

/Users/lplopa/Compare/camb\_simdata/power\_tilt.f90, Top line: 1

```
001      !This module provides the
002      !
003      !  $\ln P_s = \ln A_s + (n_s -$ 
004      !
005      ! so if  $n_{\text{run}} = 0$ ,  $n_{\text{ru}}$ 
006      !
007      !  $P_s = A_s (k/k_0_{\text{scalar}})$ 
008      !
009      !for the scalar spectrum,
010      !is a pivot scale, fixed h
011      !
012      !The tensor spectrum has t
013      !
014      !  $\ln P_t = \ln A_t + n_t \ln$ 
015      !
016      ! tensor_parameterization=
017      !
018      !  $A_t = r A_s$ 
019      !
020      ! tensor_parameterization=
021      !
022      !  $A_t = r P_s(k_0_{\text{tensor}})$ 
023      !
024      ! tensor_parameterization=
025      !
026      !  $A_t = \text{tensor\_amp}$ 
027      !
028      !The absolute normalizatio
029      !of the tensor and scalar
030      !
```

/Users/lplopa/Compare/camb\_des/power\_tilt.f90, Top line: 1

```
001      !This module provides the
002      !
003      !  $\ln P_s = \ln A_s + (n_s -$ 
004      !
005      ! so if  $n_{\text{run}} = 0$ ,  $n_{\text{ru}}$ 
006      !
007      !  $P_s = A_s (k/k_0_{\text{scalar}})$ 
008      !
009      !for the scalar spectrum,
010      !is a pivot scale, fixed h
011      !
012      !The tensor spectrum has t
013      !
014      !  $\ln P_t = \ln A_t + n_t \ln$ 
015      !
016      ! tensor_parameterization=
017      !
018      !  $A_t = r A_s$ 
019      !
020      ! tensor_parameterization=
021      !
022      !  $A_t = r P_s(k_0_{\text{tensor}})$ 
023      !
024      ! tensor_parameterization=
025      !
026      !  $A_t = \text{tensor\_amp}$ 
027      !
028      !The absolute normalizatio
029      !of the tensor and scalar
030      !
```

/Users/lp1opa/Compare/camb\_simdata/powe  
r\_tilt.f90, Top line: 31

```
031      !December 2003 - changed d
032      !April 2014 added differen
033
034      module InitialPower
035      use Precision
036
037      implicit none
038
039      private
040
041      character(LEN=*) , paramete
042
043      integer, parameter :: nnma
044      !Maximum possible number o
045
046      integer, parameter, public
047
048      Type InitialPowerParams
049      integer :: tensor_para
050      integer nn !Must have
051      !The actual number of
052
053      !For the default imple
054      real(dl) an(nnmax) !sc
055      real(dl) n_run(nnmax)
056      real(dl) n_runrun(nnma
057      real(dl) ant(nnmax) !t
058      real(dl) nt_run(nnmax)
059      real(dl) rat(nnmax) !r
060      real(dl) k_0_scalar, k
```

/Users/lp1opa/Compare/camb\_des/power\_ti  
lt.f90, Top line: 31

```
031      !December 2003 - changed d
032      !April 2014 added differen
033
034      module InitialPower
035      use Precision
036      use AMLutils
037      implicit none
038
039      private
040
041      character(LEN=*) , paramete
042
043      integer, parameter :: nnma
044      !Maximum possible number o
045
046      integer, parameter, public
047
048      Type InitialPowerParams
049      integer :: tensor_para
050      integer nn !Must have
051      !The actual number of
052
053      !For the default imple
054      real(dl) an(nnmax) !sc
055      real(dl) n_run(nnmax)
056      real(dl) n_runrun(nnma
057      real(dl) ant(nnmax) !t
058      real(dl) nt_run(nnmax)
059      real(dl) rat(nnmax) !r
060      real(dl) k_0_scalar, k
```

/Users/lp1opa/Compare/camb\_simdata/power\_tilt.f90, Top line: 60

```
060      real(dl) ScalarPowerAm
061      real(dl) TensorPowerAm
062      end Type InitialPowerParam
063
064      real(dl) curv !Curvature
065
066      Type(InitialPowerParams),
067
068      !Make things visible as ne
069
070      public InitialPowerParams,
071      public nnmax,Power_Descrip
072      contains
073
074
075      subroutine SetDefPowerPara
076      Type (InitialPowerParams)
077
078      AP%nn      = 1 !number of i
079      AP%an      = 1 !scalar spec
080      AP%n_run   = 0 !running of
081      AP%n_runrun = 0 !running
082      AP%ant     = 0 !tensor spec
083      AP%nt_run  = 0 !running o
084      AP%rat     = 1
085      AP%k_0_scalar = 0.05
086      AP%k_0_tensor = 0.05
087      AP%ScalarPowerAmp = 1
088      AP%TensorPowerAmp = 1
089      AP%tensor_parameterization
```

/Users/lp1opa/Compare/camb\_des/power\_tilt.f90, Top line: 61

```
061      real(dl) ScalarPowerAm
062      real(dl) TensorPowerAm
063      end Type InitialPowerParam
064
065      real(dl) curv !Curvature
066
067      Type(InitialPowerParams),
068
069      !Make things visible as ne
070
071      public InitialPowerParams,
072      public nnmax,Power_Descrip
073      contains
074
075
076      subroutine SetDefPowerPara
077      Type (InitialPowerParams)
078
079      AP%nn      = 1 !number of i
080      AP%an      = 1 !scalar spec
081      AP%n_run   = 0 !running of
082      AP%n_runrun = 0 !running
083      AP%ant     = 0 !tensor spec
084      AP%nt_run  = 0 !running o
085      AP%rat     = 1
086      AP%k_0_scalar = 0.05
087      AP%k_0_tensor = 0.05
088      AP%ScalarPowerAmp = 1
089      AP%TensorPowerAmp = 1
090      AP%tensor_parameterization
```

/Users/lp1opa/Compare/camb\_simdata/powe  
r\_tilt.f90, Top line: 90

```
090
091      end subroutine SetDefPower
092
093      subroutine InitializePower
094      Type (InitialPowerParams)
095      !Called before computing f
096      !Could read spectra from d
097
098      real(dl) acurv
099
100      if (AParamSet%nn > nnmax)
101          write (*,*) 'To use ',
102          write (*,*) 'nnmax in
103      end if
104      P = AParamSet
105
106      curv=acurv
107
108      !Write implementation spec
109
110      end subroutine InitializeP
111
112
113      function ScalarPower(k,ix)
114
115      !"ix" gives the index of t
116      !ScalarPower = const for s
117      !The normalization is defi
118      !scalar on co-moving hyper
119      ! < |D_a R^{{(3)}}|^2 > = in
```

/Users/lp1opa/Compare/camb\_des/power\_ti  
lt.f90, Top line: 91

```
091
092      end subroutine SetDefPower
093
094      subroutine InitializePower
095      Type (InitialPowerParams)
096      !Called before computing f
097      !Could read spectra from d
098
099      real(dl) acurv
100
101      if (AParamSet%nn > nnmax)
102          write (*,*) 'To use ',
103          write (*,*) 'nnmax in
104      end if
105      P = AParamSet
106
107      curv=acurv
108
109      !Write implementation spec
110
111      end subroutine InitializeP
112
113
114      function ScalarPower(k,ix)
115
116      !"ix" gives the index of t
117      !ScalarPower = const for s
118      !The normalization is defi
119      !scalar on co-moving hyper
120      ! < |D_a R^{{(3)}}|^2 > = in
```

/Users/lp1opa/Compare/camb\_simdata/power\_tilt.f90, Top line: 120

```
120      !In other words ScalarPowe
121      !-chi = \Phi + 2/3*\Omega^
122      !(w=p/\rho), so < |\chi(x)
123      !Near the end of inflation
124      !Here nu^2 = (k^2 + curv)/
125
126      !This power spectrum is al
127      !< |\Delta(x)|^2 > = \int
128      !For the isocurvature veloc
129
130      real(dl) ScalarPower,k, ln
131      integer ix
132
133      lnrat = log(k/P%k_0_scalar
134      ScalarPower=P%ScalarPowerA
135
136      !          ScalarPower = Sc
137
138      end function ScalarPower
139
140
141      function TensorPower(k,ix)
142
143      !TensorPower= const for sc
144      !The normalization is defi
145      ! < h_{ij}(x) h^{ij}(x) >
146      !for a closed model
147      ! < h_{ij}(x) h^{ij}(x) >
148      !for an open model
149      !"ix" gives the index of t
```

/Users/lp1opa/Compare/camb\_des/power\_tilt.f90, Top line: 121

```
121      !In other words ScalarPowe
122      !-chi = \Phi + 2/3*\Omega^
123      !(w=p/\rho), so < |\chi(x)
124      !Near the end of inflation
125      !Here nu^2 = (k^2 + curv)/
126
127      !This power spectrum is al
128      !< |\Delta(x)|^2 > = \int
129      !For the isocurvature veloc
130
131      real(dl) ScalarPower,k, ln
132      integer ix
133
134      lnrat = log(k/P%k_0_scalar
135      ScalarPower=P%ScalarPowerA
136
137      !          ScalarPower = Sc
138
139      end function ScalarPower
140
141
142      function TensorPower(k,ix)
143
144      !TensorPower= const for sc
145      !The normalization is defi
146      ! < h_{ij}(x) h^{ij}(x) >
147      !for a closed model
148      ! < h_{ij}(x) h^{ij}(x) >
149      !for an open model
150      !"ix" gives the index of t
```

/Users/lp1opa/Compare/camb\_simdata/power\_tilt.f90, Top line: 150

```
150      !Here nu^2 = (k^2 + 3*curv
151
152      real(dl) TensorPower,k
153      real(dl), parameter :: PiB
154      integer ix
155      real(dl) lnrat, k_dep
156
157      lnrat = log(k/P%k_0_tensor
158      k_dep = exp(lnrat*(P%ant(i
159      if (P%tensor_parameterizat
160          TensorPower = P%rat(ix
161      else if (P%tensor_paramete
162          TensorPower = P%rat(ix
163      else if (P%tensor_paramete
164          TensorPower = P%Tensor
165      end if
166      if (curv < 0) TensorPower=
167
168      end function TensorPower
169
170      !Get parameters describing
171      !Does not support running
172      function Power_Descript(in
173      character(LEN=8), intent(o
174      real(dl), intent(out) :: V
175      integer, intent(IN) :: in
176      logical, intent(IN) :: Sca
177      integer num, Power_Descrip
178      num=0
179      if (Scal) then
```

/Users/lp1opa/Compare/camb\_des/power\_tilt.f90, Top line: 151

```
151      !Here nu^2 = (k^2 + 3*curv
152
153      real(dl) TensorPower,k
154      real(dl), parameter :: PiB
155      integer ix
156      real(dl) lnrat, k_dep
157
158      lnrat = log(k/P%k_0_tensor
159      k_dep = exp(lnrat*(P%ant(i
160      if (P%tensor_parameterizat
161          TensorPower = P%rat(ix
162      else if (P%tensor_paramete
163          TensorPower = P%rat(ix
164      else if (P%tensor_paramete
165          TensorPower = P%Tensor
166      end if
167      if (curv < 0) TensorPower=
168
169      end function TensorPower
170
171      !Get parameters describing
172      !Does not support running
173      function Power_Descript(in
174      character(LEN=8), intent(o
175      real(dl), intent(out) :: V
176      integer, intent(IN) :: in
177      logical, intent(IN) :: Sca
178      integer num, Power_Descrip
179      num=0
180      if (Scal) then
```

/Users/lp1opa/Compare/camb\_simdata/power\_tilt.f90, Top line: 180

```
180      num=num+1
181      Keys(num) = 'n_s'
182      Vals(num) = P%an(in)
183      num=num+1
184      Keys(num) = 'n_run'
185      Vals(num) = P%n_run(in)
186      num=num+1
187      Keys(num) = 's_pivot'
188      Vals(num) = P%k_0_scal
189  end if
190  if (Tens) then
191      num=num+1
192      Keys(num) = 'n_t'
193      Vals(num) = P%ant(in)
194      num=num+1
195      Keys(num) = 't_pivot'
196      Vals(num) = P%k_0_tens
197      if (Scal) then
198          num=num+1
199          Keys(num) = 'p_rat'
200          Vals(num) = P%rat(in)
201      end if
202  end if
203  Power_Descript = num
204
205  end function Power_Descript
206
207  subroutine InitialPower_Reuse
208  use IniFile
209  Type(InitialPowerParams) :
```

/Users/lp1opa/Compare/camb\_des/power\_tilt.f90, Top line: 181

```
181      num=num+1
182      Keys(num) = 'n_s'
183      Vals(num) = P%an(in)
184      num=num+1
185      Keys(num) = 'n_run'
186      Vals(num) = P%n_run(in)
187      num=num+1
188      Keys(num) = 's_pivot'
189      Vals(num) = P%k_0_scal
190  end if
191  if (Tens) then
192      num=num+1
193      Keys(num) = 'n_t'
194      Vals(num) = P%ant(in)
195      num=num+1
196      Keys(num) = 't_pivot'
197      Vals(num) = P%k_0_tens
198      if (Scal) then
199          num=num+1
200          Keys(num) = 'p_rat'
201          Vals(num) = P%rat(in)
202      end if
203  end if
204  Power_Descript = num
205
206  end function Power_Descript
207
208  subroutine InitialPower_Reuse
209  use IniFile
210  Type(InitialPowerParams) :
```

/Users/lp1opa/Compare/camb\_simdata/powe  
r\_tilt.f90, Top line: 210

```
210      Type(TIniFile) :: Ini
211      logical, intent(in) :: Wan
212      integer i
213
214      InitPower%k_0_scalar = Ini
215      InitPower%k_0_tensor = Ini
216      InitPower%nn = Ini_Read_In
217      if (InitPower%nn>nnmax) st
218      if (WantTensors) then
219          InitPower%tensor_param
220          if (InitPower%tensor_p
221              & InitPower%tensor_par
222              & stop 'InitialPower:
223      end if
224      InitPower%rat(:) = 1
225      do i=1, InitPower%nn
226          InitPower%an(i) = Ini_
227          InitPower%n_run(i) = I
228          InitPower%n_runrun(i)
229
230          if (WantTensors) then
231              InitPower%ant(i) =
232              InitPower%nt_run(i
233              if (InitPower%tens
234                  InitPower%Tens
235              else
236                  InitPower%rat(
237              end if
238          end if
239
```

/Users/lp1opa/Compare/camb\_des/power\_ti  
lt.f90, Top line: 211

```
211      Type(TIniFile) :: Ini
212      logical, intent(in) :: Wan
213      integer i
214
215      InitPower%k_0_scalar = Ini
216      InitPower%k_0_tensor = Ini
217      InitPower%nn = Ini_Read_In
218      if (InitPower%nn>nnmax) ca
219      if (WantTensors) then
220          InitPower%tensor_param
221          if (InitPower%tensor_p
222              & InitPower%tensor
223              & call MpiStop('In
224      end if
225      InitPower%rat(:) = 1
226      do i=1, InitPower%nn
227          InitPower%an(i) = Ini_
228          InitPower%n_run(i) = I
229          InitPower%n_runrun(i)
230
231          if (WantTensors) then
232              InitPower%ant(i) =
233              InitPower%nt_run(i
234              if (InitPower%tens
235                  InitPower%Tens
236              else
237                  InitPower%rat(
238              end if
239          end if
240
```



/Users/lp1opa/Compare/camb\_simdata/power\_tilt.f90, Top line: 240

240	InitPower%ScalarPowerA
241	!Always need this as m
242	end do
243	
244	end subroutine InitialPow
245	
246	
247	end module InitialPower
248	

/Users/lp1opa/Compare/camb\_des/power\_tilt.f90, Top line: 241

241	InitPower%ScalarPowerA
242	!Always need this as m
243	end do
244	
245	end subroutine InitialPow
246	
247	
248	end module InitialPower
249	