

```
/Users/lp1opa/Compare/camb_simdata/halo  
fit_ppf.f90, Top line: 1
```

```
001 | !simdata ana : attentie, aici,  
002 | ! %%%%%%%%%%%%%%  
003 | ! The `halofit' code model  
004 | ! cosmological power spect  
005 | ! this is done are present  
006 | !  
007 | ! The code `halofit' was w  
008 | ! See http://www.astro.up  
009 | !  
010 | ! Subsequent updates as be  
011 | ! Only tested for basic mo  
  
012 |  
013 | ! Adapted for F90 and CAMB  
014 | !!BR09 Oct 09: generalized  
015 |  
016 | ! RT12 Oct: update some fi  
017 | ! the power spec  
018 |  
019 | !!JD 08/13: generalized ex  
020 | ! w_0 and w_a  
021 | ! SPB14 Feb: update the fi  
022 | ! modifications  
023 | ! AL Sept 14: added halofi
```

```
/Users/lp1opa/Compare/camb_des/halofit_  
ppf.f90, Top line: 1
```

```
0001 | ! %%%%%%%%%%%%%%  
0002 | ! The `halofit' code mode  
0003 | ! cosmological power spec  
0004 | ! this is done are presen  
0005 | !  
0006 | ! The code `halofit' was  
0007 | ! See http://www.astro.up  
0008 | !  
0009 | ! Subsequent updates as b  
0010 | ! Only tested for basic m  
0011 | ! References for variant  
0012 | ! halofit_original: ast  
0013 | ! halofit_peacock: http  
0014 | ! halofit_bird: arXiv:  
0015 | ! halofit_takahashi: ar  
0016 | ! halofit_mead: arXiv:1  
0017 | ! halofit_casarini: arX  
0018 |  
0019 | ! Adapted for F90 and CAM  
0020 | !!BR09 Oct 09: generalize  
0021 |  
0022 | ! RT12 Oct: update some f  
0023 | ! the power spe  
0024 |  
0025 | !!JD 08/13: generalized e  
0026 | ! w_0 and w_a  
0027 | ! SPB14 Feb: update the f  
0028 | ! modifications  
0029 | ! AL Sept 14: added halof
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 24

```
024      ! separate halofit.f90 i
025      ! Jan 15: Suggested change
026
027      ! %%%%%%%%%%%%%%
028
029      module NonLinear
030      use ModelParams
031      use transfer
032      use LambdaGeneral
033      implicit none
034      private
035
036      real, parameter :: Min_kh_
037      real(dl):: om_m,om_v,fnu,o
038
039      integer, parameter :: halo
040
041      integer, parameter :: halo
042      integer :: halofit_version
043      public Min_kh_nonlinear,No
          public halofit_version,hal
```

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 30

```
0030      ! separate halofit.f90
0031      ! Jan 15: Suggested chang
0032      !AM Mar 16: Added in HMco
0033      !AM May 16: Fixed some sm
0034      !AL Jun16: put in partial
0035      !AM Sep 16: Attempted fix
0036      !LC Oct 16: extended Halo
0037      !AM May 17: Made the bary
0038      !AL Jul 17: fixed undefin
0039      !AM Jul 17: sped-up HMcod
0040
0041
0042      ! %%%%%%%%%%%%%%
0043
0044      module NonLinear
0045      use ModelParams
0046      use transfer
0047      use LambdaGeneral
0048      implicit none
0049      private
0050
0051      real, parameter :: Min_kh_
0052      real(dl):: om_m,om_v,fnu,
0053
0054      integer, parameter :: halo
0055      integer, parameter :: halo
0056      integer, parameter :: halo
0057      integer :: halofit_version
0058      public Min_kh_nonlinear,No
          public halofit_version,hal
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 44

044 contains  
045  
046 subroutine NonLinear\_ReadP  
047 use IniFile

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 60

0060 public halofit\_mead, halo  
0061  
0062 !!AM - Added these types  
0063 INTEGER :: imead !!AM - a  
0064 logical :: HM\_verbose = .  
0065  
0066 TYPE HM\_cosmology  
0067 !Contains only things  
0068 REAL :: om\_m, om\_v, w  
0069 REAL, ALLOCATABLE ::  
0070 REAL, ALLOCATABLE ::  
0071 REAL, ALLOCATABLE ::  
0072 INTEGER :: nk, ng, ns  
0073 !AM - Added feedback  
0074 REAL :: A\_baryon=3.13  
0075 REAL :: eta\_baryon=0.  
0076 END TYPE HM\_cosmology  
0077  
0078 TYPE HM\_tables  
0079 !Stuff that needs to  
0080 REAL, ALLOCATABLE ::  
0081 REAL :: sigv, sigv100  
0082 INTEGER :: n  
0083  
0084 END TYPE HM\_tables  
0085 !!AM - End of my addition  
0086 contains  
0087  
0088 subroutine NonLinear\_Read  
0089 use IniFile

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 48

```
048      Type(TIniFile) :: Ini
049
050      halofit_version = Ini_Read
051
052      end subroutine NonLinear_R
053
054      subroutine NonLinear_GetNo
055      !Fill the CAMB_Pk%nonlin_s
056      !for each redshift and wav
057      !This implementation uses
058      type(MatterPowerData) :: C
059      integer itf
060      real(dl) a,plin,pq,ph,pnl,
061      real(dl) sig,rknl,rneff,rn
062      real(dl) diff,xlogr1,xlogr
063      integer i
064
065      ! !BR09 putting neutrinos in
066      omm0 = CP%omegac+CP%omegab
067      fnu = CP%omegan/omm0
068
069      CAMB_Pk%nonlin_ratio = 1
070
```

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 90

```
0090      Type(TIniFile) :: Ini
0091
0092      halofit_version = Ini_Rea
0093
0094      end subroutine NonLinear_
0095
0096      subroutine NonLinear_GetN
0097      !Fill the CAMB_Pk%nonlin_
0098      !for each redshift and wa
0099      !This implementation uses
0100      type(MatterPowerData) :::
0101      integer itf
0102      real(dl) a,plin,pq,ph,pnl
0103      real(dl) sig,rknl,rneff,r
0104      real(dl) diff,xlogr1,xlog
0105      integer i
0106
0107      IF(halofit_version==halof
0108
0109      !AM - Call HMcode her
0110      CALL HMcode(CAMB_Pk)
0111
0112      ELSE
0113
0114      ! !BR09 putting neutri
0115      omm0 = CP%omegac+CP%o
0116      fnu = CP%omegan/omm0
0117
0118      CAMB_Pk%nonlin_ratio
0119
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 71

```
071 do itf = 1, CAMB_Pk%num_z
072
073      ! calculate nonlinear
074      ! curvature (rncur) of
075      ! described in Smith e
076      a = 1/real(1+CAMB_Pk%R
077
078      !SimDataReplace
079      !
080      om_m = omega_m(a, omm
081      !
082      om_m = omega_m(a, omm0,
083      om_v = omega_v(a, omm0,
084      !
085      !#SimDataReplace
086
087      acur = a
088      xlogr1=-2.0
089      xlogr2=3.5
090      do
091          rmid=(xlogr2+xlogr
092          rmid=10**rmid
093          call wint(CAMB_Pk,
```

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 120

```
0120 do itf = 1, CAMB_Pk%n
0121
0122      w_hf=w_lam
0123      wa_hf=wa_ppf
0124      if (halofit_versi
0125          ! calculate e
0126          ! [Casarini+
0127          call PKequal(
0128      endif
0129
0130      ! calculate nonli
0131      ! curvature (rncu
0132      ! described in Sm
0133      a = 1/real(1+CAMB
0134      om_m = omega_m(a,
0135      om_v = omega_v(a,
0136
0137
0138
0139
0140
0141
0142      acur = a
0143      xlogr1=-2.0
0144      xlogr2=3.5
0145      do
0146          rmid=(xlogr2+
0147          rmid=10**rmid
0148          call wint(CAM
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 94

```
094      diff=sig-1.0
095      if (abs(diff).le.0
096          rkn1=1./rmid
097          rneff=-3-d1
098          rncur=-d2
099          exit
100      elseif (diff.gt.0.
101          xlogr1=log10(r
102      elseif (diff.lt.-0
103          xlogr2=log10(r
104      endif
105      if (xlogr2 < -1.99
106          !is still line
107          goto 101
108      else if (xlogr2>3.
109          ! Totally craz
110          global_error_f
111          write(*,*) 'Er
112          goto 101
113      end if
114      end do
115
116      ! now calculate power
117
118      do i=1, CAMB_PK%num_k
119          rk = exp(CAMB_Pk%l
120
121          if (rk > Min_kh_no
122              ! linear power
123
```

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 143

```
0143      diff=sig-1.0
0144      if (abs(diff)
0145          rkn1=1./r
0146          rneff=-3-
0147          rncur=-d2
0148          exit
0149      elseif (diff.
0150          xlogr1=lo
0151      elseif (diff.
0152          xlogr2=lo
0153      endif
0154      if (xlogr2 < -1.99
0155          !is still line
0156          goto 101
0157      else if (xlogr2>3.
0158          ! Totally craz
0159          global_error_f
0160          write(*,*) 'Er
0161          goto 101
0162      end if
0163      end do
0164
0165      ! now calculate p
0166
0167      do i=1, CAMB_PK%n
0168          rk = exp(CAMB_Pk%
0169
0170          if (rk > Min_
0171              ! linear
0172
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 124

```
124      ! constant = 4
125
126      plin= MatterPo
127
128      ! calculate no
129      ! where pq rep
130      ! where ph is
131
132      call halofit(r
133          CAMB_Pk%nonlin
134
135      end if
136
137      enddo
138
139 101    continue
140    end do
141
142    end subroutine NonLinear_G
143
144    ! % % % % % % % %
145
146
147    subroutine halofit(rk,rn,r
148        implicit none
149
150        real(dl) gam,a,b,c,xmu,xnu
151        real(dl) rk,rn,plin,pnl,pq
```

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 173

```
0173      ! constan
0174
0175      plin= Mat
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191      END IF
0192
0193
0194
0195
0196
0197
0198
0199
0200
0201      real(dl) gam,a,b,c,xmu,xn
              real(dl) rk,rn,plin,pnl,p
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 152

```
152      real(dl) rkn1,y,rncur
153      real(dl) f1a,f2a,f3a,f1b,f
154      real(dl) extragam, peacock
155
156      if (halofit_version ==halo
157          .or. halofit_version =
158              ! halo model nonlinear fit
159              ! Appendix C of Smith et a
160              !SPB11: Standard halofit u
161              !factor of two. Add an ext
162              !Haehnelt 2011 which parti
163      if (halofit_version ==halo
164          extragam = 0.3159 -0.0
165          gam=extragam+0.86485+0
166      else
167          gam=0.86485+0.2989*rn+
168      end if
169      a=1.4861+1.83693*rn+1.6761
170          0.1670756*rn*rn*rn*rn-
171      a=10**a
172      b=10*** (0.9463+0.9466*rn+0.
173      c=10*** (-0.2807+0.6669*rn+0
174      xmu=10*** (-3.54419+0.19086*
175      xnu=10*** (0.95897+1.2857*rn
176      alpha=1.38848+0.3701*rn-0.
177      beta=0.8291+0.9854*rn+0.34
178      elseif (halofit_version ==
179          !RT12 Oct: the halofit
180          !than latest N-body si
181          !Update the following
```

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 202

```
0202      real(dl) rkn1,y,rncur
0203      real(dl) f1a,f2a,f3a,f1b,
0204      real(dl) extragam, peacoc
0205
0206      if (halofit_version ==halo
0207          .or. halofit_version =
0208              ! halo model nonlinear
0209              ! Appendix C of Smith
0210              !SPB11: Standard halo
0211              !factor of two. Add a
0212              !Haehnelt 2011 which
0213      if (halofit_version =
0214          extragam = 0.3159
0215          gam=extragam+0.86
0216      else
0217          gam=0.86485+0.298
0218      end if
0219      a=1.4861+1.83693*rn+
0220          0.1670756*rn*rn*rn*rn-
0221      a=10**a
0222      b=10*** (0.9463+0.9466*
0223      c=10*** (-0.2807+0.6669*
0224      xmu=10*** (-3.54419+0.1
0225      xnu=10*** (0.95897+1.28
0226      alpha=1.38848+0.3701*
0227      beta=0.8291+0.9854*rn
0228      elseif (halofit_version =
0229          !RT12 Oct: the halofit
0230          !than latest N-body s
0231          !Update the following
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 182

```
182      !alpha & beta from the
183      !The improved halofit
184      !cosmological models w
185      gam=0.1971-0.0843*rn+
186
187      !#SimDataReplace
188
189      !
190      !      a=1.5222+2.8553*rn+2.
191
192      a=1.5222+2.8553*rn+2.3
193      0.2250*rn*rn*rn*rn
194
195
196      !#SimDataReplace
197
198      a=10**a
199
200      !#SimDataReplace
201      !
202      !      b=10**(-0.5642+0.5864
203
204      b=10**(-0.5642+0.5864*
205      0.2279*om_v*(1.+CP
206      !!! atentie ca wa_ppf sa fie 0
207      !print*, 'In halofit_ppf wa_pp
208      !#SimDataReplace
209      c=10** (0.3698+2.0404*r
210      xmu=0.
```

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 232

```
0232      !alpha & beta from th
0233      !The improved halofit
0234      !cosmological models
0235      !LC16 Jun: Casarini+
0236      gam=0.1971-0.0843*rn+
0237
0238
0239
0240
0241
0242
0243
a=1.5222+2.8553*rn+2.
0.2250*rn*rn*rn*rn
a=10**a
b=10** (-0.5642+0.5864
0.2279*om_v*(1.+w
c=10** (0.3698+2.0404*
xmu=0.
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 211

```
211      xnu=10** (5.2105+3.6902
212      alpha=abs(6.0835+1.337
213      beta=2.0379-0.7354*rn+
214          0.3980*rn**4-0.168
215      else
216          stop 'Unknown halofit_
217      end if
218
219      if(abs(1-om_m).gt.0.01) th
220          f1a=om_m** (-0.0732)
221          f2a=om_m** (-0.1423)
222          f3a=om_m** (0.0725)
223          f1b=om_m** (-0.0307)
224          f2b=om_m** (-0.0585)
225          f3b=om_m** (0.0743)
226          frac=om_v/(1.-om_m)
227          f1=frac*f1b + (1-frac)
228          f2=frac*f2b + (1-frac)
229          f3=frac*f3b + (1-frac)
230      else
231          f1=1.0
232          f2=1.
233          f3=1.
234      endif
235
236      y=(rk/rkn1)
237
238
239      ph=a*y** (f1*3)/(1+b*y** (f2
240      ph=ph/(1+xmu*y** (-1)+xnu*y
```

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 244

```
0244      xnu=10** (5.2105+3.690
0245      alpha=abs(6.0835+1.33
0246      beta=2.0379-0.7354*rn
0247          0.3980*rn**4-0.16
0248      else
0249          call MPIStop('Unknown
0250      end if
0251
0252      if(abs(1-om_m).gt.0.01) t
0253          f1a=om_m** (-0.0732)
0254          f2a=om_m** (-0.1423)
0255          f3a=om_m** (0.0725)
0256          f1b=om_m** (-0.0307)
0257          f2b=om_m** (-0.0585)
0258          f3b=om_m** (0.0743)
0259          frac=om_v/(1.-om_m)
0260          f1=frac*f1b + (1-frac)
0261          f2=frac*f2b + (1-frac)
0262          f3=frac*f3b + (1-frac
0263      else
0264          f1=1.0
0265          f2=1.
0266          f3=1.
0267      endif
0268
0269      y=(rk/rkn1)
0270
0271
0272      ph=a*y** (f1*3)/(1+b*y** (f
0273      ph=ph/(1+xmu*y** (-1)+xnu*
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 241

```
241      plinaa=plin*(1+fnu*47.48*r
242      pq=plin*(1+plinaa)**beta/(
243
244      pnl=pq+ph
245
246      if (halofit_version == hal
247          !From http://www.roe.a
248          !(P-P_linear) -> (P-P_
249          peacock_fudge = rk/10
250          pnl = plin + (pn1-plin
251      end if
252
253      end subroutine halofit
254
255
256      ! % % % % % % % % % % % %
257
258      ! The subroutine wint, fin
259      ! rkn1, rneff & rncur. Thi
260      ! the Gaussian filter at w
261      ! rneff is defined as the
262      ! at the nonlinear wavenum
263      ! derivative at the nonlin
264
265      subroutine wint(CAMB_Pk,it
266      implicit none
267      integer, intent(in) :: itf
268      type(MatterPowerData) :: C
269      real(dl) sum1,sum2,sum3,t,
270      real(dl) x2,rk, fac,r, sig
```

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 274

```
0274      plinaa=plin*(1+fnu*47.48*
0275      pq=plin*(1+plinaa)**beta/
0276
0277      pnl=pq+ph
0278
0279      if (halofit_version == ha
0280          !From http://www.roe.
0281          !(P-P_linear) -> (P-P_
0282          peacock_fudge = rk/10
0283          pnl = plin + (pn1-pli
0284      end if
0285
0286      end subroutine halofit
0287
0288
0289      ! % % % % % % % % % % % %
0290
0291      ! The subroutine wint, fi
0292      ! rkn1, rneff & rncur. Th
0293      ! the Gaussian filter at
0294      ! rneff is defined as the
0295      ! at the nonlinear wavenu
0296      ! derivative at the nonli
0297
0298      subroutine wint(CAMB_Pk,i
0299
0300      integer, intent(in) :: itf
0301      type(MatterPowerData) :: C
0302      real(dl) sum1,sum2,sum3,t,
0303      real(dl) x2,rk, fac,r, si
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 271

```
271      integer i,nint
272
273      nint=3000
274      sum1=0.d0
275      sum2=0.d0
276      sum3=0.d0
277      anorm = 1/(2*pi**2)
278      do i=1,nint
279          t=(i-0.5_d1)/nint
280          y=-1.d0+1.d0/t
281          rk=y
282          d2=MatterPowerData_k(C)
283          x=y*r
284          x2=x*x
285          w1=exp(-x2)
286          w2=2*x2*w1
287          w3=4*x2*(1-x2)*w1
288          fac=d2/y/t/t
289          sum1=sum1+w1*fac
290          sum2=sum2+w2*fac
291          sum3=sum3+w3*fac
292      enddo
293      sum1=sum1/nint
294      sum2=sum2/nint
295      sum3=sum3/nint
296      sig=sqrt(sum1)
297      d1=-sum2/sum1
298      d2=-sum2*sum2/sum1/sum1 -
299
300      end subroutine wint
```

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 303

```
0303      integer i,nint
0304
0305      nint=3000
0306      sum1=0.d0
0307      sum2=0.d0
0308      sum3=0.d0
0309      anorm = 1/(2*pi**2)
0310      do i=1,nint
0311          t=(i-0.5_d1)/nint
0312          y=-1.d0+1.d0/t
0313          rk=y
0314          d2=MatterPowerData_k(
0315          x=y*r
0316          x2=x*x
0317          w1=exp(-x2)
0318          w2=2*x2*w1
0319          w3=4*x2*(1-x2)*w1
0320          fac=d2/y/t/t
0321          sum1=sum1+w1*fac
0322          sum2=sum2+w2*fac
0323          sum3=sum3+w3*fac
0324      enddo
0325      sum1=sum1/nint
0326      sum2=sum2/nint
0327      sum3=sum3/nint
0328      sig=sqrt(sum1)
0329      d1=-sum2/sum1
0330      d2=-sum2*sum2/sum1/sum1 -
0331
0332      end subroutine wint
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 301

```
301
302      !!JD 08/13 generalize to v
303      !      #Simdata : aproximativ ac
304
305      !      function omega_m(aa,om_m0
306      !      implicit none
307      !      real(dl) omega_m,omega_t,
308      !      Qa2= aa**(-1.0-3.0*(wval+
309      !      omega_t=1.0+(om_m0+om_v0-
310      !      omega_m=omega_t*om_m0/(om_
311      !      end function omega_m

312
313      function omega_m(aa,om_m0
314      implicit none
315      real(dl) omega_m,omega_t,
316      v_fac= aa**(-1.0-3.0*(w0v
317      omega_t=1.0+(om_m0+om_v0-
318      omega_m=omega_t*om_m0/(om_
319      end function omega_m

320
321      !%%%%%%%%%%%%%
322
323      ! evolution of omega lambd
324
325      !      function omega_v(aa,om_m0
326      !      implicit none
327      !      real(dl) aa,omega_v,om_m0
328      !      Qa2= aa**(-1.0-3.0*(wval+
329      !      omega_t=1.0+(om_m0+om_v0-
330      !      omega_v=omega_t*om_v0*Qa2
```

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 333

```
0333
0334      !!JD 08/13 generalize to
0335
0336      function omega_m(aa,om_m0
0337      real(dl) omega_m,omega_t,
0338      Qa2= aa**(-1.0-3.0*(wval+
0339      omega_t=1.0+(om_m0+om_v0-
0340      omega_m=omega_t*om_m0/(om_
0341      end function omega_m

0342
0343
0344
0345
0346
0347
0348
0349
0350
0351      !%%%%%%%%%%%%%
0352
0353      ! evolution of omega lambd
0354
0355      function omega_v(aa,om_m0
0356      real(dl) aa,omega_v,om_m0
0357      Qa2= aa**(-1.0-3.0*(wval+
0358      omega_t=1.0+(om_m0+om_v0-
0359      omega_v=omega_t*om_v0*Qa2
```

```
/Users/lp1opa/Compare/camb_simdata/halo  
fit_ppf.f90, Top line: 331
```

```
331      !    end function omega_v  
332  
333      function omega_v(aa,om_m  
334      implicit none  
335      real(dl) aa,omega_v,om_m  
336      v_fac = aa**(-1.d0-3.d0*  
337      omega_t=1.0+(om_m0+om_v0  
338      omega_v=omega_t*om_v0*v_  
339      end function omega_v  
340      !#SimDataReplace  
341      !!JD end generalize to var  
342
```

```
/Users/lp1opa/Compare/camb_des/halofit_  
ppf.f90, Top line: 352
```

```
0352      end function omega_v  
0353  
0354  
0355  
0356      !!AM Below is for HMcode  
0357      SUBROUTINE HMcode(CAMB_Pk  
0358      !!AM - A CAMB derived typ  
0359      TYPE(MatterPowerData) ::  
0360      REAL :: z, k  
0361      REAL :: p1h, p2h, pfull,  
0362      INTEGER :: i, j, nk, nz  
0363      TYPE(HM_cosmology) :: cos  
0364      TYPE(HM_tables) :: lut  
0365      REAL, PARAMETER :: pi=3.1  
0366  
0367      !HMcode developed by Alex  
0368      !Please contact me if you  
0369      !If you use this in your  
0370      !If you use the extension  
0371      !Also consider citing the  
0372  
0373      !Use imead to switch betw
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 374

```
0374      !0 - Standard (this is ju
0375      !1 - Accurate from Mead e
0376      !2 - Accurate from Mead e
0377      IF(halofit_version==halof
0378      IF(halofit_version==halof
0379      IF(halofit_version==halof
0380
0381      IF(FeedbackLevel>0) HM_ve
0382
0383      IF(HM_verbose) WRITE(*,*) 
0384      IF(HM_verbose) WRITE(*,*) 
0385      IF(HM_verbose) WRITE(*,*) 
0386
0387      !!AM - Translate from CAM
0388      nz=CAMB_PK%num_z
0389      nk=CAMB_PK%num_k
0390
0391      !!AM - Assign cosmological
0392      CALL assign_HM_cosmology(
0393
0394      !Fill growth function tab
0395      CALL fill_growtab(cosi)
0396
0397      !Loop over redshifts
0398      DO j=1,nz
0399
0400      !Initialise the speci
0401      !Currently this needs
0402      !For non-massive-neut
0403      CALL initialise_HM_co
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 404

```
0404 !Sets the current red
0405 z=CAMB_Pk%Redshifts(j)
0406
0407 !Initialisation for t
0408 CALL halomod_init(z,l)
0409
0410
0411 !Loop over k values a
0412 !$OMP PARALLEL DO DEF
0413 DO i=1,nk
0414   k=exp(CAMB_Pk%log
0415   plin=p_lin(k,z,0,
0416   CALL halomod(k,z,
0417   CAMB_Pk%nonlin_ra
0418 END DO
0419 !$OMP END PARALLEL DO
0420
0421 END DO
0422
0423 END SUBROUTINE HMcode
0424
0425 FUNCTION Delta_v(z,lut,co
0426
0427 !Function for the viriali
0428 REAL :: Delta_v
0429 REAL, INTENT(IN) :: z
0430 TYPE(HM_cosmology), INTEN
0431 TYPE(HM_tables), INTENT(I
0432
0433 IF(imead==0) THEN
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 434

```
0434 !Value that is normal
0435 Delta_v=200.
0436 ELSE IF(imead==1 .OR. ime
0437 !Mead et al. (2015; a
0438 Delta_v=418.* (Omega_m
0439 !Mead et al. (2016; a
0440 IF(imead==1) Delta_v=
0441 END IF
0442
0443
0444
0445 FUNCTION delta_c(z,lut,co
0446
0447 !Function for the linear
0448 REAL :: delta_c
0449 REAL, INTENT(IN) :: z
0450 TYPE(HM_cosmology), INTEN
0451 TYPE(HM_tables), INTEN(I
0452
0453 IF(imead==0) THEN
0454     delta_c=1.686
0455 ELSE IF(imead==1 .OR. ime
0456 !Mead et al. (2015; a
0457 delta_c=1.59+0.0314*1
0458 IF(imead==1) THEN
0459     delta_c=delta_c*( )
0460     delta_c=delta_c*( )
0461 END IF
0462
0463 END IF
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 464

```
0464      END FUNCTION delta_c
0465
0466      FUNCTION eta(z,lut,cosm)
0467
0468      !Function eta that puffs
0469      REAL :: eta
0470      REAL, INTENT(IN) :: z
0471      TYPE(HM_cosmology), INTEN
0472      TYPE(HM_tables), INTENT(I
0473      REAL :: eta0
0474
0475      IF(imead==0) THEN
0476          eta=0.
0477      ELSE IF(imead==1 .OR. ime
0478          !The first parameter
0479          !eta=0.603-0.3*lut%si
0480          !AM - made baryon fee
0481          eta0=cosm%eta_baryon
0482          !eta0=0.98-0.12*cosm%
0483          eta=eta0-0.3*lut%sig8
0484      END IF
0485
0486      END FUNCTION eta
0487
0488      FUNCTION kstar(z,lut,cosm)
0489
0490      !Function k* that cuts off
0491      REAL :: kstar
0492      REAL, INTENT(IN) :: z
0493      TYPE(HM_cosmology), INTEN
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 494

```
0494      TYPE(HM_tables), INTENT(I
0495
0496      IF(imead==0) THEN
0497          !Set to zero for the
0498          kstar=0.
0499      ELSE IF(imead==1 .OR. ime
0500          !One-halo cut-off wav
0501          !Mead et al. (2015; a
0502          kstar=0.584*(lut%sigv
0503      END IF
0504
0505      END FUNCTION kstar
0506
0507      FUNCTION As(z,lut,cosm)
0508
0509          !Halo concentration pre-f
0510          REAL :: As
0511          REAL, INTENT(IN) :: z
0512          TYPE(HM_cosmology), INTEN
0513          TYPE(HM_tables), INTENT(I
0514
0515          IF(imead==0) THEN
0516              !Set to 4 for the sta
0517              As=4.
0518          ELSE IF(imead==1 .OR. ime
0519              !This is the 'A' halo
0520              !As=3.13
0521              !AM - added for easy
0522              As=cosm%A_baryon
0523      END IF
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 524

```
0524  
0525  
0526  
0527  
0528  
0529  
0530  
0531  
0532  
0533  
0534  
0535  
0536  
0537  
0538  
0539  
0540  
0541  
0542  
0543  
0544  
0545  
0546  
0547  
0548  
0549  
0550  
0551  
0552  
0553
```

```
END FUNCTION As
```

```
FUNCTION fdamp(z,lut,cosm
```

```
!Linear power damping fun  
REAL :: fdamp  
REAL, INTENT(IN) :: z  
TYPE(HM_cosmology), INTEN  
TYPE(HM_tables), INTENT(I
```

```
!Linear theory damping fa  
IF(imead==0) THEN  
    !Set to 0 for the sta  
    fdamp=0.  
ELSE IF(imead==1) THEN  
    !Mead et al. (2016; a  
    fdamp=0.0095*lut%sigv  
ELSE IF(imead==2) THEN  
    !Mead et al. (2015) v  
    fdamp=0.188*lut%sig8z  
END IF
```

```
!Catches extreme values o  
IF(fdamp<1.e-3) fdamp=1.e  
IF(fdamp>0.99) fdamp=0.9
```

```
END FUNCTION fdamp
```

```
FUNCTION alpha(z,lut,cosm
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 554

0554  
0555  
0556  
0557  
0558  
0559  
0560  
0561  
0562  
0563  
0564  
0565  
0566  
0567  
0568  
0569  
0570  
0571  
0572  
0573  
0574  
0575  
0576  
0577  
0578  
0579  
0580  
0581  
0582  
0583

```
! Two- to one-halo transit
REAL :: alpha
REAL, INTENT(IN) :: z
TYPE(HM_tables), INTENT(I
TYPE(HM_cosmology), INTEN

IF(imead==0) THEN
    ! Set to 1 for the sta
    alpha=1.
ELSE IF(imead==1) THEN
    ! This uses the top-ha
    ! Mead et al. (2016; a
    alpha=3.24*1.85**lut%
ELSE IF(imead==2) THEN
    ! Mead et al. (2015) v
    alpha=2.93*1.77**lut%
END IF

! Catches values of alpha
IF(alpha>2.) alpha=2.
IF(alpha<0.5) alpha=0.5

END FUNCTION alpha

FUNCTION r_nl(lut)
! Calculates R_nl, defined
TYPE(HM_tables), INTENT(I
REAL :: r_nl
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 584

```
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613

      IF(lut%nu(1)>1.) THEN
        !This catches some ve
        r_nl=lut%rr(1)
      ELSE
        r_nl=exp(find(log(1.))
      END IF

      END FUNCTION r_nl

      SUBROUTINE halomod(k,z,p1

!Calcuates 1-halo and 2-h
REAL, INTENT(OUT) :: p1h,
REAL, INTENT(IN) :: plin,
REAL :: a
TYPE(HM_cosmology), INTEN
TYPE(HM_tables), INTENT(I

      !Calls expressions for on
!to form the full power s
      IF(k==0.) THEN
        p1h=0.
        p2h=0.
      ELSE
        p1h=p_1h(k,z,lut,cosm
        p2h=p_2h(k,z,plin,lut
      END IF

      a=alpha(z,lut,cosm)
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 614

```
0614 pfull=(p2h**a+p1h**a)**(1
0615
0616      END SUBROUTINE halomod
0617
0618      SUBROUTINE fill_table(min
0619
0620      !Fills array 'arr' in equ
0621      IMPLICIT NONE
0622      INTEGER :: i
0623      REAL, INTENT(IN) :: min,
0624      REAL, ALLOCATABLE :: arr(
0625      INTEGER, INTENT(IN) :: n
0626
0627      !Allocate the array, and
0628      IF(ALLOCATED(arr)) DEALLO
0629      ALLOCATE(arr(n))
0630      arr=0.
0631
0632      IF(n==1) THEN
0633          arr(1)=min
0634      ELSE IF(n>1) THEN
0635          DO i=1,n
0636              arr(i)=min+(max-m
0637          END DO
0638      END IF
0639
0640      END SUBROUTINE fill_table
0641
0642      SUBROUTINE fill_table8(mi
0643
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 644

```
0644 !Fills array 'arr' in equ
0645 IMPLICIT NONE
0646 INTEGER :: i
0647 real(dl), INTENT(IN) :: m
0648 real(dl), ALLOCATABLE :: arr
0649 INTEGER, INTENT(IN) :: n
0650
0651 !Allocate the array, and
0652 IF(ALLOCATED(arr)) DEALLOCATE(arr)
0653 ALLOCATE(arr(n))
0654 arr=0.
0655
0656 IF(n==1) THEN
0657     arr(1)=min
0658 ELSE IF(n>1) THEN
0659     DO i=1,n
0660         arr(i)=min+(max-m)
0661     END DO
0662 END IF
0663
0664 END SUBROUTINE fill_table
0665
0666
0667
0668 !Fills internal HMcode HM
0669 TYPE(MatterPowerData), INTENT(IN) :: z
0670 INTEGER, INTENT(IN) :: iz
0671 TYPE(HM_cosmology) :: cos
0672 INTEGER :: i
0673 REAL :: z, g
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 674

```
0674      INTEGER, PARAMETER :: imel=1
0675      REAL, PARAMETER :: pi=3.141592653589793
0676      REAL, PARAMETER :: kmin=1.0
0677      REAL, PARAMETER :: kmax=1.0
0678      INTEGER, PARAMETER :: nk=1000
0679
0680      IF(HM_verbose) WRITE(*,*) 'Halofit'
0681
0682      !Fill arrays
0683      IF(ALLOCATED(cosm%k_plin))
0684      IF(ALLOCATED(cosm%plin))
0685      IF(ALLOCATED(cosm%plinc))
0686
0687      IF(imeth==1) THEN
0688
0689          !Fill k-table with the user's choice
0690          !If a user has specified a calculation chug
0691          !cosm%nk=CAMB_PK%num_k
0692          ALLOCATE(cosm%k_plin(nk))
0693          DO i=1,cosm%nk
0694              cosm%k_plin(i)=exp((i-1)*lognk)
0695          END DO
0696
0697
0698      ELSE IF(imeth==2) THEN
0699
0700          !Fill a k-table with the minimum
0701          !Note that the minimum value is nk
0702          cosm%nk=nk
0703          CALL fill_table(log(k),nk)
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 704

```
0704      cosm%k_plin=exp(cosm%
0705
0706      END IF
0707
0708      IF(HM_verbose) WRITE(*,*) 
0709      IF(HM_verbose) WRITE(*,*) 
0710      IF(HM_verbose) WRITE(*,*) 
0711
0712      ALLOCATE(cosm%plin(nk),co
0713
0714      !Find the redshift
0715      z=CAMB_Pk%Redshifts(iz)
0716      IF(HM_verbose) WRITE(*,*) 
0717
0718      !Fill power table, both c
0719      DO i=1,nk
0720          !Take the power from
0721          cosm%plin(i)=MatterPo
0722          cosm%plinc(i)=cosm%pl
0723      END DO
0724
0725      !Calculate the growth fac
0726      g=grow(z,cosm)
0727
0728      !Grow the power to z=0
0729      cosm%plin=cosm%plin/(g**2
0730      cosm%plinc=cosm%plinc/(g*
0731
0732      !Check sigma_8 value
0733      IF(HM_verbose) WRITE(*,*)
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 734

```
0734 IF(HM_verbose) WRITE(*,*)  
0735 IF(HM_verbose) WRITE(*,*)  
0736  
0737 END SUBROUTINE fill_plint  
0738  
0739 FUNCTION Tcb_Tcbnu_ratio(  
0740  
0741 !Calculates the ratio of  
0742 !Uses approximations in E  
0743 !Note that this assumes t  
0744 !Nnu<=3 of these being ma  
0745  
0746 REAL :: Tcb_Tcbnu_ratio  
0747 REAL, INTENT(IN) :: k, z  
0748 REAL :: D, Dcb, Dcbnu, pc  
0749 REAL :: BigT  
0750 TYPE(HM_cosmology) :: cosm  
0751  
0752 IF(cosm%f_nu==0.) THEN  
0753  
0754 Tcb_Tcbnu_ratio=1.  
0755  
0756 ELSE  
0757  
0758 !Growth exponent unde  
0759 pcb=(5.-sqrt(1.+24.*(  
0760  
0761 !Theta for temperatur  
0762 BigT=cosm%Tcmb/2.7  
0763
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 764

```
0764 !The matter-radiation
0765 zeq=(2.5e4)*cosm%om_m
0766
0767 !The growth function
0768 !For my purpose (just
0769 !In any case, can't u
0770 D=(1.+zeq)/(1.+z)
0771
0772 !Wave number relative
0773 !Extra factor of h be
0774 q=k*cosm%h*BigT**2./((
0775
0776 !Free streaming scale
0777 !Note that Eisenstein
0778 !with Nnu of these be
0779 yfs=17.2*cosm%f_nu*(1
0780
0781 !These are (almost) t
0782 !Some part is missing
0783 !Equations (12) and (
0784 Dcb=(1.+(D/(1.+yfs)))*
0785 Dcbnu=((1.-cosm%f_nu)
0786
0787 Tcb_Tcbnu_ratio=Dcb/D
0788
0789 END IF
0790
0791 END FUNCTION Tcb_Tcbnu_ra
0792
0793 SUBROUTINE assign_HM_cosm
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 794

0794  
0795  
0796  
0797  
0798  
0799  
0800  
0801  
0802  
0803  
0804  
0805  
0806  
0807  
0808  
0809  
0810  
0811  
0812  
0813  
0814  
0815  
0816  
0817  
0818  
0819  
0820  
0821  
0822  
0823

**!Assigns the internal HMc  
TYPE(HM\_cosmology) :: cos**

**!Converts CAMB parameters  
cosm%om\_m=CP%omegac+CP%om  
cosm%om\_v=CP%omegav  
cosm%w=w\_lam  
cosm%wa=wa\_ppf  
cosm%f\_nu=CP%omegan/cosm%  
cosm%h=CP%H0/100.  
cosm%Tcmb=CP%tcmb  
cosm%Nnu=CP%Num\_Nu\_massiv**

**!n\_s is read in here. The  
!one value in this array,  
cosm%ns=CP%InitPower%an(1**

**!Write out cosmological p  
IF(HM\_verbose) WRITE(\*,\*)  
IF(HM\_verbose) WRITE(\*,\*)**

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 824

```
0824 IF(HM_verbose) WRITE(*,*)  
0825  
0826 END SUBROUTINE assign_HM_  
0827  
0828 SUBROUTINE initialise_HM_  
0829  
0830 !Sets up HM_tables of sig  
0831 TYPE(MatterPowerData), INTENT(IN)  
0832 TYPE(HM_cosmology) :: cosm  
0833 INTEGER, INTENT(IN) :: iz  
0834  
0835 !Fill linear power table  
0836 CALL fill_plintab(iz,cosm)  
0837  
0838 !Fill sigma(r) table  
0839 CALL fill_sigtab(cosm)  
0840  
0841 END SUBROUTINE initialise_  
0842  
0843 SUBROUTINE allocate_LUT(l  
0844  
0845 !Allocates memory for the  
0846 TYPE(HM_tables) :: lut  
0847 INTEGER :: n  
0848  
0849 n=lut%n  
0850 ALLOCATE(lut%zc(n),lut%m(  
0851 ALLOCATE(lut%nu(n),lut%rr  
0852  
0853 lut%zc=0.
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 854

```
0854      lut%m=0.  
0855      lut%c=0.  
0856      lut%rv=0.  
0857      lut%nu=0.  
0858      lut%rr=0.  
0859      lut%sigf=0.  
0860      lut%sig=0.  
0861  
0862      END SUBROUTINE allocate_L  
0863  
0864      SUBROUTINE deallocate_LUT  
0865  
0866      !Deallocates HMcode look-  
0867      TYPE(HM_tables) :: lut  
0868  
0869      DEALLOCATE(lut%zc,lut%m,l  
0870  
0871      END SUBROUTINE deallocate  
0872  
0873      SUBROUTINE halomod_init(z  
0874  
0875      !Halo-model initialisatio  
0876      !Computes look-up HM_tabl  
0877      REAL, INTENT(IN) :: z  
0878      INTEGER :: i  
0879      REAL :: Dv, dc, f, m, nu,  
0880      TYPE(HM_cosmology) :: cos  
0881      TYPE(HM_tables) :: lut  
0882      REAL, PARAMETER :: mmin=1  
0883      REAL, PARAMETER :: mmax=1
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 884

```
0884      INTEGER, PARAMETER :: n=2
0885
0886      IF(HM_verbose) WRITE(*,*) 
0887      IF(HM_verbose) WRITE(*,*) 
0888
0889      !Find value of sigma_v, s
0890
0891      lut%sigv=sqrt(displint(0.,
0892      IF(HM_verbose) WRITE(*,*) 
0893      lut%sigv100=sqrt(displint(
0894      IF(HM_verbose) WRITE(*,*) 
0895      lut%sig8z=sigma(8.,z,0,co
0896      IF(HM_verbose) WRITE(*,*) 
0897
0898      IF(ALLOCATED(lut%rr)) CAL
0899
0900      lut%n=n
0901      CALL allocate_lut(lut)
0902
0903      IF(HM_verbose) WRITE(*,*) 
0904      IF(HM_verbose) WRITE(*,*) 
0905
0906      dc=delta_c(z,lut,cosm)
0907
0908      !$OMP PARALLEL DO default
0909      DO i=1,n
0910
0911      m=exp(log(mmin)+log(m
0912      r=radius_m(m,cosm)
0913      sig=sigmac(r,z,cosm)
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 914

```
0914          nu=dc/sig
0915
0916          lut%m(i)=m
0917          lut%rr(i)=r
0918          lut%sig(i)=sig
0919          lut%nu(i)=nu
0920
0921          END DO
0922          !$OMP END PARALLEL DO
0923
0924          IF(HM_verbose) WRITE(*,*) 
0925
0926          !Fills up a table for sig
0927          !This is the f=0.01 param
0928          f=0.01**(1./3.)
0929          DO i=1,lut%n
0930              lut%sigf(i)=sigmac(lu
0931          END DO
0932          IF(HM_verbose) WRITE(*,*) 
0933
0934          !Fill virial radius table
0935          Dv=Delta_v(z,lut,cosm)
0936          lut%rv=lut%rr/(Dv**(1./3.
0937
0938          IF(HM_verbose) WRITE(*,*) 
0939          IF(HM_verbose) WRITE(*,*) 
0940          IF(HM_verbose) WRITE(*,*) 
0941          IF(HM_verbose) WRITE(*,*) 
0942          IF(HM_verbose) WRITE(*,*) 
0943          IF(HM_verbose) WRITE(*,*) 
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 944



0944	IF(HM_verbose) WRITE(*,*)
0945	
0946	!Find non-linear radius a
0947	lut%rnl=r_nl(lut)
0948	lut%knl=1./lut%rnl
0949	
0950	IF(HM_verbose) WRITE(*,*)
0951	IF(HM_verbose) WRITE(*,*)
0952	
0953	!Calcuate the effective s
0954	lut%neff=neff(lut,cosm)
0955	
0956	IF(HM_verbose) WRITE(*,*)
0957	
0958	!Get the concentration fo
0959	CALL conc_bull(z,lut,cosm)
0960	
0961	IF(HM_verbose) WRITE(*,*)
0962	IF(HM_verbose) WRITE(*,*)
0963	IF(HM_verbose) WRITE(*,*)
0964	IF(HM_verbose) WRITE(*,*)
0965	IF(HM_verbose) WRITE(*,*)
0966	IF(HM_verbose) WRITE(*,*)
0967	
0968	!Switch off verbose mode
0969	HM_verbose=.false.
0970	
0971	END SUBROUTINE halomod_in
0972	
0973	SUBROUTINE write_paramete



/Users/lp1lopa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1lopa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1004

```
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033
```

```
END FUNCTION radius_m
```

```
FUNCTION neff(lut,cosm)
```

```
!Finds the effective spec  
!Where nu(r_nl)=1.  
REAL :: neff  
REAL :: ns  
TYPE(HM_cosmology), INTEN  
TYPE(HM_tables), INTENT(I
```

```
!Numerical differentiation  
neff=-3.-derivative_table
```

```
!For some bizarre cosmolo  
!In this case the n_eff c  
ns=cosm%ns  
IF(neff<ns-4.) neff=ns-4.  
IF(neff>ns) neff=ns
```

```
END FUNCTION neff
```

```
SUBROUTINE conc_bull(z,lut)
```

```
!Calculates the Bullock e  
REAL, INTENT(IN) :: z  
TYPE(HM_cosmology) :: cosm  
TYPE(HM_tables) :: lut  
REAL :: A, zf, ainf, zinf
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1034

```
1034      INTEGER :: i
1035
1036      !Amplitude of relation (4
1037      A=As(z,lut,cosm)
1038
1039      !Fill the collapse time 1
1040      CALL zcoll_bull(z,cosm,lu
1041
1042      !Fill the concentration 1
1043      DO i=1,lut%n
1044          zf=lut%zc(i)
1045          lut%c(i)=A*(1.+zf)/(1.
1046      END DO
1047
1048      !Dolag (2004) prescriptio
1049
1050      !This is approximately z=
1051      zinf=10.
1052      g_wcdm=grow(zinf,cosm)
1053
1054      !Make a LCDM HM_cosmology
1055      !Only need to make sure m
1056      !This is *only* used for
1057      cos_lcdm=cosm
1058      DEALLOCATE(cos_lcdm%growt
1059      DEALLOCATE(cos_lcdm%a_gro
1060      cos_lcdm%w=-1.
1061      cos_lcdm%wa=0.
1062      cos_lcdm%om_v=1.-cosm%om_
1063      
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1064

```
1064 ainf=1./(1.+zinf)
1065
1066 !Needs to use grow_int ex
1067 g_lcdm=growint(ainf,cos_l
1068
1069 !This is the Dolag et al.
1070 IF(imead==0 .OR. imead==1
1071     pow=1.
1072 ELSE IF(imead==2) THEN
1073     pow=1.5
1074 END IF
1075 lut%c=lut%c*((g_wcdm/g_lc
1076
1077 END SUBROUTINE conc_bull
1078
1079 FUNCTION growint(a,cosm)
1080
1081 !Integrates between a and
1082 !Stores information to re
1083 IMPLICIT NONE
1084 REAL :: growint
1085 REAL, INTENT(IN) :: a
1086 TYPE(HM_cosmology), INTEN
1087 REAL :: b
1088 INTEGER :: i, j
1089 INTEGER :: n
1090 REAL :: x, dx
1091 REAL :: f1, f2, fx
1092 real(dl) :: sum_n, sum_2n
1093 INTEGER, PARAMETER :: jmi
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1094

```
1094      INTEGER, PARAMETER :: jma
1095      real(dl), PARAMETER :: ac
1096      INTEGER, PARAMETER :: ior
1097
1098      !Integration range for in
1099      !Note a -> 1
1100      b=1.d0
1101
1102      IF(a==b) THEN
1103
1104          !Fix the answer to ze
1105          growint=exp(0.d0)
1106
1107      ELSE
1108
1109          !Reset the sum variab
1110          sum_2n=0.d0
1111          sum_n=0.d0
1112          sum_old=0.d0
1113          sum_new=0.d0
1114
1115      DO j=1,jmax
1116
1117          !Note, you need t
1118          !j=1 n=2; j=2 n=3
1119          n=1+2** (j-1)
1120
1121          !Calculate the dx
1122          dx=(b-a)/REAL(n-1
1123
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1124



```
1124 IF(j==1) THEN
1125 !The first go
1126 f1=growint_in
1127 f2=growint_in
1128 sum_2n=0.5d0*
1129 sum_new=sum_2
1130
1131
1132 ELSE
1133
1134 !Loop over on
1135 DO i=2,n,2
1136 x=a+(b-a)
1137 fx=growin
1138 sum_2n=sum_
1139 END DO
1140
1141 !Now create t
1142 sum_2n=sum_n/
1143
1144 !Now calculate
1145 IF(iorder==1)
1146 sum_new=s
1147 ELSE IF(iorde
1148 sum_new=(
1149 ELSE
1150 STOP 'GRO
1151 END IF
1152
1153 END IF
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1154

```
1154 IF((j>=jmin) .AND.  
1155 !jmin avoids  
1156 growint=exp(s  
1157 EXIT  
1158 ELSE IF(j==jmax)  
1159 STOP 'GROWINT'  
1160 ELSE  
1161 !Integral has  
1162 sum_old=sum_n  
1163 sum_n=sum_2n  
1164 sum_2n=0.d0  
1165 END IF  
1166  
1167 END DO  
1168  
1169 END IF  
1170  
1171  
1172 END FUNCTION growint  
1173  
1174 FUNCTION growint_integrand  
1175 !Integrand for the approx  
1176 IMPLICIT NONE  
1177 REAL :: growint_integrand  
1178 REAL, INTENT(IN) :: a  
1179 TYPE(HM_cosmology), INTEN  
1180 REAL :: gam  
1181  
1182 IF(cosm%w<-1.) THEN  
1183
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1184

```
1184      gam=0.55+0.02*(1.+cos  
1185      ELSE IF(cosm%w>-1) THEN  
1186          gam=0.55+0.05*(1.+cos  
1187      ELSE  
1188          gam=0.55  
1189      END IF  
1190  
1191      !Note the minus sign here  
1192      growint_integrand=-(Omega  
1193  
1194      END FUNCTION growint_inte  
1195  
1196      SUBROUTINE zcoll_bull(z,c  
1197          !Calculates the halo colla  
1198          REAL, INTENT(IN) :: z  
1199          TYPE(HM_cosmology) :: cosm  
1200          TYPE(HM_tables) :: lut  
1201          REAL :: dc  
1202          REAL :: af, zf, RHS, a, g  
1203          INTEGER :: i  
1204  
1205          !This fills up the halo f  
1206          !Needs to interpolate g(z  
1207          !in 'g(a)' vs 'a' space for  
1208          !dc=delta_c(z,lut,cosm)  
1209  
1210          !Find the growth function
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1214

```
1214      a=1./(1.+z)
1215      growz=find(a,cosm%a_growt
1216
1217      !Do numerical inversion
1218      DO i=1,lut%n
1219
1220      RHS=dc*grow(z,cosm)/1
1221
1222      IF(RHS>growz) THEN
1223          !This is the case
1224          !in this case set
1225          zf=z
1226      ELSE
1227          af=find(RHS,cosm%
1228          zf=-1.+1./af
1229      END IF
1230
1231      lut%zc(i)=zf
1232
1233      END DO
1234
1235      END SUBROUTINE zcoll_bull
1236
1237      FUNCTION mass_r(r,cosm)
1238
1239      !Calcuates the average ma
1240      REAL :: mass_r, r
1241      TYPE(HM_cosmology) :: cos
1242      REAL, PARAMETER :: pi=3.1
1243
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1244

```
1244 !Relation between mean co
1245 mass_r=(4.*pi/3.)*cosmic_
1246
1247 END FUNCTION mass_r
1248
1249 PURE FUNCTION cosmic_dens
1250
1251 !The z=0 cosmological mat
1252 REAL :: cosmic_density
1253 TYPE(HM_cosmology), INTEN
1254
1255 !In M_sun per Mpc^3 with
1256 cosmic_density=(2.775e11)
1257
1258 END FUNCTION cosmic_densi
1259
1260 FUNCTION find_pk(k,itype,
1261
1262 !Look-up and interpolatio
1263 REAL :: find_pk
1264 REAL :: kmax, ns
1265 REAL, INTENT(IN) :: k
1266 INTEGER, INTENT(IN) :: it
1267 INTEGER :: n
1268 TYPE(HM_cosmology), INTEN
1269
1270 !Set number of k points a
1271 !Note that the min k valu
1272 n=SIZE(cosm%k_plin)
1273 kmax=cosm%k_plin(n)
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1274

```
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
```

! Spectral index used in t  
ns=cosm%ns

IF(k>kmax) THEN  
 !Do some interpolation  
 IF(itype==0) THEN  
 find\_pk=cosm%plin  
 ELSE IF(itype==1) THE  
 find\_pk=cosm%plin  
 END IF

ELSE  
 !Otherwise use the st  
 IF(itype==0) THEN  
 find\_pk=exp(find(  
 ELSE IF(itype==1) THE  
 find\_pk=exp(find(  
 END IF

END IF

!Old method, works fine f  
!IF(itype==1) find\_pk=fin

END FUNCTION find\_pk

FUNCTION p\_lin(k,z,itype,  
REAL :: p\_lin  
REAL, INTENT (IN) :: k, z

!Looks up the value for t

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1304

```
1304      INTEGER, INTENT(IN) :: it
1305      TYPE(HM_cosmology), INTEN
1306
1307      !This gives the linear po
1308      !P(k) should have been pr
1309
1310      p_lin=(grow(z,cosm)**2.)*
1311
1312      END FUNCTION p_lin
1313
1314      FUNCTION p_2h(k,z,plin,lu
1315
1316      !Calculates the 2-halo te
1317      REAL :: p_2h
1318      REAL, INTENT(IN) :: k, pl
1319      REAL :: sigv, frac
1320      REAL, INTENT(IN) :: z
1321      TYPE(HM_tables), INTENT(I
1322      TYPE(HM_cosmology), INTEN
1323
1324      !Damping function
1325      frac=fdamp(z,lut,cosm)
1326
1327      IF(imead==0 .OR. frac<1.e
1328          p_2h=plin
1329      ELSE
1330          sigv=lut%sigv
1331          p_2h=plin*(1.-frac*(t
1332      END IF
1333
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1334

```
1334 !For some strange cosmolo
1335 IF(p_2h<0.) p_2h=0.
1336
1337 END FUNCTION p_2h
1338
1339 FUNCTION p_1h(k,z,lut,cos
1340
1341 !Calculates the 1-halo te
1342 REAL :: p_1h
1343 REAL, INTENT(IN) :: k, z
1344 TYPE(HM_tables), INTENT(I
1345 TYPE(HM_cosmology), INTEN
1346 REAL :: Dv, g, fac, et, k
1347 REAL :: integrand(lut%n)
1348 REAL :: sum
1349 INTEGER :: i
1350 REAL, PARAMETER :: pi=3.1
1351
1352 !Does the one-halo power
1353
1354 !Only call eta once
1355 et=eta(z,lut,cosm)
1356
1357 !Calculates the value of
1358 DO i=1,lut%n
1359   g=gnu(lut%nu(i))
1360   wk=win(k*(lut%nu(i)**2)
1361   integrand(i)=(lut%rv(
1362   END DO
1363 
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1364

```
1364 !Carries out the integrat
1365 sum=REAL(inttab(lut%nu,RE
1366 
1367 !Virial density
1368 Dv=Delta_v(z,lut,cosm)
1369 
1370 !These are just numerical
1371 p_1h=sum*2.*Dv*(k**3.)/(3
1372 
1373 !Damping of the 1-halo te
1374 ks=kstar(z,lut,cosm)
1375 
1376 !Prevents problems if k/k
1377 IF(ks==0.) THEN
1378   fac=0.
1379 ELSE IF((k/ks)**2.>7.) TH
1380   fac=0.
1381 ELSE
1382   fac=exp(-((k/ks)**2.))
1383 END IF
1384 
1385 !Damping of the one-halo
1386 p_1h=p_1h*(1.-fac)
1387 
1388 END FUNCTION p_1h
1389 
1390 SUBROUTINE fill_sigtab(co
1391 
1392 !Fills look-up HM_tables
1393 REAL :: r, sig
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1394

```
1394 INTEGER :: i
1395 TYPE(HM_cosmology) :: cos
1396 REAL, PARAMETER :: rmin=1
1397 REAL, PARAMETER :: rmax=1
1398 INTEGER, PARAMETER :: nsig=1000
1399
1400 !This fills up HM tables
1401 !It is used only in look-
1402 !and prevents a large number of points
1403 !rmin and rmax need to be set
1404 !R vs. sigma(R) is approx
1405 !This wouldn't be appropriate
1406
1407 !Allocate arrays
1408 IF(ALLOCATED(cosm%r_sigma)) DEALLOCATE(cosm%r_sigma)
1409 IF(ALLOCATED(cosm%sigma)) DEALLOCATE(cosm%sigma)
1410
1411 !These values of 'r' work
1412 !Having nsig as a 2** num
1413 cosm%nsig=nsig
1414 ALLOCATE(cosm%r_sigma(nsig))
1415
1416 IF(HM_verbose) WRITE(*,*)
1417 IF(HM_verbose) WRITE(*,*)
1418 IF(HM_verbose) WRITE(*,*)
1419 IF(HM_verbose) WRITE(*,*)
1420
1421 !$OMP PARALLEL DO default(none)
1422 DO i=1,nsig
1423
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1424

```
1424 !Equally spaced r in
1425 r=exp(log(rmin)+log(r
1426
1427 sig=sigma(r,0.,1,cosm
1428
1429 cosm%r_sigma(i)=r
1430 cosm%sigma(i)=sig
1431
1432 END DO
1433 !$OMP END PARALLEL DO
1434
1435 IF(HM_verbose) WRITE(*,*) 
1436 IF(HM_verbose) WRITE(*,*) 
1437 IF(HM_verbose) WRITE(*,*) 
1438 IF(HM_verbose) WRITE(*,*) 
1439
1440 END SUBROUTINE fill_sigta
1441
1442 FUNCTION sigmac(r,z,cosm)
1443
1444 !Finds sigma_cold(R) from
1445 REAL :: sigmac
1446 REAL, INTENT(IN) :: r, z
1447 TYPE(HM_cosmology), INTEN
1448
1449 !Assumes scale-independet
1450 !Uses the approximation s
1451
1452 sigmac=grow(z,cosm)*exp(f
1453
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1454

```
1454      END FUNCTION sigmac
1455
1456      FUNCTION wk_tophat(x)
1457
1458      !The normlaised Fourier T
1459      REAL :: wk_tophat, x
1460
1461      !Taylor expansion used fo
1462      IF(x<0.01) THEN
1463          wk_tophat=1.-(x**2.)/
1464      ELSE
1465          wk_tophat=3.* (sin(x)-
1466      END IF
1467
1468      END FUNCTION wk_tophat
1469
1470      FUNCTION inttab(x,y,n,ior
1471
1472      !Integrates tables y(x)dx
1473      REAL :: inttab
1474      INTEGER, INTENT(IN) :: n
1475      REAL, INTENT(IN) :: x(n),
1476      REAL :: a, b, c, d, h
1477      REAL :: q1, q2, q3, qi, q
1478      REAL :: x1, x2, x3, x4, y
1479      real(dl) :: sum
1480      INTEGER :: i, i1, i2, i3,
1481      INTEGER, INTENT(IN) :: io
1482
1483      sum=0.d0
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1484



```
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
```

IF (iorder==1) THEN

! Sums over all Trapez

DO i=1,n-1

    a=y(i+1)

    b=y(i)

    h=x(i+1)-x(i)

    sum=sum+(a+b)\*h/2

END DO

ELSE IF (iorder==2) THEN

DO i=1,n-2

    x1=x(i)

    x2=x(i+1)

    x3=x(i+2)

    y1=y(i)

    y2=y(i+1)

    y3=y(i+2)

CALL fit\_quadrati

    q1=a\*(x1\*\*3.)/3.+

    q2=a\*(x2\*\*3.)/3.+

    q3=a\*(x3\*\*3.)/3.+

! Takes value for

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1514

```
1514      ! have two indepen
1515      IF(n==3) THEN
1516          sum=sum+q3-q1
1517      ELSE IF(i==1) THE
1518          sum=sum+(q2-q
1519      ELSE IF(i==n-2) T
1520          sum=sum+(q2-q
1521      ELSE
1522          sum=sum+(q3-q
1523      END IF
1524
1525      END DO
1526
1527      ELSE IF(iorder==3) THEN
1528          DO i=1,n-1
1529
1530              !First choose the
1531              !First and last a
1532
1533
1534          IF(i==1) THEN
1535
1536              i1=1
1537              i2=2
1538              i3=3
1539              i4=4
1540
1541          ELSE IF(i==n-1) T
1542
1543              i1=n-3
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1544



```
1544          i2=n-2
1545          i3=n-1
1546          i4=n
1547
1548          ELSE
1549
1550          i1=i-1
1551          i2=i
1552          i3=i+1
1553          i4=i+2
1554
1555          END IF
1556
1557          x1=x(i1)
1558          x2=x(i2)
1559          x3=x(i3)
1560          x4=x(i4)
1561
1562          y1=y(i1)
1563          y2=y(i2)
1564          y3=y(i3)
1565          y4=y(i4)
1566
1567          CALL fit_cubic(a,
1568
1569          !These are the li
1570          xi=x(i)
1571          xf=x(i+1)
1572
1573          qi=a*(xi**4.)/4.+
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1574

```
1574      qf=a*(xf**4.)/4.+
1575      sum=sum+qf-qi
1576      END DO
1577      ELSE
1578      ERROR STOP 'INTTAB: E'
1579      END IF
1580      inttab=REAL(sum)
1581      END FUNCTION inttab
1582
1583      FUNCTION sigma(r,z,ittype,
1584      !Gets sigma(R)
1585      IMPLICIT NONE
1586      REAL :: sigma
1587      REAL, INTENT(IN) :: r, z
1588      INTEGER, INTENT(IN) :: it
1589      TYPE(HM_cosmology), INTEN
1590      REAL, PARAMETER :: acc=1d
1591      INTEGER, PARAMETER :: ior
1592      REAL, PARAMETER :: rsplit
1593
1594      IF(r>=rsplit) THEN
1595      sigma=sqrt(sigint0(r,
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1604

```
1604      ELSE IF(r<rsplit) THEN
1605          sigma=sqrt(sigint1(r,
1606      ELSE
1607          STOP 'SIGMA: Error, s
1608      END IF
1609
1610      END FUNCTION sigma
1611
1612      FUNCTION sigma_integrand(
1613
1614!The integrand for the si
1615IMPLICIT NONE
1616REAL :: sigma_integrand
1617REAL, INTENT(IN) :: k, R,
1618INTEGER, INTENT(IN) :: it
1619TYPE(HM_cosmology), INTEN
1620REAL :: y, w_hat
1621
1622IF(k==0.d0) THEN
1623    sigma_integrand=0.d0
1624ELSE
1625    y=k*R
1626    w_hat=wk_tophat(y)
1627    sigma_integrand=p_lin
1628END IF
1629
1630END FUNCTION sigma_integ
1631
1632      FUNCTION sigma_integrand_
1633
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1634

```
1634 !The integrand for the si
1635 IMPLICIT NONE
1636 REAL :: sigma_integrand_t
1637 REAL, INTENT(IN) :: t, R,
1638 INTEGER, INTENT(IN) :: it
1639 TYPE(HM_cosmology), INTEN
1640 REAL :: k, y, w_hat
1641
1642 INTERFACE
1643 FUNCTION f(x)
1644 REAL :: f
1645 REAL, INTENT(IN) :: x
1646 END FUNCTION f
1647 END INTERFACE
1648
1649 !Integrand to the sigma i
1650
1651 IF(t==0.d0) THEN
1652   !t=0 corresponds to k
1653   sigma_integrand_trans
1654 ELSE IF(t==1.d0) THEN
1655   !t=1 corresponds to k
1656   sigma_integrand_trans
1657 ELSE
1658   !f(R) can be *any* fu
1659   k=(-1.d0+1.d0/t)/f(R)
1660   y=k*R
1661   w_hat=wk_tophat(y)
1662   sigma_integrand_trans
1663 END IF
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1664

1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693

```
END FUNCTION sigma_integr

FUNCTION sigint0(r,z,ityp)

!Integrates between a and
!Stores information to re
IMPLICIT NONE
REAL :: sigint0
REAL, INTENT(IN) :: r, z
INTEGER, INTENT(IN) :: it
TYPE(HM_cosmology), INTEN
REAL, INTENT(IN) :: acc
INTEGER, INTENT(IN) :: io
INTEGER :: i, j
INTEGER :: n
REAL :: x, dx
REAL :: f1, f2, fx
real(dl) :: sum_n, sum_2n
INTEGER, PARAMETER :: jmi
INTEGER, PARAMETER :: jma
REAL, PARAMETER :: a=0.d0
REAL, PARAMETER :: b=1.d0

IF(a==b) THEN
    !Fix the answer to ze
    sigint0=0.d0
ELSE
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1694

```
1694 !Reset the sum variab
1695 sum_2n=0.d0
1696 sum_n=0.d0
1697 sum_old=0.d0
1698 sum_new=0.d0
1699
1700
1701 DO j=1,jmax
1702
1703 !Note, you need t
1704 ! j=1 n=2; j=2 n=3
1705 n=1+2** (j-1)
1706
1707 !Calculate the dx
1708 dx=(b-a)/REAL(n-1)
1709
1710 IF(j==1) THEN
1711
1712 !The first go
1713 f1=sigma_inte
1714 f2=sigma_inte
1715 sum_2n=0.5d0*
1716 sum_new=sum_2
1717
1718 ELSE
1719
1720 !Loop over on
1721 DO i=2,n,2
1722 x=a+(b-a)
1723 fx=sigma_
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1724



```
1724      sum_2n=su
1725      END DO
1726
1727      !Now create t
1728      sum_2n=sum_n/
1729
1730      !Now calculat
1731      IF(iorder==1)
1732          sum_new=s
1733      ELSE IF(iorde
1734          sum_new=(
1735      ELSE
1736          STOP 'SIG
1737      END IF
1738
1739      END IF
1740
1741      IF((j>=jmin) .AND.
1742          !jmin avoids
1743          sigint0=REAL(
1744          EXIT
1745      ELSE IF(j==jmax)
1746          STOP 'SIGINTO
1747      ELSE
1748          !Integral has
1749          sum_old=sum_n
1750          sum_n=sum_2n
1751          sum_2n=0.d0
1752      END IF
1753
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1754

```
1754      END DO
1755
1756      END IF
1757
1758      END FUNCTION sigint0
1759
1760      FUNCTION f0_rapid(r)
1761
1762      !This is the 'rapidising'
1763      !for sigma(R). Found by t
1764      IMPLICIT NONE
1765      REAL :: f0_rapid
1766      REAL, INTENT(IN) :: r
1767      REAL :: alpha
1768      REAL, PARAMETER :: rsplit
1769
1770      IF(r>rsplit) THEN
1771          !alpha 0.3-0.5 works
1772          alpha=0.5d0
1773      ELSE
1774          !If alpha=1 this goes
1775          !alpha 0.7-0.9 works
1776          alpha=0.8d0
1777      END IF
1778
1779      f0_rapid=r**alpha
1780
1781      END FUNCTION f0_rapid
1782
1783      FUNCTION sigint1(r,z,ityp)
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1784

```
1784
1785      ! Integrates between a and
1786      ! Stores information to re
1787      IMPLICIT NONE
1788      REAL :: sigint1
1789      REAL, INTENT(IN) :: r, z
1790      INTEGER, INTENT(IN) :: it
1791      TYPE(HM_cosmology), INTEN
1792      REAL, INTENT(IN) :: acc
1793      INTEGER, INTENT(IN) :: io
1794      REAL :: a, b
1795      INTEGER :: i, j
1796      INTEGER :: n
1797      REAL :: x, dx
1798      REAL :: f1, f2, fx
1799      real(dl) :: sum_n, sum_2n
1800      INTEGER, PARAMETER :: jmi
1801      INTEGER, PARAMETER :: jma
1802
1803      a=r/(r+r**.5d0)
1804      b=1.d0
1805
1806      IF(a==b) THEN
1807
1808          ! Fix the answer to zero
1809          sigint1=0.d0
1810
1811      ELSE
1812
1813          ! Reset the sum variable
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1814

```
1814      sum_2n=0.d0
1815      sum_n=0.d0
1816      sum_old=0.d0
1817      sum_new=0.d0
1818
1819      DO j=1,jmax
1820
1821      !Note, you need t
1822      !j=1 n=2; j=2 n=3
1823      n=1+2** (j-1)
1824
1825      !Calculate the dx
1826      dx=(b-a)/REAL(n-1)
1827
1828      IF(j==1) THEN
1829
1830      !The first go
1831      f1=sigma_inte
1832      f2=sigma_inte
1833      sum_2n=0.5d0*
1834      sum_new=sum_2
1835
1836      ELSE
1837
1838      !Loop over on
1839      DO i=2,n,2
1840          x=a+(b-a)
1841          fx=sigma_
1842          sum_2n=su
1843      END DO
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1844



```
1844 !Now create t
1845 sum_2n=sum_n/
1846
1847
1848 !Now calculate
1849 IF(iorder==1)
1850     sum_new=s
1851 ELSE IF(iorder>1)
1852     sum_new=(sum_
1853
1854 ELSE
1855     STOP 'SIG
1856 END IF
1857
1858
1859 IF((j>=jmin) .AND.
1860 ! jmin avoids
1861 sigint1=REAL(
1862 EXIT
1863 ELSE IF(j==jmax)
1864     STOP 'SIGINT1
1865 ELSE
1866     ! Integral has
1867     sum_old=sum_n
1868     sum_n=sum_2n
1869     sum_2n=0.d0
1870 END IF
1871
1872 END DO
1873
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1874

```
1874      END IF
1875
1876      END FUNCTION sigint1
1877
1878      FUNCTION f1_rapid(r)
1879
1880      !This is the 'rapidising'
1881      !for sigma(R). Found by t
1882      IMPLICIT NONE
1883      REAL :: f1_rapid
1884      REAL, INTENT(IN) :: r
1885      REAL, PARAMETER :: alpha=
1886
1887      f1_rapid=r**alpha
1888
1889      END FUNCTION f1_rapid
1890
1891      FUNCTION sigint2(r,z,ityp)
1892
1893      !Integrates between a and
1894      !Stores information to re
1895      IMPLICIT NONE
1896      REAL :: sigint2
1897      REAL, INTENT(IN) :: r, z
1898      INTEGER, INTENT(IN) :: it
1899      TYPE(HM_cosmology), INTEN
1900      REAL, INTENT(IN) :: acc
1901      INTEGER, INTENT(IN) :: io
1902      REAL :: a, b
1903      INTEGER :: i, j
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1904

```
1904      INTEGER :: n
1905      REAL :: x, dx
1906      REAL :: f1, f2, fx
1907      real(dl) :: sum_n, sum_2n
1908      INTEGER, PARAMETER :: jmi
1909      INTEGER, PARAMETER :: jma
1910      REAL, PARAMETER :: C=10.d
1911
1912      a=1.d0/r
1913      b=C/r
1914
1915      IF(a==b) THEN
1916
1917          !Fix the answer to zero
1918          sigint2=0.d0
1919
1920      ELSE
1921
1922          !Reset the sum variables
1923          sum_2n=0.d0
1924          sum_n=0.d0
1925          sum_old=0.d0
1926          sum_new=0.d0
1927
1928      DO j=1,jmax
1929
1930          !Note, you need to
1931          !j=1 n=2; j=2 n=3
1932          n=1+2** (j-1)
1933
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1934

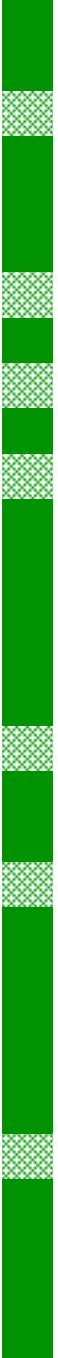
```
1934 !Calculate the dx
      dx=(b-a)/REAL(n-1)
1935
1936
1937 IF(j==1) THEN
1938
1939 !The first go
1940 f1=sigma_inte
1941 f2=sigma_inte
1942 sum_2n=0.5d0*
1943 sum_new=sum_2
1944
1945 ELSE
1946
1947 !Loop over on
1948 DO i=2,n,2
1949   x=a+(b-a)
1950   fx=sigma_
1951   sum_2n=sum_
1952 END DO
1953
1954 !Now create t
1955 sum_2n=sum_n/
1956
1957 !Now calculat
1958 IF(iorder==1)
1959   sum_new=s
1960 ELSE IF(iorde
1961   sum_new=(
1962 ELSE
1963 STOP 'SIG
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1964

```
1964      END IF
1965
1966      END IF
1967
1968      IF((j>=jmin) .AND.
1969          !jmin avoids
1970          sigint2=REAL(
1971              !WRITE(*,*) '
1972              EXIT
1973      ELSE IF(j==jmax)
1974          STOP 'SIGINT2
1975      ELSE
1976          !Integral has
1977          sum_old=sum_n
1978          sum_n=sum_2n
1979          sum_2n=0.d0
1980      END IF
1981
1982      END DO
1983
1984      END IF
1985
1986      END FUNCTION sigint2
1987
1988      FUNCTION win(k,rv,c)
1989
1990          !Selects the halo window
1991          REAL :: win
1992          REAL, INTENT(IN) :: k, rv
```

/Users/lp1lopa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343



/Users/lp1lopa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 1994

```
1994 !Choose the NFW analytic
1995 win=winnfw(k,rv,c)
1996
1997 !Correct for the case of
1998 IF(win>1.) win=1.
1999 IF(win<0.) win=0.
2000
2001 END FUNCTION win
2002
2003
2004
2005 !The analytic Fourier Tra
2006 REAL :: winnfw
2007 REAL, INTENT(IN) :: k, rv
2008 REAL :: sil, si2, cil, ci
2009 REAL :: p1, p2, p3
2010
2011 !Define the scale wavenum
2012 ks=k*rv/c
2013
2014 !Sine and cosine integral
2015 sil=Si(ks)
2016 si2=Si((1.+c)*ks)
2017 cil=Ci(ks)
2018 ci2=Ci((1.+c)*ks)
2019
2020 !These three parts sum to
2021 p1=cos(ks)*(ci2-ci1)
2022 p2=sin(ks)*(si2-sil)
2023 p3=sin(ks*c)/(ks*(1.+c))
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2024

2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053

```
!Create full w(k) and div  
winnfw=p1+p2-p3  
winnfw=winnfw/mass(c)
```

```
END FUNCTION winnfw
```

```
FUNCTION mass(c)
```

```
!This calculates the (nor  
!The 'normalised' mass is  
!where rho_n is the profi  
REAL :: mass  
REAL, INTENT(IN) :: c
```

```
mass=log(1.+c)-c/(1.+c)
```

```
END FUNCTION mass
```

```
FUNCTION gnu(nu)
```

```
!Select the mass function  
REAL :: gnu  
REAL, INTENT(IN) :: nu
```

```
!Sheth & Torman (1999)  
gnu=gst(nu)
```

```
END FUNCTION gnu
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2054

```
2054 FUNCTION gst(nu)
2055 !Sheth & Tormen (1999) ma
2056 REAL :: gst
2057 REAL, INTENT(IN) :: nu
2058 REAL :: p, a, bigA
2059
2060 !Note I use nu=dc/sigma w
2061 !This accounts for the di
2062 ! $f(\nu^2)d(\nu^2)=2*\nu*f(\nu$ 
2063
2064
2065 !Sheth & Tormen fitting n
2066 p=0.3
2067 a=0.707
2068 bigA=0.21616
2069
2070 !Full mass function. Note
2071 gst=bigA*(1.+((a*nu*nu)**2))
2072
2073 END FUNCTION gst
2074
2075 FUNCTION Hubble2(z,cosm)
2076 !This calculates the dime
2077 !Ignores contributions fr
2078 REAL :: Hubble2
2079 REAL, INTENT(IN) :: z
2080 TYPE(HM_cosmology), INTEN
2081 REAL :: a
2082
2083
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343



/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2084

```
2084      a=1./(1.+z)
2085
2086      Hubble2=cosm%om_m*(1.+z)*
2087
2088      END FUNCTION Hubble2
2089
2090      FUNCTION X_de(a,cosm)
2091
2092      !The time evolution for d
2093      !X(a)=1 for LCDM but chan
2094      REAL :: X_de
2095      REAL, INTENT(IN) :: a
2096      TYPE(HM_cosmology), INTEN
2097
2098      X_de=(a**(-3.*(1.+cosm%w+
2099
2100      END FUNCTION X_de
2101
2102      FUNCTION w_de_hm(a,cosm)
2103
2104      !The dark energy w(a) fun
2105      REAL :: w_de_hm
2106      REAL, INTENT(IN) :: a
2107      TYPE(HM_cosmology), INTEN
2108
2109      w_de_hm=cosm%w+(1.-a)*cos
2110
2111      END FUNCTION w_de_hm
2112
2113      FUNCTION Omega_m_hm(z,cos
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2114

```
2114 !This calculates omega_m
2115 REAL :: Omega_m_hm
2116 REAL, INTENT(IN) :: z
2117 REAL :: om_m, a
2118 TYPE(HM_cosmology), INTEN
2119
2120
2121 om_m=cosm%om_m
2122 Omega_m_hm=(om_m*(1.+z)**2)
2123
2124 END FUNCTION Omega_m_hm
2125
2126 FUNCTION grow(z,cosm)
2127 !Finds the scale-independent growth factor
2128 REAL :: grow
2129 REAL, INTENT(IN) :: z
2130 REAL :: a
2131 TYPE(HM_cosmology), INTEN
2132
2133 IF(z==0.) THEN
2134   grow=1.
2135 ELSE
2136   a=1./(1.+z)
2137   grow=find(a,cosm%a_gr)
2138 END IF
2139
2140
2141 END FUNCTION grow
2142
2143 FUNCTION dispint(R,z,cosm)
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2144

```
2144
2145      ! Integrates between a and
2146      ! Stores information to re
2147      IMPLICIT NONE
2148      REAL :: dispint
2149      REAL, INTENT(IN) :: z, R
2150      TYPE(HM_cosmology), INTEN
2151      REAL :: a, b
2152      INTEGER :: i, j
2153      INTEGER :: n
2154      REAL :: x, dx
2155      REAL :: f1, f2, fx
2156      real(dl) :: sum_n, sum_2n
2157      INTEGER, PARAMETER :: jmi
2158      INTEGER, PARAMETER :: jma
2159      REAL, PARAMETER :: acc=1d
2160      INTEGER, PARAMETER :: ior
2161
2162      ! Integration range for in
2163      ! Note 0 -> infinity in k
2164      a=0.d0
2165      b=1.d0
2166
2167      IF(a==b) THEN
2168
2169          ! Fix the answer to ze
2170          dispint=0.
2171
2172      ELSE
2173
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2174

```
2174      !Reset the sum variab
2175      sum_2n=0.d0
2176      sum_n=0.d0
2177      sum_old=0.d0
2178      sum_new=0.d0
2179
2180      DO j=1,jmax
2181
2182      !Note, you need t
2183      !j=1 n=2; j=2 n=3
2184      n=1+2** (j-1)
2185
2186      !Calculate the dx
2187      dx=(b-a)/REAL(n-1)
2188
2189      IF(j==1) THEN
2190
2191      !The first go
2192      f1=dispint_in
2193      f2=dispint_in
2194      sum_2n=0.5d0*
2195      sum_new=sum_2
2196
2197      ELSE
2198
2199      !Loop over on
2200      DO i=2,n,2
2201          x=a+(b-a)
2202          fx=dispin
2203          sum_2n=su
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2204

```
2204      END DO
2205      !Now create t
2206      sum_2n=sum_n/
2207
2208      !Now calculat
2209      IF(iorder==1)
2210          sum_new=s
2211      ELSE IF(iorde
2212          sum_new=(
2213      ELSE
2214          STOP 'DIS
2215      END IF
2216
2217      END IF
2218
2219      IF((j>=jmin) .AND.
2220          !jmin avoids
2221          dispint=REAL(
2222          EXIT
2223
2224      ELSE IF(j==jmax)
2225          STOP 'DISPINT
2226      ELSE
2227          !Integral has
2228          sum_old=sum_n
2229          sum_n=sum_2n
2230          sum_2n=0.d0
2231      END IF
2232
2233      END DO
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2234

```
2234
2235      END IF
2236
2237      END FUNCTION dispint
2238
2239      FUNCTION dispint_integrand
2240
2241 !This is the integrand fo
2242 IMPLICIT NONE
2243 REAL :: dispint_integrand
2244 REAL, INTENT(IN) :: theta
2245 TYPE(HM_cosmology), INTEN
2246 REAL :: k
2247 REAL, PARAMETER :: alpha=
2248 REAL, PARAMETER :: Rsplit
2249
2250 !Note that I have not inc
2251 !The choice of alpha=1.65
2252 !Rsplit=10 is thoughlessl
2253 !Including this seems to
2254
2255 IF(theta==0. .OR. theta==
2256     dispint_integrand=0.
2257 ELSE
2258     !IF(r>Rsplit) THEN
2259     !    k=(-1.d0+1.d0/the
2260     !ELSE
2261     k=(-1.+1./theta)
2262     !END IF
2263     dispint_integrand=(p_
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2264

```
2264      END IF
2265
2266      END FUNCTION dispint_inte
2267
2268      FUNCTION Si(x)
2269
2270      !Calculates the 'sine int
2271      REAL :: Si
2272      REAL, INTENT(IN) :: x
2273      REAL(dl) :: x2, y, f, g,
2274      REAL(dl), PARAMETER :: pi
2275
2276      !Expansions for high and
2277      IF(ABS(x)<=4.) THEN
2278
2279          x2=x*x
2280
2281          si8 = x*(1.d0+x2*(-4.
2282              +x2*(-1.410185368
2283                  +x2*(7.0824028227
2284                      (1.+x2*(1.0116214
2285                          x2*(1.55654986308
2286                          +x2*(3.2110705119
2287
2288          Si=si8
2289
2290          ELSE IF(ABS(x)>4.) THEN
2291
2292              y=1.d0/(x*x)
2293
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2294

```
2294      f = (1.d0 + y*(7.4443
2295          y*(2.377503101254
2296          + y*(6.4053383057
2297          y*(1.007951829803
2298          y*(-4.94701168645
2299          y*(1.978652470315
2300          y*(1.474789521929
2301          y*(7.085013081495
2302          y*(1.434685491715
2303
2304
2305      g = y*(1.d0 + y*(8.13
2306          y*(3.125575707957
2307          y*(1.090495284503
2308          y*(6.432916131430
2309          (1. + y*(8.195952
2310          + y*(2.2335554327
2311          + y*(1.1716472337
2312
2313      Si=pi/2.d0-f*cos(x)-g
2314
2315      END IF
2316
2317      END FUNCTION Si
2318
2319      FUNCTION Ci(x)
2320
2321      !Calculates the 'cosine i
2322      REAL :: Ci
2323      REAL, INTENT(IN) :: x
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2324

```
2324      REAL(dl) :: x2, y, f, g,  
2325      REAL(dl), PARAMETER :: em  
2326  
2327      !Expansions for high and  
2328      IF(ABS(x)<=4.) THEN  
2329          x2=x*x  
2330  
2331          ci8=em_const+log(x)+x  
2332              +x2*(1.0529736384  
2333                  +x2*(-9.937284888  
2334                      x2*(6.72126800814  
2335                          x2*(1.38536352772  
2336  
2337          Ci=ci8  
2338  
2339      ELSE IF(ABS(x)>4.) THEN  
2340          y=1./(x*x)  
2341  
2342          f = (1.d0 + y*(7.4443  
2343              y*(2.377503101254  
2344                  + y*(6.4053383057  
2345                      + y*(4.9481668819  
2346                          (x*(1. +y*(7.4643  
2347                              y*(2.415356701651  
2348                                  y*(4.585951158477  
2349                                      + y*(1.4346854917  
2350  
2351  
2352  
2353      g = y*(1.d0 + y*(8.13
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2354

```
2354          + y*(2.0629759514
2355          y*(1.090495284503
2356          y*(1.810044874646
2357          / (1. + y*(8.1959
2358          y*(3.260266616470
2359          + y*(1.3986671069

2360
2361          Ci=f*sin(x)-g*cos(x)
2362
2363          END IF
2364
2365          END FUNCTION Ci

2366
2367          FUNCTION derivative_table
2368
2369          !Takes the derivative y'(
2370          REAL :: derivative_table
2371          INTEGER, INTENT(IN) :: n
2372          REAL, INTENT(IN) :: x, xi
2373          REAL, ALLOCATABLE :: xta
2374          REAL :: a, b, c, d
2375          REAL :: x1, x2, x3, x4
2376          REAL :: y1, y2, y3, y4
2377          INTEGER :: i
2378          INTEGER, INTENT(IN) :: im
2379
2380          !This version interpolate
2381          !Care should be chosen to
2382          !Results from the interpo
2383
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2384

```
2384      !imeth = 1 => find x in x
2385      !imeth = 2 => find x in x
2386      !imeth = 3 => find x in x
2387
2388      !iorder = 1 => linear int
2389      !iorder = 2 => quadratic
2390      !iorder = 3 => cubic inte
2391
2392      ALLOCATE(xtab(n),ytab(n))
2393
2394      xtab=xin
2395      ytab=yin
2396
2397      IF(xtab(1)>xtab(n)) THEN
2398          !Reverse the arrays i
2399          CALL reverse(xtab,n)
2400          CALL reverse(ytab,n)
2401      END IF
2402
2403      IF(iorder==1) THEN
2404
2405          IF(n<2) ERROR STOP 'D
2406
2407          IF(x<=xtab(2)) THEN
2408
2409              x2=xtab(2)
2410              x1=xtab(1)
2411
2412              y2=ytab(2)
2413              y1=ytab(1)
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343



/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2414

```
2414
2415 ELSE IF (x>=xtab(n-1)
2416
2417 x2=xtab(n)
2418 x1=xtab(n-1)
2419
2420 y2=ytab(n)
2421 y1=ytab(n-1)
2422
2423 ELSE
2424
2425 i=table_integer(x
2426
2427 x2=xtab(i+1)
2428 x1=xtab(i)
2429
2430 y2=ytab(i+1)
2431 y1=ytab(i)
2432
2433 END IF
2434
2435 CALL fit_line(a,b,x1,
2436 derivative_table=a
2437
2438 ELSE IF(iorder==2) THEN
2439
2440 IF(n<3) ERROR STOP 'D
2441
2442 IF(x<=xtab(2) .OR. x>
2443
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2444



```
2444 IF(x<=xtab(2)) TH
2445
2446 x3=xtab(3)
2447 x2=xtab(2)
2448 x1=xtab(1)
2449
2450 y3=ytab(3)
2451 y2=ytab(2)
2452 y1=ytab(1)
2453
2454 ELSE IF (x>=xtab(
2455
2456 x3=xtab(n)
2457 x2=xtab(n-1)
2458 x1=xtab(n-2)
2459
2460 y3=ytab(n)
2461 y2=ytab(n-1)
2462 y1=ytab(n-2)
2463
2464 END IF
2465
2466 CALL fit_quadrati
2467
2468 derivative_table=
2469
2470 ELSE
2471
2472 i=table_integer(x
2473
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2474

```
2474          x1=xtab(i-1)
2475          x2=xtab(i)
2476          x3=xtab(i+1)
2477          x4=xtab(i+2)
2478          y1=ytab(i-1)
2479          y2=ytab(i)
2480          y3=ytab(i+1)
2481          y4=ytab(i+2)
2482
2483          ! In this case tak
2484          derivative_table=
2485
2486          CALL fit_quadrati
2487          derivative_table=
2488          CALL fit_quadrati
2489          derivative_table=
2490          CALL fit_quadrati
2491          derivative_table=
2492
2493
2494          END IF
2495
2496          ELSE IF(iorder==3) THEN
2497
2498          IF(n<4) ERROR STOP 'D
2499
2500          IF(x<=xtab(3)) THEN
2501
2502          x4=xtab(4)
2503          x3=xtab(3)
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343



/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2504

```
2504      x2=xtab(2)
2505      x1=xtab(1)
2506
2507      y4=ytab(4)
2508      y3=ytab(3)
2509      y2=ytab(2)
2510      y1=ytab(1)
2511
2512      ELSE IF (x>=xtab(n-2))
2513
2514      x4=xtab(n)
2515      x3=xtab(n-1)
2516      x2=xtab(n-2)
2517      x1=xtab(n-3)
2518
2519      y4=ytab(n)
2520      y3=ytab(n-1)
2521      y2=ytab(n-2)
2522      y1=ytab(n-3)
2523
2524      ELSE
2525
2526      i=table_integer(x)
2527
2528      x1=xtab(i-1)
2529      x2=xtab(i)
2530      x3=xtab(i+1)
2531      x4=xtab(i+2)
2532
2533      y1=ytab(i-1)
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2534

```
2534          y2=ytab(i)
2535          y3=ytab(i+1)
2536          y4=ytab(i+2)
2537
2538      END IF
2539
2540      CALL fit_cubic(a,b,c,
2541                      derivative_table=3.*a
2542
2543      ELSE
2544
2545      ERROR STOP 'DERIVATIV
2546
2547      END IF
2548
2549      END FUNCTION derivative_t
2550
2551      FUNCTION find(x,xin,yin,n
2552
2553!Given two arrays x and y
2554 IMPLICIT NONE
2555 REAL :: find
2556 INTEGER, INTENT(IN) :: n
2557 REAL, INTENT(IN) :: x, xi
2558 REAL, ALLOCATABLE :: xta
2559 REAL :: a, b, c, d
2560 REAL :: x1, x2, x3, x4
2561 REAL :: y1, y2, y3, y4
2562 INTEGER :: i
2563 INTEGER, INTENT(IN) :: io
```

/Users/lp1lopa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1lopa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2564

```
2564  
2565  
2566  
2567  
2568  
2569  
2570  
2571  
2572  
2573  
2574  
2575  
2576  
2577  
2578  
2579  
2580  
2581  
2582  
2583  
2584  
2585  
2586  
2587  
2588  
2589  
2590  
2591  
2592  
2593
```

!This version interpolate  
!Care should be chosen to  
!Results from the interpo

!If the value required is

!iorder = 1 => linear int  
!iorder = 2 => quadratic  
!iorder = 3 => cubic inte

!ifind = 1 => find x in x  
!ifind = 2 => find x in x  
!ifind = 3 => find x in x

!imeth = 1 => Uses cubic  
!imeth = 2 => Uses Lagran

ALLOCATE(xtab(n),ytab(n))

xtab=xin  
ytab=yin

IF(xtab(1)>xtab(n)) THEN  
!Reverse the arrays i  
CALL reverse(xtab,n)  
CALL reverse(ytab,n)  
END IF

IF(x<xtab(1)) THEN

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2594



```
2594 !Do a linear interpol
2595
2596
2597
2598
2599
2600
2601
2602
2603 IF(imeth==1) THEN
2604     CALL fit_line(a,b
2605         find=a*x+b
2606     ELSE IF(imeth==2) THE
2607         find=Lagrange_pol
2608     ELSE
2609         STOP 'FIND: Error
2610     END IF
2611
2612 ELSE IF(x>xtab(n)) THEN
2613
2614 !Do a linear interpol
2615
2616
2617
2618
2619
2620
2621
2622 IF(imeth==1) THEN
2623     CALL fit_line(a,b
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2624

```
2624      find=a*x+b
2625      ELSE IF(imeth==2) THE
2626          find=Lagrange_pol
2627      ELSE
2628          STOP 'FIND: Error
2629      END IF
2630
2631      ELSE IF(iorder==1) THEN
2632
2633          IF(n<2) STOP 'FIND: N
2634
2635          IF(x<=xtab(2)) THEN
2636
2637              x1=xtab(1)
2638              x2=xtab(2)
2639
2640              y1=ytab(1)
2641              y2=ytab(2)
2642
2643          ELSE IF (x>=xtab(n-1))
2644
2645              x1=xtab(n-1)
2646              x2=xtab(n)
2647
2648              y1=ytab(n-1)
2649              y2=ytab(n)
2650
2651      ELSE
2652
2653          i=table_integer(x)
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2654

```
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
```

```
x1=xtab(i)
x2=xtab(i+1)

y1=ytab(i)
y2=ytab(i+1)

END IF

IF(imeth==1) THEN
    CALL fit_line(a,b
    find=a*x+b
ELSE IF(imeth==2) THE
    find=Lagrange_pol
ELSE
    STOP 'FIND: Error
END IF

ELSE IF(iorder==2) THEN

    IF(n<3) STOP 'FIND: N
    IF(x<=xtab(2) .OR. x>
    IF(x<=xtab(2)) TH

        x1=xtab(1)
        x2=xtab(2)
        x3=xtab(3)
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2684

```
2684      y1=ytab(1)
2685      y2=ytab(2)
2686      y3=ytab(3)
2687      ELSE IF (x>=xtab(
2688      x1=xtab(n-2)
2689      x2=xtab(n-1)
2690      x3=xtab(n)
2691      2692
2693      2694      y1=ytab(n-2)
2695      2696      y2=ytab(n-1)
2697      2698      y3=ytab(n)
2699      END IF
2700      IF(imeth==1) THEN
2701          CALL fit_quad
2702          find=a*(x**2)
2703      ELSE IF(imeth==2)
2704          find=Lagrange
2705      ELSE
2706          STOP 'FIND: E
2707      END IF
2708      ELSE
2709      i=table_integer(x
2710
2711
2712
2713      x1=xtab(i-1)
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2714

```
2714          x2=xtab(i)
2715          x3=xtab(i+1)
2716          x4=xtab(i+2)
2717
2718          y1=ytab(i-1)
2719          y2=ytab(i)
2720          y3=ytab(i+1)
2721          y4=ytab(i+2)
2722
2723      IF(imeth==1) THEN
2724          !In this case
2725          CALL fit_quad
2726          find=(a*x**2+
2727          CALL fit_quad
2728          find=find+(a*
2729      ELSE IF(imeth==2)
2730          !In this case
2731          find=(Lagrang
2732      ELSE
2733          STOP 'FIND: E
2734      END IF
2735
2736      END IF
2737
2738      ELSE IF(iorder==3) THEN
2739
2740          IF(n<4) STOP 'FIND: N
2741
2742          IF(x<=xtab(3)) THEN
2743
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2744



```
2744 x1=xtab(1)
2745 x2=xtab(2)
2746 x3=xtab(3)
2747 x4=xtab(4)
2748
2749 y1=ytab(1)
2750 y2=ytab(2)
2751 y3=ytab(3)
2752 y4=ytab(4)
2753
2754 ELSE IF (x>=xtab(n-2))
2755
2756 x1=xtab(n-3)
2757 x2=xtab(n-2)
2758 x3=xtab(n-1)
2759 x4=xtab(n)
2760
2761 y1=ytab(n-3)
2762 y2=ytab(n-2)
2763 y3=ytab(n-1)
2764 y4=ytab(n)
2765
2766 ELSE
2767
2768 i=table_integer(x)
2769
2770 x1=xtab(i-1)
2771 x2=xtab(i)
2772 x3=xtab(i+1)
2773 x4=xtab(i+2)
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2774

```
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
```

y1=ytab(i-1)  
y2=ytab(i)  
y3=ytab(i+1)  
y4=ytab(i+2)

END IF

IF(imeth==1) THEN  
 CALL fit\_cubic(a,  
 find=a\*x\*\*3+b\*x\*\*  
ELSE IF(imeth==2) THE  
 find=Lagrange\_pol  
ELSE  
 STOP 'FIND: Error'  
END IF

ELSE  
 STOP 'FIND: Error, in  
END IF

END FUNCTION find

FUNCTION table\_integer(x,  
!Chooses between ways to  
IMPLICIT NONE  
INTEGER :: table\_integer

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2804

```
2804      INTEGER, INTENT(IN) :: n
2805      REAL, INTENT(IN) :: x, xt
2806      INTEGER, INTENT(IN) :: im
2807
2808      IF(imeth==1) THEN
2809          table_integer=linear_
2810      ELSE IF(imeth==2) THEN
2811          table_integer=search_
2812      ELSE IF(imeth==3) THEN
2813          table_integer=int_spl
2814      ELSE
2815          STOP 'TABLE INTEGER:'
2816      END IF
2817
2818      END FUNCTION table_int
2819
2820      FUNCTION linear_table_int
2821
2822!Assuming the table is ex
2823      IMPLICIT NONE
2824      INTEGER :: linear_table_i
2825      INTEGER, INTENT(IN) :: n
2826      REAL, INTENT(IN) :: x, xt
2827      REAL :: x1, x2, xn
2828      REAL, PARAMETER :: acc=1d
2829
2830      !Returns the integer (tab
2831      !eg. if x(3)=6. and x(4)=
2832      !Assumes table is organis
2833
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2834

```
2834 ! n=SIZE(xtab)
2835 x1=xtab(1)
2836 x2=xtab(2)
2837 xn=xtab(n)
2838
2839 IF(x1>xn) STOP 'LINEAR_TA'
2840 IF(ABS(-1.+float(n-1)*(x2
2841
2842 linear_table_integer=1+FL
2843
2844 END FUNCTION linear_table
2845
2846 FUNCTION search_int(x,xta
2847
2848 ! Does a stupid search thr
2849 IMPLICIT NONE
2850 INTEGER :: search_int
2851 INTEGER, INTENT(IN) :: n
2852 REAL, INTENT(IN) :: x, xt
2853 INTEGER :: i
2854
2855 IF(xtab(1)>xtab(n)) STOP
2856
2857 DO i=1,n
2858     IF(x>=xtab(i) .AND. x
2859 END DO
2860
2861 search_int=i
2862
2863 END FUNCTION search_int
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2864

```
2864  
2865  
2866  
2867  
2868  
2869  
2870  
2871  
2872  
2873  
2874  
2875  
2876  
2877  
2878  
2879  
2880  
2881  
2882  
2883  
2884  
2885  
2886  
2887  
2888  
2889  
2890  
2891  
2892  
2893
```

```
FUNCTION int_split(x,xtab  
!Finds the position of th  
IMPLICIT NONE  
INTEGER :: int_split  
INTEGER, INTENT(IN) :: n  
REAL, INTENT(IN) :: x, xt  
INTEGER :: i1, i2, imid  
IF(xtab(1)>xtab(n)) STOP  
i1=1  
i2=n  
DO  
    imid=NINT((i1+i2)/2.)  
    IF(x<xtab(imid)) THEN  
        i2=imid  
    ELSE  
        i1=imid  
    END IF  
    IF(i2==i1+1) EXIT  
END DO  
int_split=i1
```

/Users/lp1lopa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343



/Users/lp1lopa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2894

```
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
```

```
END FUNCTION int_split
```

```
SUBROUTINE fit_line(a1,a0
```

```
!Given xi, yi i=1,2 fits
IMPLICIT NONE
REAL, INTENT(OUT) :: a0,
REAL, INTENT(IN) :: x1, y
```

```
a1=(y2-y1)/(x2-x1)
a0=y1-a1*x1
```

```
END SUBROUTINE fit_line
```

```
SUBROUTINE fit_quadratic(
```

```
!Given xi, yi i=1,2,3 fit
IMPLICIT NONE
REAL, INTENT(OUT) :: a0,
REAL, INTENT(IN) :: x1, y
```

```
a2=((y2-y1)/(x2-x1)-(y3-y
a1=(y2-y1)/(x2-x1)-a2*(x2
a0=y1-a2*(x1**2)-a1*x1
```

```
END SUBROUTINE fit_quadrat
```

```
SUBROUTINE fit_cubic(a,b,
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2924

```
2924 !Given xi, yi i=1,2,3,4 f
2925 IMPLICIT NONE
2926 REAL, INTENT(OUT) :: a, b
2927 REAL, INTENT(IN) :: x1, y
2928 REAL :: f1, f2, f3
2929
2930 f1=(y4-y1)/((x4-x2)*(x4-x
2931 f2=(y3-y1)/((x3-x2)*(x3-x
2932 f3=(y2-y1)/((x2-x1)*(x4-x
2933
2934 a=f1-f2-f3
2935
2936 f1=(y3-y1)/((x3-x2)*(x3-x
2937 f2=(y2-y1)/((x2-x1)*(x3-x
2938 f3=a*(x3+x2+x1)
2939
2940 b=f1-f2-f3
2941
2942 f1=(y4-y1)/(x4-x1)
2943 f2=a*(x4**2+x4*x1+x1**2)
2944 f3=b*(x4+x1)
2945
2946 c=f1-f2-f3
2947
2948 d=y1-a*x1**3-b*x1**2-c*x1
2949
2950 END SUBROUTINE fit_cubic
2951
2952 FUNCTION Lagrange_polynom
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2954

```
2954 !Computes the result of t
2955 IMPLICIT NONE
2956 REAL :: Lagrange_polynomi
2957 REAL, INTENT(IN) :: x, xv
2958 REAL :: l(n+1)
2959 INTEGER, INTENT(IN) :: n
2960 INTEGER :: i, j
2961
2962 !Initialise variables, on
2963 Lagrange_polynomial=0.
2964 l=1.
2965
2966 !Loops to find the polyno
2967 DO i=0,n
2968     DO j=0,n
2969         IF(i .NE. j) l(i+
2970     END DO
2971     Lagrange_polynomial=L
2972 END DO
2973
2974 END FUNCTION Lagrange_pol
2975
2976 SUBROUTINE reverse(arry,n
2977
2978 !This reverses the conten
2979 IMPLICIT NONE
2980 INTEGER, INTENT(IN) :: n
2981 REAL, INTENT(INOUT) :: ar
2982 INTEGER :: i
2983 REAL, ALLOCATABLE :: hold
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343



/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 2984

```
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
```

ALLOCATE (hold(n))

hold=arry

DO i=1,n  
arry(i)=hold(n-i+1)  
END DO

DEALLOCATE (hold)

END SUBROUTINE reverse

SUBROUTINE fill\_growtab(c

!Fills a table of values  
TYPE(HM\_cosmology) :: cos  
INTEGER :: i  
REAL :: a, norm  
REAL, ALLOCATABLE :: d\_ta  
REAL :: ainit, amax, dini  
REAL :: acc  
INTEGER, PARAMETER :: n=6

The calculation should s  
!of starting in the g\pro  
ainit=0.001  
!Final should be a=1. unl  
amax=1.

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 3014

```
3014 !These set the initial co
3015 dinit=ainit
3016 vinit=1.
3017
3018 !Overall accuracy for the
3019 acc=1d-3
3020
3021 IF(HM_verbose) WRITE(*,*) 
3022 CALL ode_growth(d_tab,v_t)
3023 IF(HM_verbose) WRITE(*,*) 
3024
3025 !Normalise so that g(z=0)
3026 norm=find(1.,a_tab,d_tab,
3027 IF(HM_verbose) WRITE(*,*) 
3028 d_tab=d_tab/norm
3029
3030 !Could use some table-int
3031 IF(ALLOCATED(cosm%a_growt)
3032 IF(ALLOCATED(cosm%growth))
3033
3034 cosm%ng=n
3035 ALLOCATE(cosm%a_growth(n))
3036 DO i=1,n
3037     a=ainit+(amax-ainit)*
3038     cosm%a_growth(i)=a
3039     cosm%growth(i)=find(a
3040 END DO
3041
3042 IF(HM_verbose) WRITE(*,*) 
3043 IF(HM_verbose) WRITE(*,*) 
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 3044

3044  
3045  
3046  
3047  
3048  
3049  
3050  
3051  
3052  
3053  
3054  
3055  
3056  
3057  
3058  
3059  
3060  
3061  
3062  
3063  
3064  
3065  
3066  
3067  
3068  
3069  
3070  
3071  
3072  
3073

END SUBROUTINE fill\_growt

SUBROUTINE ode\_growth(x,v

!Solves 2nd order ODE x''  
IMPLICIT NONE  
REAL :: xi, ti, tf, dt, a  
REAL :: kx1, kx2, kx3, kx  
REAL(dl), ALLOCATABLE ::  
REAL, ALLOCATABLE :: x(:)  
INTEGER :: i, j, k, n, np  
TYPE(HM\_cosmology) :: cos  
INTEGER, PARAMETER :: jmax=1000  
INTEGER, PARAMETER :: nin=1000

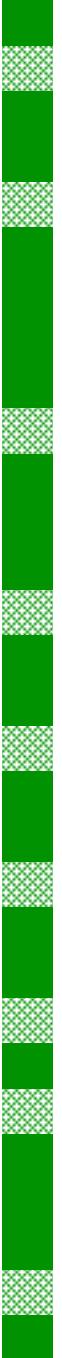
!xi and vi are the initial values  
!fx is what x' is equal to at xi  
!fv is what v' is equal to at xi  
!acc is the desired accuracy  
!imeth selects method

IF(ALLOCATED(x)) DEALLOCATE(x)  
IF(ALLOCATED(v)) DEALLOCATE(v)  
IF(ALLOCATED(t)) DEALLOCATE(t)

DO j=1,jmax

!Set the number of points  
n=ninit\*(2\*\*((j-1)))

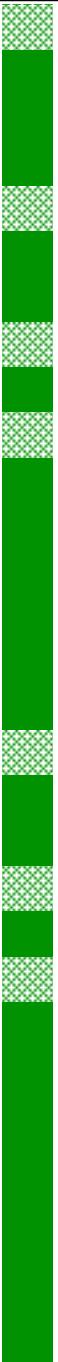
/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343



/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 3074

```
3074      n=n+1
3075
3076      !Allocate arrays
3077      ALLOCATE(x8(n),t8(n),
3078
3079      !Set the arrays to in
3080      x8=0.d0
3081      t8=0.d0
3082      v8=0.d0
3083
3084      !Set the intial condi
3085      x8(1)=xi
3086      v8(1)=vi
3087
3088      !Fill up a table for
3089      CALL fill_table8(DBLE
3090
3091      !Set the time interval
3092      dt=(tf-ti)/float(n-1)
3093
3094      !Intially fix this to
3095      ifail=0
3096
3097      DO i=1,n-1
3098
3099      x4=real(x8(i))
3100      v4=real(v8(i))
3101      t4=real(t8(i))
3102
3103      IF(imeth==1) THEN
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343



/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 3104

```
3104 !Crude method
3105 kx1=dt*fd(x4,
3106 kv1=dt*fv(x4,
3107
3108 x8(i+1)=x8(i)
3109 v8(i+1)=v8(i)
3110
3111 ELSE IF(imeth==2)
3112
3113 !Mid-point me
3114 !2017/06/18 -
3115 kx1=dt*fd(x4,
3116 kv1=dt*fv(x4,
3117 kx2=dt*fd(x4+
3118 kv2=dt*fv(x4+
3119
3120 x8(i+1)=x8(i)
3121 v8(i+1)=v8(i)
3122
3123 ELSE IF(imeth==3)
3124
3125 !4th order Ru
3126 kx1=dt*fd(x4,
3127 kv1=dt*fv(x4,
3128 kx2=dt*fd(x4+
3129 kv2=dt*fv(x4+
3130 kx3=dt*fd(x4+
3131 kv3=dt*fv(x4+
3132 kx4=dt*fd(x4+
3133
```

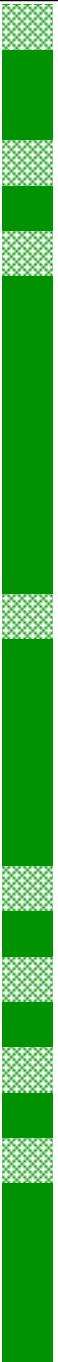
/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343



/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 3134

```
3134          kv4=dt*fv(x4+  
3135          x8(i+1)=x8(i)  
3136          v8(i+1)=v8(i)  
3137  
3138  
3139          END IF  
3140  
3141          !t8(i+1)=t8(i)+dt  
3142  
3143          END DO  
3144  
3145          IF(j==1) ifail=1  
3146  
3147          IF(j .NE. 1) THEN  
3148  
3149          np=1+(n-1)/2  
3150  
3151          DO k=1,1+(n-1)/2  
3152  
3153          kn=2*k-1  
3154  
3155          IF(ifail==0)  
3156  
3157          IF(xh(k)>  
3158          IF(vh(k)>  
3159  
3160          IF(ifail=  
3161          DEALL  
3162          EXIT  
3163          END IF
```

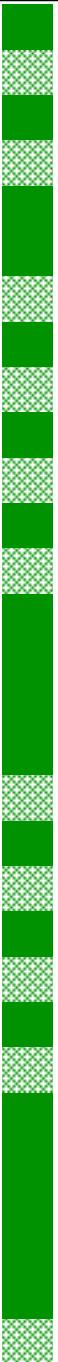
/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343



/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 3164

```
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193      END IF
          END DO
      END IF
      IF(ifail==0) THEN
          ALLOCATE(x(n),t(n)
          x=real(x8)
          v=real(v8)
          t=real(t8)
          EXIT
      END IF
      ALLOCATE(xh(n),th(n),
      xh=x8
      vh=v8
      th=t8
      DEALLOCATE(x8,t8,v8)
      END DO
  END SUBROUTINE ode_growth
  FUNCTION fv(d,v,k,a,cosm)
!v'=f(v) in ODE solver
  REAL :: fv
  REAL, INTENT(IN) :: d, v,
  REAL :: f1, f2, z
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343



/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 3194

```
3194      TYPE(HM_cosmology), INTEN
3195
3196      z=-1.+ (1./a)
3197
3198      f1=3.*Omega_m_hm(z,cosm)*
3199      f2=(2.+AH(z,cosm)/Hubble2
3200
3201      fv=f1-f2
3202
3203      END FUNCTION fv
3204
3205      FUNCTION fd(d,v,k,a,cosm)
3206
3207      !d'=f(d) in ODE solver
3208      REAL :: fd
3209      REAL, INTENT(IN) :: d, v,
3210      TYPE(HM_cosmology), INTEN
3211
3212      fd=v
3213
3214      END FUNCTION fd
3215
3216      FUNCTION AH(z,cosm)
3217
3218      !The Hubble acceleration
3219      REAL :: AH
3220      REAL, INTENT(IN) :: z
3221      REAL :: a
3222      TYPE(HM_cosmology), INTEN
3223
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 3224

```
3224 a=1./(1.+z)
3225 AH=cosm%om_m*(a**(-3.))+c
3226 AH=-AH/2.
3227
3228 END FUNCTION AH
3229
3230 !!AM End HMcode
3231
3232 subroutine PKequal(redshift
3233 !used by halofit_casarini
3234 implicit none
3235 real(dl) :: redshift,w_la_
3236 real(dl) :: z_star,tau_st
3237
3238 z_star=ThermoDerivedParam
3239 tau_star=TimeOfz(z_star)
3240 dlsb=TimeOfz(redshift)-ta
3241 w_true=w_lam
3242 wa_true=wa_ppf
3243 wa_ppf=0._dl
3244 do
3245 z_star=ThermoDerivedP
3246 tau_star=TimeOfz(z_st
3247 dlsb_eq=TimeOfz(redsh
3248 error=1.d0-dlsb_eq/dl
3249 if (abs(error).le.1e-
3250 w_lam=w_lam*(1+error)
3251 enddo
3252 w_hf=w_lam
3253 wa_hf=0._dl
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 343

```
343      end module NonLinear
344
345
346      !workaround for f90 circul
347      subroutine NonLinear_GetRa
348          use Transfer
349          use NonLinear
350          type(MatterPowerData) :: C
351
352          call NonLinear_GetNonLinRa
353
354      end subroutine NonLinear_G
355
356
357
358      subroutine NonLinear_GetRa
359          use Transfer
360          use NonLinear
361          type(MatterPowerData) :: C
362
363          stop 'Halofit module doesn
364
365      end subroutine NonLinear_G
```

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 3254

```
3254      w_lam=w_true
3255      wa_ppf=wa_true
3256      write(*,*)"at z = ",real(
3257
3258      end subroutine PKequal
3259
3260
3261
3262
3263
3264      !workaround for f90 circu
3265      subroutine NonLinear_GetR
3266          use Transfer
3267          use NonLinear
3268          type(MatterPowerData) :::
3269
3270          call NonLinear_GetNonLinR
3271
3272
3273
3274
3275
3276      subroutine NonLinear_GetR
3277          use Transfer
3278          use NonLinear
3279          type(MatterPowerData) :::
3280
3281      call MPIStop('Halofit mod
3282
3283
3284      end subroutine NonLinear_
```

/Users/lp1opa/Compare/camb\_simdata/halo  
fit\_ppf.f90, Top line: 366

366  
367

/Users/lp1opa/Compare/camb\_des/halofit\_  
ppf.f90, Top line: 3284

3284