(dl) ep,tau,grho,rhop
      logical finished_tightcou
      real(dl), dimension(:),po
      real(dl)  grhob_t,grhor_t
      real(dl) sigma, qg,pig, q
      real(dl) k,k2,a,a2, adotd
      real(dl) pir,adotoa
```

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

```
2550
2551        k2=EV%k2_buf
2552        k=EV%k_buf
2553
2554        !E and B start at l=2. Se
2555        E => yv(EV%lmaxv+3:)
2556        Eprime=> yvprime(EV%lmaxv
2557        B => E(EV%lmaxpolv:)
2558        Bprime => Eprime(EV%lmaxp
2559        neutprime => Bprime(EV%lm
2560        neut => B(EV%lmaxpolv+1:)
2561
2562        a=yv(1)
2563
2564        sigma=yv(2)
2565
2566        a2=a*a
2567
2568        !  Get sound speed and op
2569
2570        call thermo(tau,cs2,opaci
2571        if (k > 0.06_dl*epsw) the
2572            ep=ep0
2573        else
2574            ep=0.2_dl*ep0
2575        end if
2576
2577        finished_tightcoupling =
2578        ((k/opacity > ep).or.(1._
2579
```

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

```
2287
2288        k2=EV%k2_buf
2289        k=EV%k_buf
2290
2291        !E and B start at l=2. Se
2292        E => yv(EV%lmaxv+3:)
2293        Eprime=> yvprime(EV%lmaxv
2294        B => E(EV%lmaxpolv:)
2295        Bprime => Eprime(EV%lmaxp
2296        neutprime => Bprime(EV%lm
2297        neut => B(EV%lmaxpolv+1:)
2298
2299        a=yv(1)
2300
2301        sigma=yv(2)
2302
2303        a2=a*a
2304
2305        !  Get sound speed and op
2306
2307        call thermo(tau,cs2,opaci
2308        if (k > 0.06_dl*epsw) the
2309            ep=ep0
2310        else
2311            ep=0.2_dl*ep0
2312        end if
2313
2314        finished_tightcoupling =
2315            ((k/opacity > ep).or.
2316
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2580

```
2580
2581        ! Compute expansion rate
2582        ! Also calculate gpres: 8
2583        grhob_t=grhob/a
2584        grhoc_t=grhoc/a
2585        grhor_t=grhornomass/a2
2586        grhog_t=grhog/a2
2587
2588    !#SimDataReplace
2589        grhov_t=grho_de(a)/a2
2590        !grhov_t=grhov*a**(-1-3*w
2591    !#SimDataReplace
2592
2593
2594        grho=grhob_t+grhoc_t+grho
2595    !#SimDataReplace
2596    !    gpres=(grhog_t+grhor_t)/
2597
2598        gpres=(grhog_t+grhor_t)/3
2599    !#SimDataReplace
2600        adotoa=sqrt(grho/3._dl)
2601        adotdota=(adotoa*adotoa-g
2602
2603        photbar=grhog_t/grhob_t
2604        pb43=4._dl/3*photbar
2605
2606        yvprime(1)=adotoa*a
2607
2608        vb = yv(3)
2609        qg = yv(4)
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2317

```
2317
2318        ! Compute expansion rate
2319        ! Also calculate gpres: 8
2320        grhob_t=grhob/a
2321        grhoc_t=grhoc/a
2322        grhor_t=grhornomass/a2
2323        grhog_t=grhog/a2
2324        grhov_t=grhov*a**(-1-3*w_
2325
2326        grho=grhob_t+grhoc_t+grho
2327        gpres=(grhog_t+grhor_t)/3
2328
2329        adotoa=sqrt(grho/3._dl)
2330        adotdota=(adotoa*adotoa-g
2331
2332        photbar=grhog_t/grhob_t
2333        pb43=4._dl/3*photbar
2334
2335        yvprime(1)=adotoa*a
2336
2337        vb = yv(3)
2338        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

```
2670            - pb43/(1+pb43)/opaci
2671            ( 2*pb43*adotoa**2/(1
2672            )/(1+pb43)
2673
2674            !  Equation for the p
2675            ! Get drag from vbdot
2676            yvprime(4)=-0.5_dl*k*
2677            (vbdot+adotoa*vb)/pho
2678
2679            !  Set the derivative
2680            yvprime(5:n)=0._dl
2681            yv(5)=pig
2682            E(2)=  pig/4
2683         endif
2684
2685         yvprime(3) = vbdot
2686
2687         !  Neutrino equations:
2688
2689         !  Massless neutrino anis
2690         pir=neut(2)
2691         neutprime(1)= -0.5_dl*k*p
2692         neutprime(2)=2._dl/5*k*qr
2693         !  And for the moments
2694         do  l=3,EV%lmaxnrv-1
2695             neutprime(l)=k*denl(l
2696         end do
2697
2698         !  Truncate the hierarchy
2699         neutprime(EV%lmaxnrv)=k*E
```

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

```
2399            - pb43/(1+pb43)/o
2400            ( 2*pb43*adotoa**
2401            )/(1+pb43)
2402
2403            !  Equation for the p
2404            ! Get drag from vbdot
2405            yvprime(4)=-0.5_dl*k*
2406               (vbdot+adotoa*vb)
2407
2408            !  Set the derivative
2409            yvprime(5:n)=0._dl
2410            yv(5)=pig
2411            E(2)=  pig/4
2412         endif
2413
2414         yvprime(3) = vbdot
2415
2416         !  Neutrino equations:
2417
2418         !  Massless neutrino anis
2419         pir=neut(2)
2420         neutprime(1)= -0.5_dl*k*p
2421         neutprime(2)=2._dl/5*k*qr
2422         !  And for the moments
2423         do  l=3,EV%lmaxnrv-1
2424             neutprime(l)=k*denl(l
2425         end do
2426
2427         !  Truncate the hierarchy
2428         neutprime(EV%lmaxnrv)=k*E
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2700

```
2700        (EV%lmaxnrv+2._dl)*neut(E
2701
2702
2703        !  Get the propagation eq
2704
2705        rhopi=grhog_t*pig+grhor_t
2706
2707        yvprime(2)=-2*adotoa*sigm
2708
2709        end subroutine derivsv
2710
2711
2712
2713        !cccccccccccccccccccccccc
2714        subroutine derivst(EV,n,t
2715        !  Evaluate the time deri
2716        use ThermoData
2717        use MassiveNu
2718        implicit none
2719        type(EvolutionVars) EV
2720        integer n,l,i,ind, nu_i
2721        real(dl), target ::  ayt(
2722        real(dl) tau,grho,rhopi,c
2723        real(dl), dimension(:),po
2724        real(dl) q,aq,v
2725        real(dl)  grhob_t,grhor_t
2726        real(dl) Hchi,pinu, pig
2727        real(dl) k,k2,a,a2
2728        real(dl) pir, adotoa, rho
2729
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2429

```
2429        (EV%lmaxnrv+2._dl)*ne
2430
2431
2432        !  Get the propagation eq
2433
2434        rhopi=grhog_t*pig+grhor_t
2435
2436        yvprime(2)=-2*adotoa*sigm
2437
2438        end subroutine derivsv
2439
2440
2441
2442        !cccccccccccccccccccccccc
2443        subroutine derivst(EV,n,t
2444        !  Evaluate the time deri
2445        use ThermoData
2446        use MassiveNu
2447        implicit none
2448        type(EvolutionVars) EV
2449        integer n,l,i,ind, nu_i
2450        real(dl), target ::  ayt(
2451        real(dl) tau,grho,rhopi,c
2452        real(dl), dimension(:),po
2453        real(dl) q,aq,v
2454        real(dl)  grhob_t,grhor_t
2455        real(dl) Hchi,pinu, pig
2456        real(dl) k,k2,a,a2
2457        real(dl) pir, adotoa, rho
2458
```

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

```
2730          real(dl) cothxor
2731
2732          k2=EV%k2_buf
2733          k= EV%k_buf
2734
2735          a=ayt(1)
2736
2737          Hchi=ayt(2)
2738
2739          shear=ayt(3)
2740
2741          a2=a*a
2742
2743          ! Compute expansion rate
2744          ! Also calculate gpres: 8
2745          grhob_t=grhob/a
2746          grhoc_t=grhoc/a
2747          grhor_t=grhornomass/a2
2748          grhog_t=grhog/a2
2749
2750       !#SimDataReplace
2751       !     if (w_lam==-1._dl) then
2752       !         grhov_t=grhov*a2
2753       !     else
2754       !         grhov_t=grhov*a**(-1
```

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

```
2459          real(dl) cothxor
2460
2461          k2=EV%k2_buf
2462          k= EV%k_buf
2463
2464          a=ayt(1)
2465
2466          Hchi=ayt(2)
2467
2468          shear=ayt(3)
2469
2470          a2=a*a
2471
2472          ! Compute expansion rate
2473          ! Also calculate gpres: 8
2474          grhob_t=grhob/a
2475          grhoc_t=grhoc/a
2476          grhor_t=grhornomass/a2
2477          grhog_t=grhog/a2
2478          if (w_lam==-1._dl) then
2479              grhov_t=grhov*a2
2480          else
2481              grhov_t=grhov*a**(-1-
2482          end if
2483
```

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

```
2755 !     end if
2756  grhov_t=grho_de(a)/a2
2757 !#SimDataReplace
2758       grho=grhob_t+grhoc_t+grho
2759
2760       !Do massive neutrinos
2761       if (CP%Num_Nu_Massive >0)
2762           do nu_i=1,CP%Nu_mass_
2763               call Nu_rho(a*nu_
2764               grho=grho+grhorma
2765           end do
2766       end if
2767
2768       if (CP%flat) then
2769           cothxor=1._dl/tau
2770           adotoa=sqrt(grho/3._d
2771       else
2772           cothxor=1._dl/tanfunc
2773           adotoa=sqrt((grho+grh
2774       end if
2775
2776       aytprime(1)=adotoa*a
2777
2778       call thermo(tau,cs2,opaci
2779
2780       if (.not. EV%TensTightCou
2781           !  Don't use tight co
2782           !  Equation for the p
2783
2784
```

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

```
2484       grho=grhob_t+grhoc_t+grho
2485
2486       !Do massive neutrinos
2487       if (CP%Num_Nu_Massive >0)
2488           do nu_i=1,CP%Nu_mass_
2489               call Nu_rho(a*nu_
2490               grho=grho+grhorma
2491           end do
2492       end if
2493
2494       if (CP%flat) then
2495           cothxor=1._dl/tau
2496           adotoa=sqrt(grho/3._d
2497       else
2498           cothxor=1._dl/tanfunc
2499           adotoa=sqrt((grho+grh
2500       end if
2501
2502       aytprime(1)=adotoa*a
2503
2504       call thermo(tau,cs2,opaci
2505
2506       if (.not. EV%TensTightCou
2507           !  Don't use tight co
2508           !  Equation for the p
2509
2510
```

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

```
2785              !E and B start at l=2
2786              E => ayt(EV%E_ix+1:)
2787              B => ayt(EV%B_ix+1:)
2788              Eprime=> aytprime(EV%
2789              Bprime => aytprime(EV
2790
2791              ind = EV%g_ix+2
2792
2793              !  Photon anisotropic
2794              pig=ayt(ind)
2795              polter = 0.1_dl*pig +
2796
2797              if (EV%lmaxt > 2) the
2798                  aytprime(ind)=-EV
2799                  -opacity*(pig - p
2800
2801                  do l=3, EV%lmaxt
2802                      ind = ind+1
2803                      aytprime(ind)
2804                  end do
2805
2806                  !Truncate the hie
2807                  ind=ind+1
2808                  aytprime(ind)=k*E
2809                  (EV%lmaxt+3._dl)*
2810
2811                  !E and B-bar equa
2812
2813                  Eprime(2) = - opa
2814                  EV%denlkt(3,2)*E(
```

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

```
2511              !E and B start at l=2
2512              E => ayt(EV%E_ix+1:)
2513              B => ayt(EV%B_ix+1:)
2514              Eprime=> aytprime(EV%
2515              Bprime => aytprime(EV
2516
2517              ind = EV%g_ix+2
2518
2519              !  Photon anisotropic
2520              pig=ayt(ind)
2521              polter = 0.1_dl*pig +
2522
2523              if (EV%lmaxt > 2) the
2524                  aytprime(ind)=-EV
2525                      -opacity*(pig
2526
2527                  do l=3, EV%lmaxt
2528                      ind = ind+1
2529                      aytprime(ind)
2530                  end do
2531
2532                  !Truncate the hie
2533                  ind=ind+1
2534                  aytprime(ind)=k*E
2535                      (EV%lmaxt+3._
2536
2537                  !E and B-bar equa
2538
2539                  Eprime(2) = - opa
2540                      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
```

/Users/lp1opa/Compare/camb_simdata/equations.f90, Top line: 2845

```
2845
2846
2847          !  Neutrino equations:
2848          !  Anisotropic stress
2849          if (DoTensorNeutrinos) th
2850           neutprime => aytprime
2851           neut => ayt(EV%r_ix+1
2852
2853           !  Massless neutrino
2854           pir=neut(2)
2855
2856           rhopi=rhopi+grhor_t*p
2857
2858           if (EV%lmaxnrt>2) the
2859            pirdt=-EV%denlkt(
2860            neutprime(2)=pird
2861           !  And for the mom
2862            do  l=3, EV%lmaxn
2863             neutprime(l)=
2864            end do
2865
2866            !  Truncate the h
2867            neutprime(EV%lmax
2868            (EV%lmaxnrt+3._dl
2869           else
2870            pirdt= 8._dl/15._
2871            neutprime(2)=pird
2872           end if
2873
2874           !  Massive neutrino e
```

/Users/lp1opa/Compare/camb_des/equations.f90, Top line: 2571

```
2571
2572
2573          !  Neutrino equations:
2574          !  Anisotropic stress
2575          if (DoTensorNeutrinos) th
2576           neutprime => aytprime
2577           neut => ayt(EV%r_ix+1
2578
2579           !  Massless neutrino
2580           pir=neut(2)
2581
2582           rhopi=rhopi+grhor_t*p
2583
2584           if (EV%lmaxnrt>2) the
2585            pirdt=-EV%denlkt(
2586            neutprime(2)=pird
2587           !  And for the mo
2588            do  l=3, EV%lmaxn
2589             neutprime(l)=
2590            end do
2591
2592            !  Truncate the h
2593            neutprime(EV%lmax
2594             (EV%lmaxnrt+3
2595           else
2596            pirdt= 8._dl/15._
2597            neutprime(2)=pird
2598           end if
2599
2600           !  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

```
2905 |            end if
2906 |        end if
2907 |
2908 |        !  Get the propagation eq
2909 |
2910 |        if (CP%flat) then
2911 |            aytprime(3)=-2*adotoa
2912 |        else
2913 |            aytprime(3)=-2*adotoa
2914 |        endif
2915 |
2916 |        aytprime(2)=-k*shear
2917 |
2918 |        end subroutine derivst
2919 |
2920 |
2921 |
2922 |        !cccccccccccccccccccccccc
2923 |
2924 |        end module GaugeInterface
2925 |
```

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

```
2631 |            end if
2632 |        end if
2633 |
2634 |        !  Get the propagation eq
2635 |
2636 |        if (CP%flat) then
2637 |            aytprime(3)=-2*adotoa
2638 |        else
2639 |            aytprime(3)=-2*adotoa
2640 |        endif
2641 |
2642 |        aytprime(2)=-k*shear
2643 |
2644 |        end subroutine derivst
2645 |
2646 |
2647 |
2648 |        !cccccccccccccccccccccccc
2649 |
2650 |        end module GaugeInterface
2651 |
```