```
(* date : 2017 May 23*)
(* author : Jinsu Kim *)
(* note :
 Primordial black holes in critical Higgs inflation. In this file,
all the parameters should be chosen at the
 top quark mass initially. For a parameter scan,
please refer to my another Mathematica file. One may use this file as
 a play ground. However, please do NOT make any change in this file.
*)
```

Primordial Black Holes in Critical Higgs Inflation
(* Mt = 170.845 GeV *)

```
Quit[];
```

Initial conditions at $\mu = M_t$

```
(* PDG2016 & 1307.3536 *)

Mt = 170.845`;

Mp = 1.221 × 10^19;
MpR = 2.4 * 10^(18); (*reduced Planck mass*)
MW = 80.385;
MZ = 91.1876;
MH = 125.09;
alphasMZ = 0.1182;
lHMt[mt_] := 0.12604 + 0.00206 (MH - 125.15) - 0.00004 (mt - 173.34);
ytMt[mt_] := 0.93690 + 0.00556 (mt - 173.34) -
    0.00003 (MH - 125.15) - 0.00042 * (alphasMZ - 0.1184) / 0.0007;
g3Mt[mt_] := 1.1666 + 0.00314 * (alphasMZ - 0.1184) / 0.0007 - 0.00046 (mt - 173.34);
g2Mt[mt_] := 0.64779 + 0.00004 (mt - 173.34) + 0.00011 * (MW - 80.384) / 0.014;
g1Mt[mt_] := 0.35830 + 0.00011 (mt - 173.34) - 0.00020 * (MW - 80.384) / 0.014;
(* mr c++ library: *)
(*g1Mt=0.41382910059588607`*)
(*g2Mt=0.648116*)
(*g3Mt=1.16528*)
(*ytMt=0.9352*)
(*lHMt=0.125886*)
```

```
(*xihMt=59.218982012703925`;*)
xihMt = 1500
```

```
1500
```

RGE

Renormalisation scale

```
(*t[μ_]:=Log[μ/Mt]*)
```

$$h[t\_] := \left(yt[t] \Big/ \sqrt{2.}\right)^{-1} mt * Exp[t];$$

Suppression factor

$$sh[t\_] := \frac{1 + xihMt * h[t]^2 / MpR^2}{1 + (1 + 6\, xihMt)\, xihMt * h[t]^2 / MpR^2};$$

1-loop beta functions

$$betalH1[t\_] := 6\left(1 + 3\, sh[t]^2\right) lH[t]^2 + 12\, lH[t]\, yt[t]^2 - 6\, yt[t]^4 - 3\, lH[t]\left(3\, g2[t]^2 + g1[t]^2\right) + \frac{3}{8}\left(2\, g2[t]^4 + \left(g1[t]^2 + g2[t]^2\right)^2\right);$$

$$betag11[t\_] := \frac{81 + sh[t]}{12} g1[t]^3;$$

$$betag21[t\_] := -\frac{39 - sh[t]}{12} g2[t]^3;$$

$$betag31[t\_] := -7\, g3[t]^3;$$

$$betayt1[t\_] := yt[t] * \left(\left(\frac{23}{6} + \frac{2}{3} sh[t]\right) yt[t]^2 - \left(8\, g3[t]^2 + \frac{9}{4} g2[t]^2 + \frac{17}{12} g1[t]^2\right)\right);$$

$$betaxih1[t\_] := \left(6\left(1 + sh[t]\right) lH[t] + 6\, yt[t]^2 - \frac{3}{2}\left(g1[t]^2 + 3\, g2[t]^2\right)\right)\left(xihMt + 1/6\right);$$

$$gamma1[t\_] := -\left(\frac{9}{4} g2[t]^2 + \frac{3}{4} g1[t]^2 - 3\, yt[t]^2\right);$$

2-loop beta functions

```
betalH2[t_] :=
  1/48 ((912 + 3 sh[t]) g2[t]^6 - (290 - sh[t]) g1[t]^2 g2[t]^4 - (560 - sh[t]) g1[t]^4 g2[t]^2 -
      (380 - sh[t]) g1[t]^6) + (38 - 8 sh[t]) yt[t]^6 -
   yt[t]^4 (8/3 g1[t]^2 + 32 g3[t]^2 + (12 - 117 sh[t] + 108 sh[t]^2) lH[t]) +
   lH[t] * (-1/8 (181 + 54 sh[t] - 162 sh[t]^2) g2[t]^4 +
      1/4 (3 - 18 sh[t] + 54 sh[t]^2) g1[t]^2 g2[t]^2 + 1/24 (90 + 377 sh[t] + 162 sh[t]^2) g1[t]^4 +
      (27 + 54 sh[t] + 27 sh[t]^2) g2[t]^2 lH[t] + (9 + 18 sh[t] + 9 sh[t]^2) g1[t]^2 lH[t] -
      (48 + 288 sh[t] - 324 sh[t]^2 + 624 sh[t]^3 - 324 sh[t]^4) lH[t]^2) +
   yt[t]^2 (-9/4 g2[t]^4 + 21/2 g1[t]^2 g2[t]^2 - 19/4 g1[t]^4 +
      lH[t] * (45/2 g2[t]^2 + 85/6 g1[t]^2 + 80 g3[t]^2 - (36 + 108 sh[t]^2) lH[t]));

betag12[t_] := 199/18 g1[t]^5 + 9/2 g1[t]^3 g2[t]^2 + 44/3 g1[t]^3 g3[t]^2 - 17/6 sh[t] * g1[t]^3 yt[t]^2;
betag22[t_] := 3/2 g1[t]^2 g2[t]^3 + 35/6 g2[t]^5 + 12 g2[t]^3 g3[t]^2 - 3/2 sh[t] * g2[t]^3 yt[t]^2;
betag32[t_] := 11/6 g1[t]^2 g3[t]^3 + 9/2 g2[t]^2 g3[t]^3 - 26 g3[t]^5 - 2 sh[t] * g3[t]^3 yt[t]^2;

betayt2[t_] :=
  yt[t] * (-23/4 g2[t]^4 - 3/4 g1[t]^2 g2[t]^2 + 1187/216 g1[t]^4 + 9 g2[t]^2 g3[t]^2 + 19/9 g1[t]^2 g3[t]^2 -
     108 g3[t]^4 + (225/16 g2[t]^2 + 131/16 g1[t]^2 + 36 g3[t]^2) sh[t] * yt[t]^2 +
     6 (-2 sh[t]^2 yt[t]^4 - 2 sh[t]^3 yt[t]^2 lH[t] + sh[t]^2 lH[t]^2));

gamma2[t_] := -(271/32 g2[t]^4 - 9/16 g1[t]^2 g2[t]^2 - 431/96 sh[t] * g1[t]^4 -
     5/2 (9/4 g2[t]^2 + 17/12 g1[t]^2 + 8 g3[t]^2) yt[t]^2 + 27/4 sh[t] * yt[t]^4 - 6 sh[t]^3 lH[t]^2);
```

RGEs

$$\text{betalH}[t\_] := \frac{1}{16\,\pi^2}\,\text{betalH1}[t] + \frac{1}{(16\,\pi^2)^2}\,\text{betalH2}[t];$$

$$\text{betag1}[t\_] := \frac{1}{16\,\pi^2}\,\text{betag11}[t] + \frac{1}{(16\,\pi^2)^2}\,\text{betag12}[t];$$

$$\text{betag2}[t\_] := \frac{1}{16\,\pi^2}\,\text{betag21}[t] + \frac{1}{(16\,\pi^2)^2}\,\text{betag22}[t];$$

$$\text{betag3}[t\_] := \frac{1}{16\,\pi^2}\,\text{betag31}[t] + \frac{1}{(16\,\pi^2)^2}\,\text{betag32}[t];$$

$$\text{betayt}[t\_] := \frac{1}{16\,\pi^2}\,\text{betayt1}[t] + \frac{1}{(16\,\pi^2)^2}\,\text{betayt2}[t];$$

$$\text{betaxih}[t\_] := \frac{1}{16\,\pi^2}\,\text{betaxih1}[t];$$

$$\text{gamma}[t\_] := \frac{1}{16\,\pi^2}\,\text{gamma1}[t] + \frac{1}{(16\,\pi^2)^2}\,\text{gamma2}[t];$$

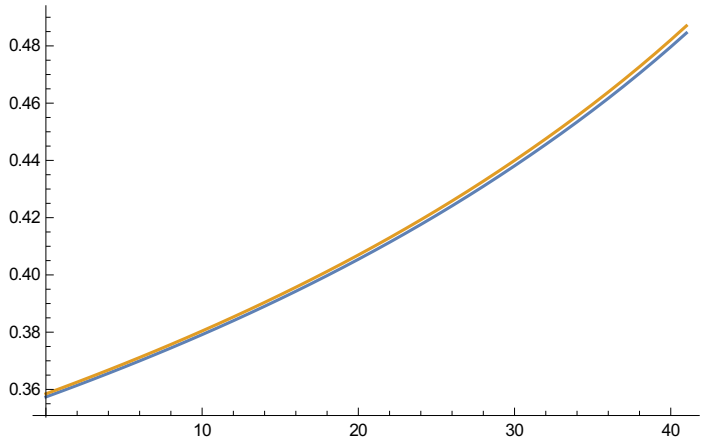
Solving the RGEs

```
mustart = 170;
muend = 10 * Mp;
tmin = 0;
tmax = Log[muend / mustart];

RGEsol = ParametricNDSolve[{
    g1'[t] == betag1[t] / (1 + gamma[t]),
    g2'[t] == betag2[t] / (1 + gamma[t]),
    g3'[t] == betag3[t] / (1 + gamma[t]),
    lH'[t] == betalH[t] / (1 + gamma[t]),
    yt'[t] == betayt[t] / (1 + gamma[t]),
    g1[tmin] == g1Mt[mt],
    g2[tmin] == g2Mt[mt],
    g3[tmin] == g3Mt[mt],
    lH[tmin] == lHMt[mt],
    yt[tmin] == ytMt[mt]
   }, {g1, g2, g3, yt, lH}, {t, tmin, tmax}, mt,
   AccuracyGoal → Automatic, PrecisionGoal → Automatic, MaxSteps → Infinity];
```

```
g1sol[mt_, t_] := g1[mt][t] /. Flatten[RGEsol];
g2sol[mt_, t_] := g2[mt][t] /. Flatten[RGEsol];
g3sol[mt_, t_] := g3[mt][t] /. Flatten[RGEsol];
ytsol[mt_, t_] := yt[mt][t] /. Flatten[RGEsol];
hsol[mt_, t_] := (ytsol[mt, t] / Sqrt[2.])^-1 * mt * Exp[t];
lHsol[mt_, t_] := lH[mt][t] /. Flatten[RGEsol];
xihsol := xihMt;
shsol[mt_, t_] := (1 + xihsol * hsol[mt, t]^2 / MpR^2) / (1 + (1 + 6 xihsol) xihsol * hsol[mt, t]^2 / MpR^2);
gamma1sol[mt_, t_] := -(9/4 g2sol[mt, t]^2 + 3/4 g1sol[mt, t]^2 - 3 ytsol[mt, t]^2);
gamma2sol[mt_, t_] :=
  -(271/32 g2sol[mt, t]^4 - 9/16 g1sol[mt, t]^2 g2sol[mt, t]^2 - 431/96 shsol[mt, t] *
      g1sol[mt, t]^4 - 5/2 (9/4 g2sol[mt, t]^2 + 17/12 g1sol[mt, t]^2 + 8 g3sol[mt, t]^2)
      ytsol[mt, t]^2 + 27/4 shsol[mt, t] * ytsol[mt, t]^4 - 6 shsol[mt, t]^3 lHsol[mt, t]^2);
gammasol[mt_, t_] := 1/(16 π^2) gamma1sol[mt, t] + 1/(16 π^2)^2 gamma2sol[mt, t];

ht[mt_, t_] := D[hsol[mt, t], t] /. Flatten[RGEsol];
httest[mt_, t_] := D[hsol[mt, x], x] /. {x → t};

Needs["NumericalCalculus`"];
```

Simple Plot

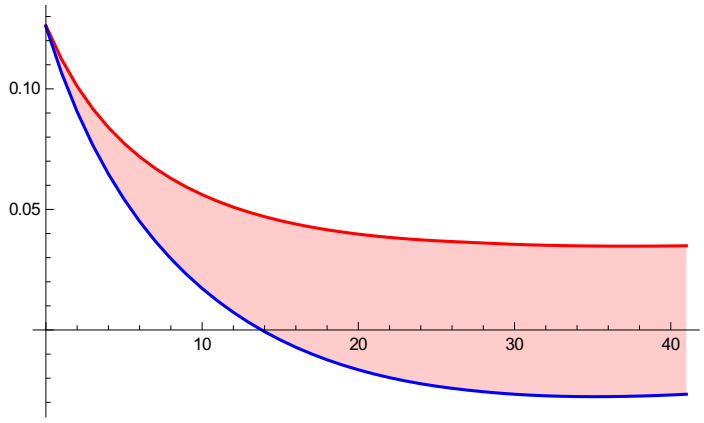
```
g1Tab1 = Table[{t, g1sol[165, t]}, {t, tmin, tmax}];
g2Tab1 = Table[{t, g2sol[165, t]}, {t, tmin, tmax}];
g3Tab1 = Table[{t, g3sol[165, t]}, {t, tmin, tmax}];

g1Tab2 = Table[{t, g1sol[175, t]}, {t, tmin, tmax}];

ListLinePlot[{g1Tab1, g1Tab2}, PlotRange → All]
```



```
lHTab1 = Table[{t, lHsol[165, t]}, {t, tmin, tmax}];
lHTab2 = Table[{t, lHsol[175, t]}, {t, tmin, tmax}];

padIt[v_?(Element[#, Reals] &), f_List] :=
  OutputForm[AccountingForm[Chop[v], f, NumberSigns → {"-", ""},
    NumberPadding → {"0", "0"}, SignPadding → True]];
Newsol[mt_, t_] := padIt[N[lHsol[mt, t], 15], {16, 16}];

lHTab = Table[
   {ParallelTable[{t, mt, Newsol[mt, t]}, {t, tmin, tmax, 0.5}]}, {mt, 165, 175}];

Export["/home/rakhan/lHTab.csv", lHTab]
```

```
/home/rakhan/lHTab.csv
```

```
ListLinePlot[{lHTab1, lHTab2}, Filling → {1 → {2}},
 PlotRange → All, PlotStyle → {Red, Blue}]
```



```
Table[{mt, FindRoot[lHsol[mt, t] == 0, {t, 20}][[1, 2]]}, {mt, 165, 175, 1}]
```

InterpolatingFunction::dmval :
Input value {70.6683} lies outside the range of data in the interpolating function. Extrapolation will be used. »

InterpolatingFunction::dmval :
Input value {–646.014} lies outside the range of data in the interpolating function. Extrapolation will be used. »

InterpolatingFunction::dmval :
Input value {–1.} lies outside the range of data in the interpolating function. Extrapolation will be used. »

General::stop : Further output of InterpolatingFunction::dmval will be suppressed during this calculation. »

FindRoot::lstol :
The line search decreased the step size to within tolerance specified by AccuracyGoal and PrecisionGoal but was unable to find a sufficient decrease in the merit function. You may need more than MachinePrecision digits of working precision to meet these tolerances. »

FindRoot::lstol :
The line search decreased the step size to within tolerance specified by AccuracyGoal and PrecisionGoal but was unable to find a sufficient decrease in the merit function. You may need more than MachinePrecision digits of working precision to meet these tolerances. »

FindRoot::lstol :
The line search decreased the step size to within tolerance specified by AccuracyGoal and PrecisionGoal but was unable to find a sufficient decrease in the merit function. You may need more than MachinePrecision digits of working precision to meet these tolerances. »

General::stop : Further output of FindRoot::lstol will be suppressed during this calculation. »

```
{{165, 37.1877}, {166, 36.5395}, {167, 36.0543},
 {168, 35.6936}, {169, 35.4308}, {170, 35.2469}, {171, 29.6117},
 {172, 21.3666}, {173, 17.812}, {174, 15.4996}, {175, 13.8057}}
```

```
Newsol[170.8555, 35.3055]
```

```
00.0000648403008521
```

```
lHTabl = Table[{ParallelTable[{t, mt, Newsol[mt, t]}, {t, tmin, tmax, 0.01}]},
   {mt, 170, 171, 0.01}];
```

```
Export["/home/rakhan/lHTabl.csv", lHTabl, "csv"]
```

```
/home/rakhan/lHTabl.csv
```

```
padIt[ND[lHsol[170, x], x, 20], {16, 16}]
```

```
-0.0013169920227875
```

```
lHdTab = Table[{ParallelTable[{t, mt, padIt[ND[lHsol[mt, x], x, t], {16, 16}]},
     {t, tmin, tmax, 0.5}]}, {mt, 165, 175}];
```

InterpolatingFunction::dmval :
Input value {41.5} lies outside the range of data in the interpolating function. Extrapolation will be used. »

InterpolatingFunction::dmval :
Input value {42.} lies outside the range of data in the interpolating function. Extrapolation will be used. »

InterpolatingFunction::dmval :
Input value {41.5} lies outside the range of data in the interpolating function. Extrapolation will be used. »

General::stop : Further output of InterpolatingFunction::dmval will be suppressed during this calculation. »

```
Export["/home/rakhan/lHdTab.csv", lHdTab, "csv"]
```

```
/home/rakhan/lHdTab.csv
```

```
lHdTabl = Table[{ParallelTable[{t, mt, padIt[ND[lHsol[mt, x], x, t], {16, 16}]},
      {t, tmin, tmax, 0.01}]}, {mt, 170, 171, 0.01}];
```

```
InterpolatingFunction::dmval :
  Input value {41.12} lies outside the range of data in the interpolating function. Extrapolation will be used. >>
InterpolatingFunction::dmval :
  Input value {41.13} lies outside the range of data in the interpolating function. Extrapolation will be used. >>
InterpolatingFunction::dmval :
  Input value {41.14} lies outside the range of data in the interpolating function. Extrapolation will be used. >>
General::stop : Further output of InterpolatingFunction::dmval will be suppressed during this calculation. >>
```

```
Export["/home/rakhan/lHdTabl.csv", lHdTabl, "csv"]
```

```
/home/rakhan/lHdTabl.csv
```

```
LHdTab = Table[{ParallelTable[{t, mt, padIt[ND[lHsol[mt, x], x, t], {16, 16}]},
      {t, 33, 37, 0.0001}]}, {mt, 170.85, 170.86, 0.0001}];
```

```
Export["/home/rakhan/LHdTab.csv", LHdTab, "csv"]
```

```
/home/rakhan/LHdTab.csv
```

```
hsol[170, 30]
```

$$6.13616 \times 10^{15}$$

```
gammasol[170, 30]
```

```
-0.00155165
```

RG-improved effective potential

```
GHexact[mt_?NumberQ, t_?NumberQ] :=
  Exp[-NIntegrate[gammasol[mt, x] / (gammasol[mt, x] + 1),
      {x, 0, t}, AccuracyGoal → Automatic, PrecisionGoal → Automatic,
      Method → {Automatic, "SymbolicProcessing" → 0}]];

tstep = (tmax - tmin) / 100.;
GHlist[mt_] := Table[{x, GHexact[mt, t] /. {t → x}}, {x, tmin, tmax, tstep}];
GHintpln[mt_, t_] := Interpolation[GHlist[mt]][t];
VJordanexact[mt_, t_] := (1. / 4) * lHsol[mt, t] * GHexact[mt, t]^4 * hsol[mt, t]^4;
VJordan[mt_, t_] := (1. / 4) * lHsol[mt, t] * GHintpln[mt, t]^4 * hsol[mt, t]^4;
```

```
NMC[mt_, t_] := 1 + xihsol * GHintpln[mt, t]^2 hsol[mt, t]^2 / MpR^2;
NMCt[mt_, t_] := D[NMC[mt, x], x] /. x → t;
NMCh[mt_, t_] := NMCt[mt, t] / ht[t];
VE[mt_, t_] := VJordan[mt, t] / NMC[mt, t]^2;
VEt[t_] := D[VE[x], x] /. x → t;
VEh[t_] := VEt[t] / ht[t];
VEht[t_] := D[VEh[x], x] /. x → t;
VEhh[t_] := VEht[t] / ht[t];
```

Potential

```
Newhsol[mt_, t_] := OutputForm[AccountingForm[hsol[mt, t] / MpR]];
NewVE[mt_, t_] := OutputForm[AccountingForm[VE[mt, t] / MpR^4]];
```

```
PoTab = Table[{ParallelTable[{t, mt, Newhsol[mt, t], NewVE[mt, t]},
      {t, 0, Log[1 * Mp / mustart], 0.1}]}, {mt, 165, 176}];
```

```
NewVE[169, 30]
```

```
0.000000000000114307
```

```
Export["/home/rakhan/PoTab.csv", PoTab, "csv"]
```

```
/home/rakhan/PoTab.csv
```

```
Log[1 * Mp / mustart]
```

```
38.813
```

```
Newhsol[170, 30]
```

```
0.00255673
```

```
SetSharedVariable[j]
```

```
Monitor[ParallelTable[
   {j = t, Newhsol[170.85, t], NewVE[170.85, t]}, {t, tmin, tmax, 0.1}], j];
```

```
POTab = Table[{ParallelTable[{t, mt, Newhsol[mt, t], NewVE[mt, t]},
      {t, 0, Log[1 * Mp / mustart], 0.1}]}, {mt, 170.8552, 170.8553, 0.0001}];
```

```
Export["/home/rakhan/POTab.csv", POTab, "csv"]
```

```
/home/rakhan/POTab.csv
```

```
Potential = Table[{ParallelTable[{t, mt, Newhsol[mt, t], NewVE[mt, t]},
      {t, 0, Log[1 * Mp / mustart], 0.1}]}, {mt, 170.844, 170.856, 0.001}];
```

```
Export["/home/rakhan/Potential.csv", Potential, "csv"]
```

```
/home/rakhan/Potential.csv
```

```
PotFinal = Table[{ParallelTable[{t, mt, Newhsol[mt, t], NewVE[mt, t]},
      {t, 0, Log[1 * Mp / mustart], 0.01}]}, {mt, 170.848, 170.856, 0.001}];
```

```
Export["/home/rakhan/PotFinal.csv", PotFinal, "csv"]
```

```
/home/rakhan/PotFinal.csv
```

```
PotFinal2 = Table[{ParallelTable[{t, mt, Newhsol[mt, t], NewVE[mt, t]},
      {t, 0, Log[1 * Mp / mustart], 0.1}]}, {mt, 170.849, 170.851, 0.0001}];
Export["/home/rakhan/PotFinal.csv", PotFinal2, "csv"]
```

```
/home/rakhan/PotFinal.csv
```

Non necessary

$$\text{dchidh}[t\_] := \text{Sqrt}\left[\text{GHintpln}[t]^2 / \text{NMC}[t] + 3 \frac{\text{MpR}^2}{2\,\text{NMC}[t]^2} (\text{NMCh}[t])^2\right];$$

```
NMCtest[t_] := 1 + xihsol[t] 1^2 hsol[t]^2 / MpR^2;
NMCttest[t_] := D[NMCtest[x], x] /. x → t;
NMChtest[t_] := NMCttest[t] / ht[t];
```

$$\text{dchidhtest}[t\_] := \text{Sqrt}\left[1^2 / \text{NMCtest}[t] + 3 \frac{\text{MpR}^2}{2\,\text{NMCtest}[t]^2} (\text{NMChtest}[t])^2\right];$$

```
VEchi[t_] := VEh[t] / dchidh[t];
VEchit[t_] := D[VEchi[x], x] /. x → t;
VEhchi[t_] := VEchit[t] / ht[t];
VEchichi[t_] := VEhchi[t] / dchidh[t];
VEchichit[t_] := D[VEchichi[x], x] /. x → t;
VEchichih[t_] := VEchichit[t] / ht[t];
VEchichichi[t_] := VEchichih[t] / dchidh[t];
```

Cosmological observables

```
α = -0.7296;
```

```
ϵV[t_] := MpR^2 / 2 (VEchi[t] / VE[t])^2;
ηV[t_] := MpR^2 (VEchichi[t] / VE[t]);
ξV2[t_] := MpR^4 ((VEchi[t] VEchichichi[t]) / VE[t]^2);
```

```
Ps[t_] := 1 / (24 * Pi^2 * MpR^4) * VE[t] / ϵV[t];
ns[t_] := 1 - 6 ϵV[t] + 2 ηV[t] - 2/3 (5 + 36 α) ϵV[t]^2 +
    2 (-1 + 8 α) ϵV[t] ηV[t] + 2 ηV[t]^2 / 3 + (2/3 - 2 α) ξV2[t];
r[t_] := 16 ϵV[t] - 64 ϵV[t]^2 / 3 + 64 α ϵV[t]^2 + 32 ϵV[t] ηV[t] / 3 - 32 α ϵV[t] ηV[t];
dnsdlnk[t_] := -24 ϵV[t]^2 + 16 ϵV[t] ϵV[t] - 2 ξV2[t];
```

```
Nef[tpivot_?NumberQ, tend_?NumberQ] :=
  NIntegrate[1 / MpR^2 (dchidh[x9]) (ht[x9]) (VE[x9] / VEchi[x9]),
   {x9, tend, tpivot}, AccuracyGoal → Automatic, PrecisionGoal → Automatic,
   Method → {Automatic, "SymbolicProcessing" → 0}];
Treh = 1.0 * 10^(15);
gstar = 100;
Nefpivot[VI_, Vend_, kpivot_] := 62. - Log[kpivot / (0.67 / 3000)] +
   Log[VI^(1 / 4) / 10^(16)] + Log[VI^(1 / 4) / Vend^(1 / 4)] -
   (1 / 3.) * Log[(Vend^(1 / 4)) * (Pi^2 / 30 * gstar * Treh^(4))^(-1 / 4)];


kpivot002 = 0.002;
kpivot05 = 0.05;

Clear[hend, VIt, kpivott, hpivot05, VI05];
tend = t /. FindRoot[ϵV[t] == 1., {t, tmin, tmax},
     AccuracyGoal → Automatic, PrecisionGoal → Automatic][[1]];
hend = hsol[t] /. t → tend;
VIt = VE[t] /. t → Log[MpR / mustart];
Vend = VE[t] /. t → tend;
kpivott = kpivot05;
tpivot05 = xxx /. Quiet[
    FindRoot[Nef[xxx, tend] == Nefpivot[VIt, Vend, kpivott], {xxx, tmin, tend}]];
hpivot05 = hsol[t] /. t → tpivot05;
VI05 = VE[tpivot05];
```

FindRoot::nlnum :

The function value $\left\{-1. + \left(2.88 \times 10^{36}\, VE'[0.]^2\right) \Big/ \left(ht[0.]^2 \left(\frac{GHintpln[\ll 1 \gg]^2}{NMC[\ll 1 \gg]} + \frac{8.64 \times 10^{36}\, NMCh[\ll 1 \gg]^2}{NMC[\ll 1 \gg]^2}\right) VE[0.]^2\right)\right\}$ is not a list of numbers with dimensions {1} at {t} = {0.}. »

ReplaceAll::reps :

$\left\{\frac{2.88 \times 10^{36}\, VE'[t]^2}{ht[t]^2 \left(\frac{GHintpln[\ll 1 \gg]^2}{NMC[\ll 1 \gg]} + \frac{8.64 \times 10^{36}\, NMCh[\ll 1 \gg]^2}{NMC[\ll 1 \gg]^2}\right) VE[t]^2} == 1.\right\}$ is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. »

ReplaceAll::reps :

$\left\{\frac{2.88 \times 10^{36}\, VE'[t]^2}{ht[t]^2 \left(\frac{GHintpln[\ll 1 \gg]^2}{NMC[\ll 1 \gg]} + \frac{8.64 \times 10^{36}\, NMCh[\ll 1 \gg]^2}{NMC[\ll 1 \gg]^2}\right) VE[t]^2} == 1.\right\}$ is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. »

FindRoot::srect : Value $t /. \frac{2.88 \times 10^{36}\, VE'[t]^2}{ht[t]^2 \left(\frac{GHintpln[\ll 1 \gg]^2}{NMC[\ll 1 \gg]} + \frac{8.64 \times 10^{36}\, NMCh[\ll 1 \gg]^2}{NMC[\ll 1 \gg]^2}\right) VE[t]^2} == 1.$ in search specification {xxx, tmin, tend} is not a number or array of numbers. »

ReplaceAll::reps :

{FindRoot[Nef[xxx, tend] == Nefpivot[VIt, Vend, kpivott], {xxx, tmin, tend}]} is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. »

FindRoot::srect : Value $t /. \frac{2.88\times 10^{36}\, VE'[t]^2}{ht[t]^2 \left(\frac{GHintpln[\ll 1 \gg]^2}{NMC[\ll 1 \gg]} + \frac{8.64\times 10^{36}\, NMCh[\ll 1 \gg]^2}{NMC[\ll 1 \gg]^2}\right) VE[t]^2} == 1.$ in search specification {xxx, tmin, tend} is not a number or array of numbers. ≫

ReplaceAll::reps : {FindRoot[Nef[xxx, tend] == Nefpivot[VIt, Vend, kpivott], {xxx, tmin, tend}]} is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. ≫

FindRoot::srect : Value $t /. \frac{2.88\times 10^{36}\, VE'[t]^2}{ht[t]^2 \left(\frac{GHintpln[\ll 1 \gg]^2}{NMC[\ll 1 \gg]} + \frac{8.64\times 10^{36}\, NMCh[\ll 1 \gg]^2}{NMC[\ll 1 \gg]^2}\right) VE[t]^2} == 1.$ in search specification {xxx, tmin, tend} is not a number or array of numbers. ≫

ReplaceAll::reps : {FindRoot[Nef[xxx, tend] == Nefpivot[VIt, Vend, kpivott], {xxx, tmin, tend}]} is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. ≫

FindRoot::srect : Value $t /. \frac{2.88\times 10^{36}\, VE'[t]^2}{ht[t]^2 \left(\frac{GHintpln[\ll 1 \gg]^2}{NMC[\ll 1 \gg]} + \frac{8.64\times 10^{36}\, NMCh[\ll 1 \gg]^2}{NMC[\ll 1 \gg]^2}\right) VE[t]^2} == 1.$ in search specification {xxx, tmin, tend} is not a number or array of numbers. ≫

ReplaceAll::reps : {FindRoot[Nef[xxx, tend] == Nefpivot[VIt, Vend, kpivott], {xxx, tmin, tend}]} is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. ≫

FindRoot::srect : Value $t /. \frac{2.88\times 10^{36}\, VE'[t]^2}{ht[t]^2 \left(\frac{GHintpln[\ll 1 \gg]^2}{NMC[\ll 1 \gg]} + \frac{8.64\times 10^{36}\, NMCh[\ll 1 \gg]^2}{NMC[\ll 1 \gg]^2}\right) VE[t]^2} == 1.$ in search specification {xxx, tmin, tend} is not a number or array of numbers. ≫

General::stop : Further output of FindRoot::srect will be suppressed during this calculation. ≫

ReplaceAll::reps : {FindRoot[Nef[xxx, tend] == Nefpivot[VIt, Vend, kpivott], {xxx, tmin, tend}]} is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. ≫

General::stop : Further output of ReplaceAll::reps will be suppressed during this calculation. ≫

FindRoot::srect : Value $t /. \frac{2.88\times 10^{36}\, VE'[t]^2}{ht[t]^2 \left(\frac{GHintpln[\ll 1 \gg]^2}{NMC[\ll 1 \gg]} + \frac{8.64\times 10^{36}\, NMCh[\ll 1 \gg]^2}{NMC[\ll 1 \gg]^2}\right) VE[t]^2} == 1.$ in search specification {xxx, tmin, tend} is not a number or array of numbers. ≫

ReplaceAll::reps : {FindRoot[Nef[xxx, tend] == Nefpivot[VIt, Vend, kpivott], {xxx, tmin, tend}]} is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. ≫

FindRoot::srect : Value $t /. \frac{2.88\times 10^{36}\, VE'[t]^2}{ht[t]^2 \left(\frac{GHintpln[\ll 1 \gg]^2}{NMC[\ll 1 \gg]} + \frac{8.64\times 10^{36}\, NMCh[\ll 1 \gg]^2}{NMC[\ll 1 \gg]^2}\right) VE[t]^2} == 1.$ in search specification {xxx, tmin, tend} is not a number or array of numbers. ≫

ReplaceAll::reps : {FindRoot[Nef[xxx, tend] == Nefpivot[VIt, Vend, kpivott], {xxx, tmin, tend}]} is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. ≫

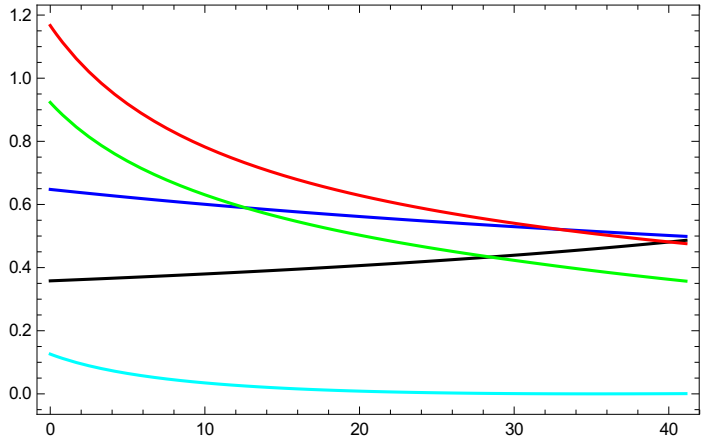
Primordial black holes

```
efffactor = 0.4;
Msunkg = 2. × 10^30; (*in units of kg*)
kgtoGeV = 5.61 * 10^26;
Msun = Msunkg * kgtoGeV;(*in units of GeV*)
zetactest = 0.0875;
tEQ = 60. × 10^3; (*in units of years*)
aEQ = 3.7 × 10^-4;
Ommh2 = 0.12;
MEQ = 7 × 10^50 * 10^-3 / Msunkg;
M = 2.4 * 10^(18); (*in units of GeV*)
MpRSUN = M / Msun;(*in units of Msun*)
Mstar = 3 * 10^12 * 10^-3 * kgtoGeV;(*in units of GeV*)
gstarFORM = 106.75;
hubble = 0.68;

betaFORM[t_, zetac_] := Erfc[zetac / Sqrt[2 Ps[t]]]

PBHmass[t_] := 4 π * efffactor * MpR^2 * Sqrt[3 MpR^2 / VE[t]] Exp[2 Nef[t, tend]]
PBHmassSUN[t_] := 4 π * efffactor * (MpR / Msun) * Sqrt[3 MpR^4 / VE[t]] Exp[2 Nef[t, tend]]
betaEQ[t_, zetac_] := betaFORM[t, zetac] aEQ / 1. Exp[Nef[tpivot05, tend] - Nef[t, tend]]
fraction[x_, zetac_] := 4.1 * 10^8 efffactor^(1/2) betaFORM[x, zetac] *
  (gstarFORM / 106.75)^(-1/4) (PBHmassSUN[x])^(-1/2) * (hubble / 0.68)^-2

findps[x_, zetac_] :=
 If[NumberQ[x] == True, FindRoot[1 == Erfc[zetac / Sqrt[2 PSvalue]] * Sqrt[MEQ / PBHmassSUN[x]], {PSvalue,
     1. × 10^-4, 1.}, AccuracyGoal → Automatic, PrecisionGoal → Automatic]][[1, 2]]
```

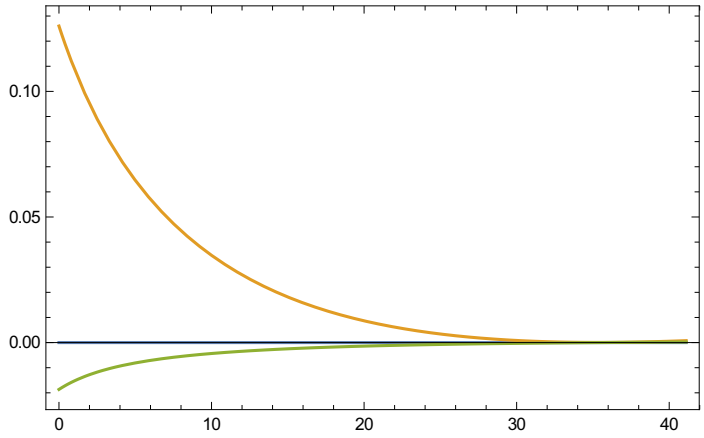
Result

Running of couplings

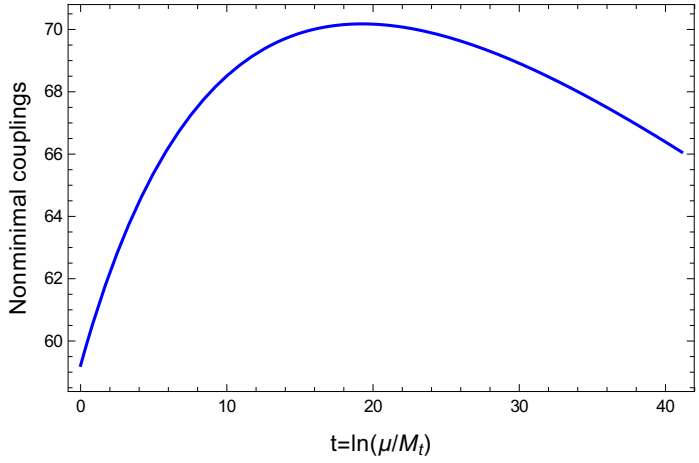
```
Plot[{g1sol[t], g2sol[t], g3sol[t], ytsol[t], lHsol[t]}, {t, 0, tmax},
 Frame → True, PlotRange → All, PlotStyle → {Black, Blue, Red, Green, Cyan}]
```



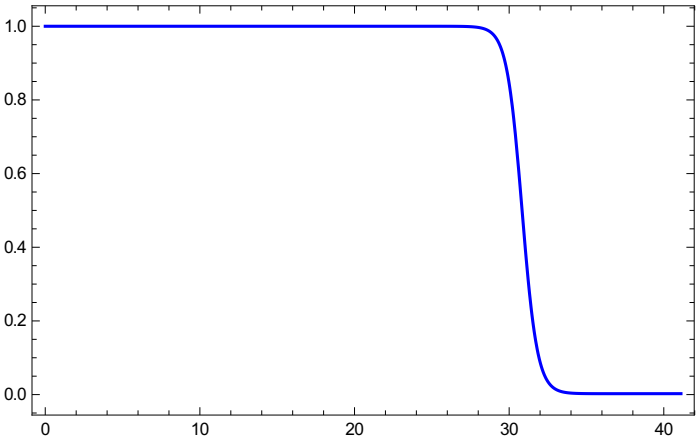

```
Plot[{0, lHsol[t], lHdsol[t]}, {t, 0, tmax}, Frame → True, PlotRange → All]
```



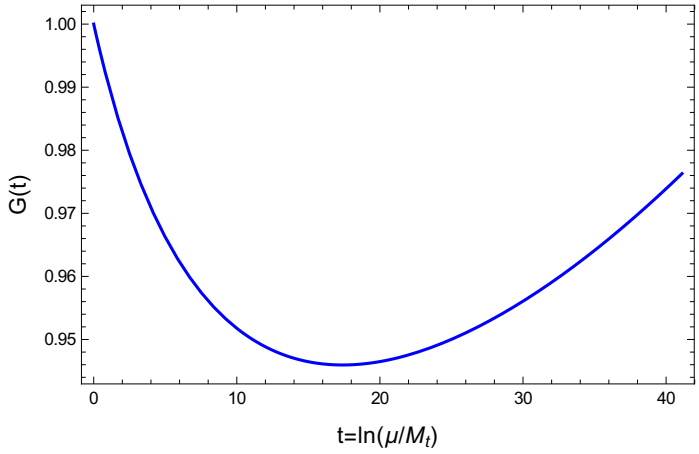
```
Plot[{xihsol[t]}, {t, 0, tmax}, Frame → True, PlotRange → All, PlotStyle → {Red, Blue},
 FrameLabel → {Style["t=ln(μ/Mt)", Medium], Style["Nonminimal couplings", Medium]}]
```




```
Plot[{sh[t] /. RGEsol}, {t, 0, tmax},
 Frame → True, PlotRange → All, PlotStyle → {Blue}]
```

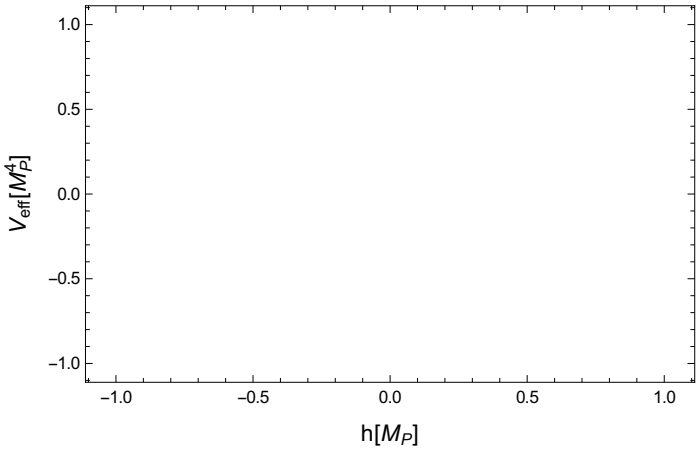



```
Plot[{GHintpln[t]}, {t, 0, tmax},
 Frame → True, PlotRange → All, PlotStyle → {Red, Blue},
 FrameLabel → {Style["t=ln(μ/Mt)", Medium], Style["G(t)", Medium]}]
```



Potential

```
ParametricPlot[{hsol[t] / MpR, VJordan[t] / MpR^4}, {t, 0, Log[Mp / mustart]},
 PlotRange → {All, All}, Frame → True, AspectRatio → 1 / GoldenRatio,
 FrameLabel → {Style["h[M_P]", Medium], Style["V_eff[M_P^4]", Medium]},
 PlotStyle → {Thick, Green}, Axes → False]
```



```
ParametricPlot[{hsol[t] / MpR, VE[t] / MpR^4}, {t, 0, Log[1 * Mp / mustart]},
 PlotRange → {All, All}, Frame → True, AspectRatio → 1 / GoldenRatio,
 GridLines → {{{hsol[tcrit] / MpR, {Dotted, Thick, Green}}},
    {hpivot05 / MpR, Dashed}, {hend / MpR, Dashed}}, {}}]
```

ReplaceAll::reps :

$\left\{\frac{2.88\times10^{36}\,VE'[t]^2}{ht[t]^2\left(\frac{GHintpln[\ll1\gg]^2}{NMC[\ll1\gg]}+\frac{8.64\times10^{36}\,NMCh[\ll1\gg]^2}{NMC[\ll1\gg]^2}\right)VE[t]^2}==1.\right\}$ is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. ≫

FindRoot::srect : Value t /. $\frac{2.88\times10^{36}\,VE'[t]^2}{ht[t]^2\left(\frac{GHintpln[\ll1\gg]^2}{NMC[\ll1\gg]}+\frac{8.64\times10^{36}\,NMCh[\ll1\gg]^2}{NMC[\ll1\gg]^2}\right)VE[t]^2}==1.$ in search specification {xxx, tmin, tend} is not a number or array of numbers. ≫

ReplaceAll::reps :

{FindRoot[Nef[xxx, tend] == Nefpivot[VIt, Vend, kpivott], {xxx, tmin, tend}]} is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. ≫

FindRoot::srect : Value t /. $\frac{2.88\times10^{36}\,VE'[t]^2}{ht[t]^2\left(\frac{GHintpln[\ll1\gg]^2}{NMC[\ll1\gg]}+\frac{8.64\times10^{36}\,NMCh[\ll1\gg]^2}{NMC[\ll1\gg]^2}\right)VE[t]^2}==1.$ in search specification {xxx, tmin, tend} is not a number or array of numbers. ≫

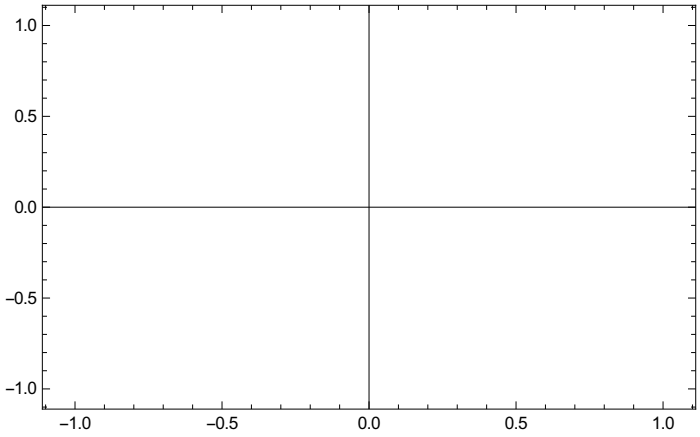
ReplaceAll::reps :

{FindRoot[Nef[xxx, tend] == Nefpivot[VIt, Vend, kpivott], {xxx, tmin, tend}]} is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. ≫

General::stop : Further output of ReplaceAll::reps will be suppressed during this calculation. ≫

FindRoot::srect : Value t /. $\frac{2.88\times10^{36}\,VE'[t]^2}{ht[t]^2\left(\frac{GHintpln[\ll1\gg]^2}{NMC[\ll1\gg]}+\frac{8.64\times10^{36}\,NMCh[\ll1\gg]^2}{NMC[\ll1\gg]^2}\right)VE[t]^2}==1.$ in search specification {xxx, tmin, tend} is not a number or array of numbers. ≫

General::stop : Further output of FindRoot::srect will be suppressed during this calculation. ≫



Observables

```
hpivot05 / MpR
```

```
7.24965
```

```
hend / MpR
```

```
0.115063
```

```
hsol[tcrit] / MpR
```

```
0.557521
```

```
Ps[t] /. t → tpivot05
ns[t] /. t → tpivot05
r[t] /. t → tpivot05
dnsdlnk[t] /. t → tpivot05
```

$$2.17312 \times 10^{-9}$$

```
0.92437
```

```
0.353871
```

```
-0.00397984
```

```
Nef[tpivot05, tend]
```

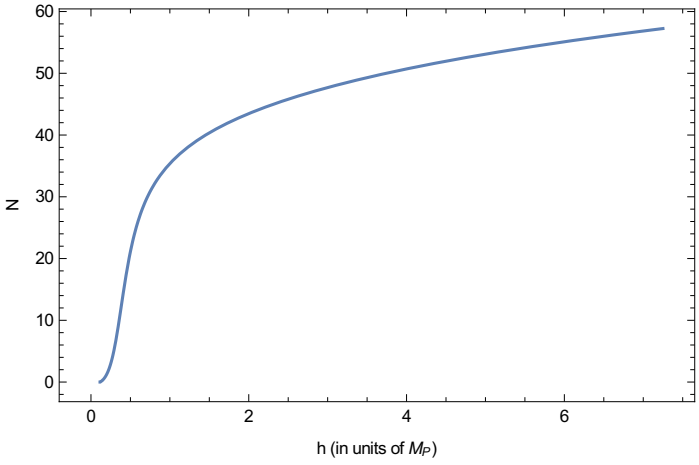
```
57.2444
```

```
Nefpivot[VI05, Vend, kpivot05]
```

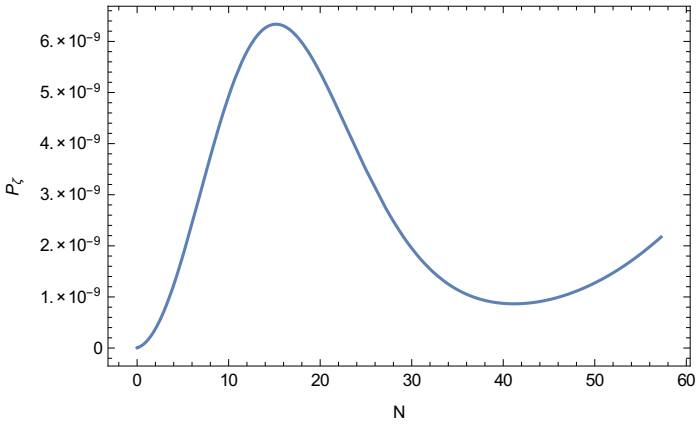
```
57.426
```

Power spectrum & PBH

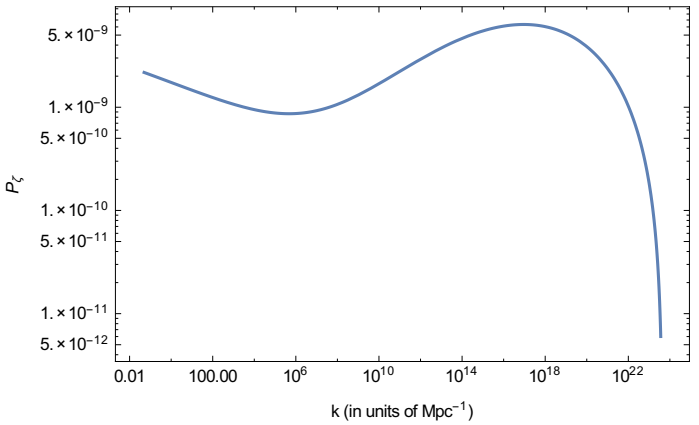
```
ParametricPlot[{hsol[t] / MpR, Nef[t, tend]},
 {t, tpivot05, tend}, AspectRatio → 1 / GoldenRatio,
 Frame → True, FrameLabel → {"h (in units of M_P)", "N"}]
```



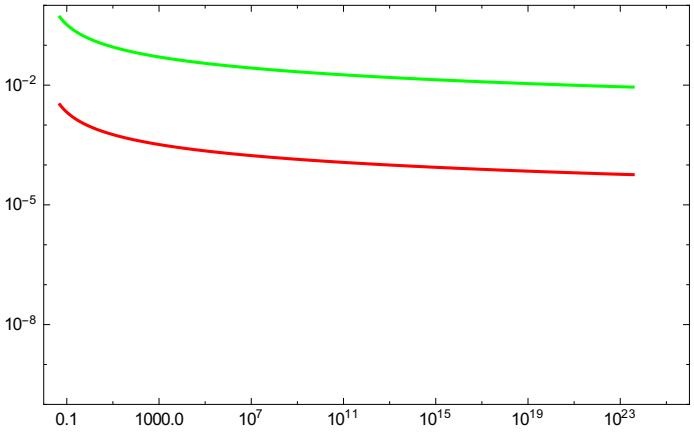
```
PSplotN = ParametricPlot[{Nef[t, tend], Ps[t]},
  {t, tpivot05, tend}, AspectRatio → 1 / GoldenRatio,
  Frame → True, PlotRange → All, FrameLabel → {"N", "P_ζ"}]
```




```
PSplotk = ParametricPlot[{kpivot05 * Exp[Nef[tpivot05, tend] - Nef[t, tend]], Ps[t]},
  {t, tpivot05, tend}, AspectRatio → 1 / GoldenRatio, Frame → True, PlotRange → All,
  FrameLabel → {"k (in units of Mpc^-1)", "P_ζ"}, ScalingFunctions → {"Log", "Log"},
  PlotRange → {{Log[10^-2], Log[10^26]}, {Log[10^-10], Log[10^-0]}}]
```



```
ParametricPlot[{kpivot05 * Exp[Nef[tpivot05, tend] - Nef[t, tend]], findps[t, 0.08]},
  {t, tpivot05, tend}, Frame → True, AspectRatio → 1 / GoldenRatio,
  ScalingFunctions → {"Log", "Log"}, PlotStyle → Red,
  PlotRange → {{Log[10^-2], Log[10^26]}, {Log[10^-10], Log[10^-0]}}];
ParametricPlot[{kpivot05 * Exp[Nef[tpivot05, tend] - Nef[t, tend]], findps[t, 1.0]},
  {t, tpivot05, tend}, Frame → True, AspectRatio → 1 / GoldenRatio,
  ScalingFunctions → {"Log", "Log"}, PlotStyle → Green,
  PlotRange → {{Log[10^-2], Log[10^26]}, {Log[10^-10], Log[10^-0]}}];
PSlimit = Show[%, %%]
```




```
Show[PSplotk, PSlimit]
```