HW #5

1. Boltzmann equation

```
<< Graphics`Graphics`
```

We start with the Boltzmann equation
$\frac{dn}{dt} + 3\,H\,n = -\langle\sigma\,v\rangle\,(n^2 - n_{\text{eq}}{}^2)$.
We defined the yield $Y = n/s$ and rewrite the equation in terms of $x = m/T$ instead of time. Because the entropy density scales as R^{-3}, it satisfies the equation
$\frac{ds}{dt} + 3\,H\,s = 0$
and hence the $3\,H$ term cancels for the yield,
$\frac{dY}{dt} = -\langle\sigma\,v\rangle\,s(Y^2 - Y_{\text{eq}}{}^2)$.
To rewrite the time variable by x, we need the relationship between time and temperature in radiation dominated universe. The expansion rate is
$H^2 = \frac{8\pi}{3}\,G_N\,\rho_R = \frac{8\pi}{3}\,G_N\,g_*\,\frac{\pi^2}{30}\,T^4$.
Using $H = \frac{\dot{R}}{R} = \frac{dR}{R\,dt}$, and noting $T \propto R^{-1}$ and hence $\frac{dR}{R} = -\frac{dT}{T}$, we find $d\,t = -\frac{1}{H}\,\frac{dT}{T}$. Because $H(T) = H(m)\,\frac{T^2}{m^2}$,
$d\,t = -\frac{m^2}{H(m)}\,\frac{dT}{T^3}$. Then the Boltzmann equation becomes
$\frac{dY}{dT} = \frac{m^2}{H(m)}\,\frac{1}{T^3}\,\langle\sigma\,v\rangle\,s(Y^2 - Y_{\text{eq}}{}^2)$
Using $s(T) = s(m)\,\frac{T^3}{m^3}$, it becomes
$\frac{dY}{dT} = \frac{m^2}{H(m)}\,\frac{s(m)}{m^3}\,\langle\sigma\,v\rangle\,(Y^2 - Y_{\text{eq}}{}^2) = \frac{s(m)}{H(m)}\,\frac{1}{m}\,\langle\sigma\,v\rangle\,(Y^2 - Y_{\text{eq}}{}^2)$.
Finally, $d\,T = d\,\frac{m}{x} = -m\,\frac{dx}{x^2}$, and
$\frac{dY}{dx} = -\frac{1}{x^2}\,\frac{s(m)}{H(m)}\,\langle\sigma\,v\rangle\,(Y^2 - Y_{\text{eq}}{}^2)$
$= -0.368\,g_*{}^{1/2}\,\frac{1}{x^2}\,m\,M_{\text{Pl}}\,\langle\sigma\,v\rangle\,(Y^2 - Y_{\text{eq}}{}^2)$
Here, $M_{\text{Pl}} = (8\,\pi\,G_N)^{-1/2} = 2.44\times 10^{18}$ GeV.

```
N[(1/3 π^2/30)^(-1/2) Zeta[3]/π^2]
```

```
0.367787
```

To integrate the Boltzmann equation, we need to have an expression for the equilibrium yield. Once the particle is non-relativistic, the difference in statistics is not important. The number density is
$n_{\text{eq}} = \int\frac{d^3 p}{(2\pi)^3}\,e^{-\beta E} = \int\frac{d^3 p}{(2\pi)^3}\,e^{-\beta(m+p^2/2m)} = e^{-\beta m}\left(\frac{mT}{2\pi}\right)^{3/2}$.
The yield is
$Y_{\text{eq}} = \frac{n_{\text{eq}}}{s} = \frac{1}{g_*}\,\frac{\pi^2}{\zeta(3)}\,\frac{1}{T^3}\,e^{-\beta m}\left(\frac{mT}{2\pi}\right)^{3/2} = \frac{\pi^2}{g_*\,\zeta(3)}\,e^{-x}\left(\frac{x}{2\pi}\right)^{3/2} = 0.521\,g_*{}^{-1}\,x^{3/2}\,e^{-x}$.

```
N[π^2/(Zeta[3] (2 π)^(3/2))]
```

```
0.521321
```

The boundary condition is such that $Y = Y_{eq}$ at $x = 0$. Unfortunately for $x = 0$ (or $T \gg m$), the non-relativistic approximation we made to work out Y_{eq} above is no longer good. Because the result is not too sensitive to the initial condition, we set $Y = Y_{eq}$ at $x = 1$. We will verify later that indeed the result is insensitive to this choice. We use GeV unit for everything.

S-wave

Mathematica unfortunately seems to have trouble dealing with big numbers such as M_{Pl}. We can help it by solving for
$y = \frac{s(m)}{H(m)} \langle \sigma v \rangle Y$
$\frac{dy}{dx} = -\frac{1}{x^2} (y^2 - y_{eq}{}^2)$,
$y_{eq} = \frac{s(m)}{H(m)} \langle \sigma v \rangle \frac{1}{s} e^{-\beta m} (\frac{m T}{2\pi})^{3/2} = \left(\frac{1}{3} \frac{\pi^2}{30}\right)^{-1/2} M_{Pl} \frac{m}{T^3} \langle \sigma v \rangle e^{-\beta m} (\frac{m T}{2\pi})^{3/2}$
$= 0.192 M_{Pl}\, m \langle \sigma v \rangle x^{3/2} e^{-x}$

$$N\left[\left(\frac{1}{3}\ \frac{\pi^2}{30}\right)^{-1/2} \left(\frac{1}{2\pi}\right)^{3/2}\right]$$

```
0.191735
```

Mathematica still balks when I let it integrate all the way from $x = 1$ to 10000. I break it up into $1 < x < 50$ and the rest.

```
solution1 =
 NDSolve[{y'[x] == -1/x^2 (y[x]^2 - (0.192 M_Pl m σ x^(3/2) E^-x)^2), y[1] == 0.192 M_Pl m σ 1^(3/2) E^-1} /.
   {m → 1000, g → 100, σ → 10^-10, M_Pl → 2.44 10^18}, y, {x, 1, 50}]
```

```
{{y → InterpolatingFunction[{{1., 50.}}, <>]}}
```

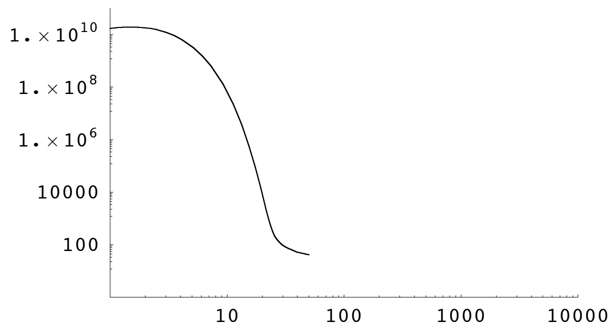
```
Evaluate[y[50] /. solution1]
```

```
{42.1441}
```

```
solution2 =
 NDSolve[{y'[x] == -1/x^2 (y[x]^2 - (0.192 M_Pl m σ x^(3/2) E^-x)^2), y[50] == 42.144129023721824`} /.
   {m → 1000, g → 100, σ → 10^-10, M_Pl → 2.44 10^18}, y, {x, 50, 10000}]
```

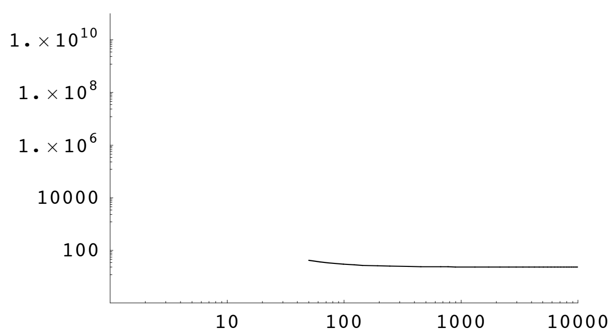
```
{{y → InterpolatingFunction[{{50., 10000.}}, <>]}}
```

```
LogLogPlot[Evaluate[y[x] /. solution1], {x, 1, 50}, PlotRange → {{1, 10000}, {1, 10^11}}]
```



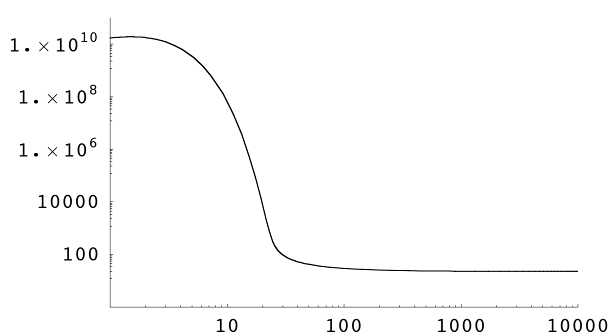

```
- Graphics -
```

```
LogLogPlot[Evaluate[y[x] /. solution2],
 {x, 50, 10000}, PlotRange → {{1, 10000}, {1, 10^11}}]
```




```
- Graphics -
```

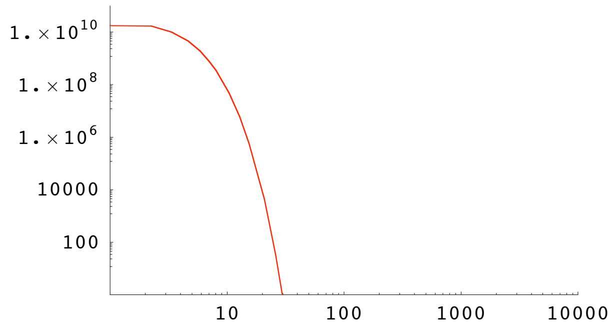
```
Show[%, %%]
```




```
- Graphics -
```

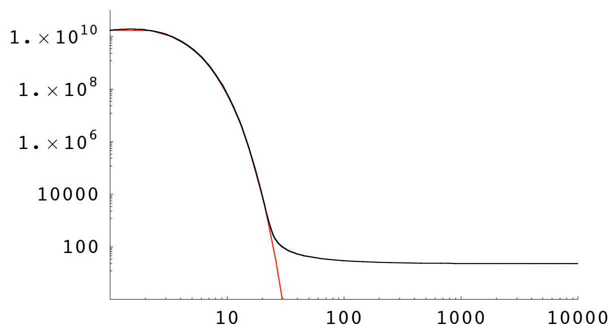
Compared to the equlibrium values,

```
LogLogPlot[0.192 M_Pl m σ x^(3/2) E^-x /. {m → 1000, g → 100, σ → 10^-10, M_Pl → 2.44 10^18},
  {x, 1, 1000}, PlotRange → {{1, 10000}, {1, 10^11}}, PlotStyle → RGBColor[1, 0, 0]]
```



- Graphics -

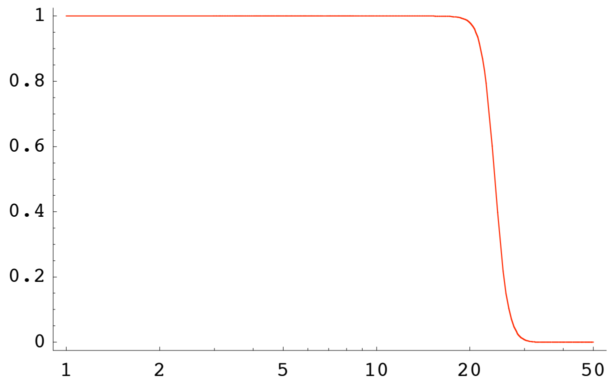
```
Show[%, %%]
```



- Graphics -

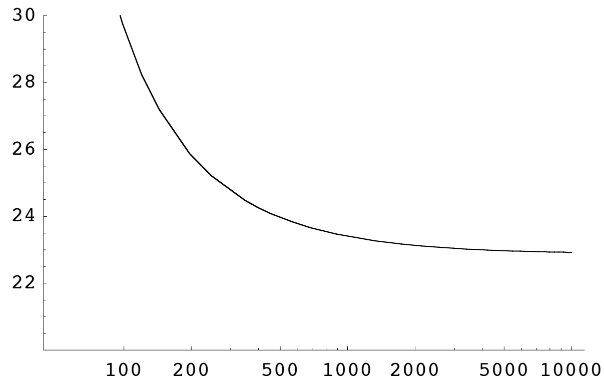
the solution is basically right on the equilibrium until $x \sim 20$, and then becomes approximately constant afterwards. This is why we don't expect the result to be sensitive to the initial condition as long as it starts on the equilibrium for $x < 10$ or so. Verify this point by taking the ratio

```
LogLinearPlot[0.192 M_Pl m σ x^(3/2) E^-x / Evaluate[y[x] /. solution1[[1]]] /.
  {m → 1000, g → 100, σ → 10^-10, M_Pl → 2.44 10^18}, {x, 1, 50}, PlotStyle → RGBColor[1, 0, 0]]
```



- Graphics -

```
LogLinearPlot[Evaluate[y[x] /. solution2], {x, 50, 10000}, PlotRange → {20, 30}]
```



- Graphics -

Therefore, using the notation in the problem, $Y(\infty) = \frac{H(m)}{s(m)\,\sigma_0}\, y(\infty)$, and $y(\infty)$ is about when the abundance starts to deviate significantly from the equilibrium value. Because this behavior is mainly due to the exponential dropoff of Y_{eq}, it is expected to be rather insensitive to the chioce of m and σ_0. Indeed, by varying m, we find

```
Table[{10^t, Clear[yint1, yint2, solution1, solution2, solution3]; solution1 =
   NDSolve[{y'[x] == -1/x^2 (y[x]^2 - (0.192 M_Pl m σ x^(3/2) E^-x)^2), y[1] == 0.192 M_Pl m σ 1^(3/2) E^-1} /.
     {m → 10^t, g → 100, σ → 10^-10, M_Pl → 2.44 10^18}, y, {x, 1, 20}];
  yint1 = Evaluate[y[20] /. solution1[[1]]]; solution2 =
   NDSolve[{y'[x] == -1/x^2 (y[x]^2 - (0.192 M_Pl m σ x^(3/2) E^-x)^2), y[20] == yint1} /.
     {m → 10^t, g → 100, σ → 10^-10, M_Pl → 2.44 10^18}, y, {x, 20, 1000}];
  yint2 = Evaluate[y[1000] /. solution2[[1]]]; solution3 =
   NDSolve[{y'[x] == -1/x^2 (y[x]^2 - (0.192 M_Pl m σ x^(3/2) E^-x)^2), y[1000] == yint2} /.
     {m → 10^t, g → 100, σ → 10^-10, M_Pl → 2.44 10^18}, y, {x, 1000, 10000}];
  Evaluate[y[10000] /. solution3[[1]]]}, {t, 0.5, 3.5, 0.1}]
```

```
Part::partd : Part specification solution3[[1]] is longer than depth of object. More…

ReplaceAll::reps : {solution3[[1]]} is neither a list of replacement
   rules nor a valid dispatch table, and so cannot be used for replacing. More…
```

```
{{3.16228, 17.2766}, {3.98107, 17.2766}, {5.01187, 17.5016},
 {6.30957, 17.7265}, {7.94328, 17.9516}, {10., 18.1768}, {12.5893, 18.402},
 {15.8489, 18.6273}, {19.9526, 18.8527}, {25.1189, 19.0782}, {31.6228, 19.3037},
 {39.8107, 19.5293}, {50.1187, 19.755}, {63.0957, 19.9807}, {79.4328, 20.2065},
 {100., 20.4324}, {125.893, 20.6583}, {158.489, 20.8843}, {199.526, 21.1104},
 {251.189, 21.3365}, {316.228, 21.5627}, {398.107, 21.7889}, {501.187, 22.0152},
 {630.957, 22.2416}, {794.328, 22.468}, {1000., 22.6945}, {1258.93, 22.921},
 {1584.89, 23.1476}, {1995.26, 23.3742}, {2511.89, 23.6009}, {3162.28, 23.8277}}
```

Only a mild variation for a very wide range of m.

(caution: the version of *Mathematica* I have seems to have a strange memory effect that always gives the first entry wrong. I get the reasonable results by quitting the kernel once and re-run the command.)

P-wave

Going back to
$\frac{dY}{dx} = -\frac{1}{x^2} \frac{s(m)}{H(m)} \langle \sigma v \rangle (Y^2 - {Y_{eq}}^2)$
$= -0.368\, {g_*}^{1/2} \frac{1}{x^2} m M_{Pl} \langle \sigma v \rangle (Y^2 - {Y_{eq}}^2)$
we now substitute $\langle \sigma v \rangle = \sigma_0\, x^{-1}$.
$\frac{dY}{dx} = -\frac{1}{x^3} \frac{s(m)}{H(m)} \sigma_0 (Y^2 - {Y_{eq}}^2)$
Mathematica again seems to have trouble dealing with big numbers such as M_{Pl}. We can help it by solving for
$y = \frac{s(m)}{H(m)} \sigma_0\, Y$
$\frac{dy}{dx} = -\frac{1}{x^3} (y^2 - {y_{eq}}^2)$,
$y_{eq} = \frac{s(m)}{H(m)} \langle \sigma v \rangle \frac{1}{s} e^{-\beta m} (\frac{mT}{2\pi})^{3/2} = \left(\frac{1}{3} \frac{\pi^2}{30}\right)^{-1/2} M_{Pl} \frac{m}{T^3} \langle \sigma v \rangle e^{-\beta m} (\frac{mT}{2\pi})^{3/2}$
$= 0.192\, M_{Pl}\, m \langle \sigma v \rangle x^{3/2} e^{-x}$

```
N[(1/3 π^2/30)^(-1/2) (1/(2 π))^(3/2)]
```

```
0.191735
```

Mathematica still balks when I let it integrate all the way from $x = 1$ to 10000. I break it up into $1 < x < 50$ and the rest.

solution1 =
$$\texttt{NDSolve}\left[\left\{y'[x] == -\frac{1}{x^3}\left(y[x]^2 - \left(0.192\, M_{Pl}\, m\, \sigma\, x^{3/2}\, E^{-x}\right)^2\right),\ y[1] == 0.192\, M_{Pl}\, m\, \sigma\, 1^{3/2}\, E^{-1}\right\} /.\ \{m \to 1000,\ g \to 100,\ \sigma \to 10^{-10},\ M_{Pl} \to 2.44\ 10^{18}\},\ y,\ \{x, 1, 50\}\right]$$

```
{{y → InterpolatingFunction[{{1., 50.}}, <>]}}
```

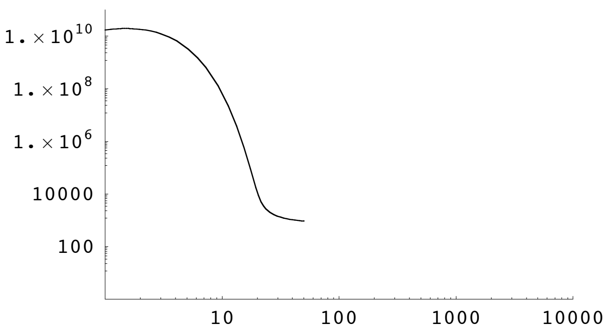
Evaluate[y[50] /. solution1]

```
{941.006}
```

solution2 =
$$\texttt{NDSolve}\left[\left\{y'[x] == -\frac{1}{x^3}\left(y[x]^2 - \left(0.192\, M_{Pl}\, m\, \sigma\, x^{3/2}\, E^{-x}\right)^2\right),\ y[50] == 941.0060810113391\text{`}\right\} /.\ \{m \to 1000,\ g \to 100,\ \sigma \to 10^{-10},\ M_{Pl} \to 2.44\ 10^{18}\},\ y,\ \{x, 50, 10000\}\right]$$

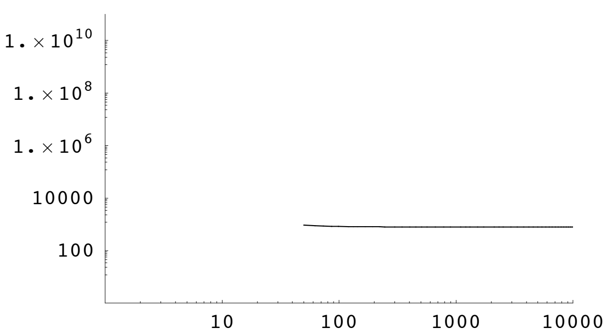
```
{{y → InterpolatingFunction[{{50., 10000.}}, <>]}}
```

LogLogPlot[Evaluate[y[x] /. solution1], {x, 1, 50}, PlotRange → {{1, 10000}, {1, 10^{11}}}]



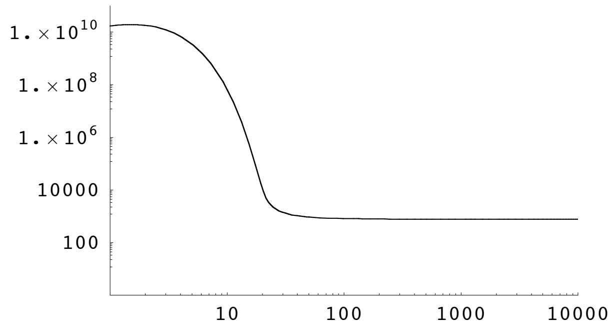
- Graphics -

LogLogPlot[Evaluate[y[x] /. solution2],
{x, 50, 10000}, PlotRange → {{1, 10000}, {1, 10^{11}}}]



- Graphics -

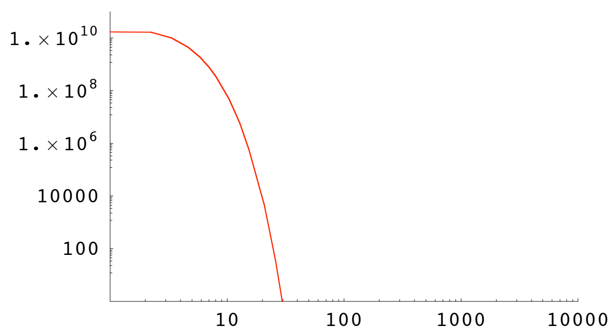
```
Show[%, %%]
```




```
- Graphics -
```

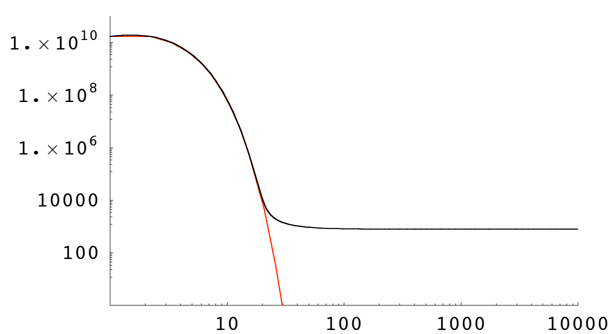
Compared to the equlibrium values,

```
LogLogPlot[0.192 M_Pl m σ x^(3/2) E^-x /. {m → 1000, g → 100, σ → 10^-10, M_Pl → 2.44 10^18},
  {x, 1, 1000}, PlotRange → {{1, 10000}, {1, 10^11}}, PlotStyle → RGBColor[1, 0, 0]]
```




```
- Graphics -
```

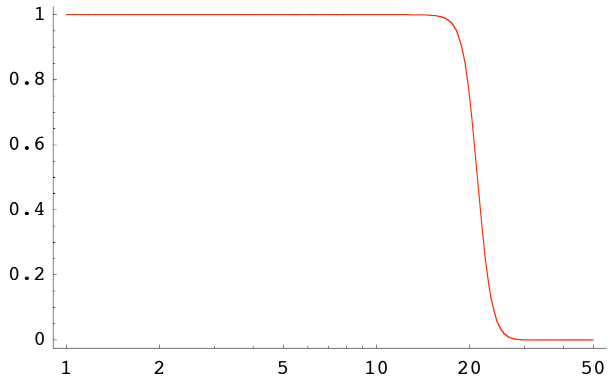
```
Show[%, %%]
```




```
- Graphics -
```

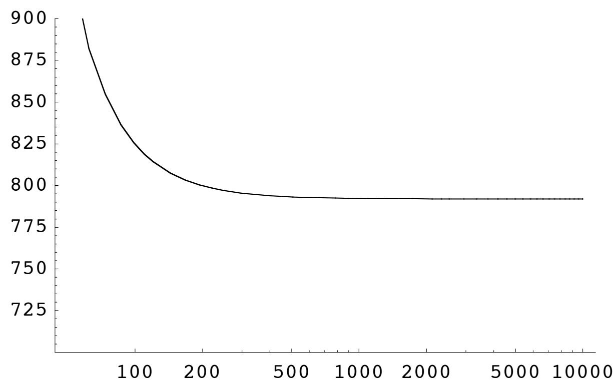
the solution is basically right on the equilibrium until $x \sim 20$, and then becomes approximately constant afterwards. This is why we don't expect the result to be sensitive to the initial condition as long as it starts on the equilibrium for $x < 10$ or so. Verify this point by taking the ratio

```
LogLinearPlot[0.192 M_Pl m σ x^(3/2) E^-x / Evaluate[y[x] /. solution1[[1]]] /.
  {m → 1000, g → 100, σ → 10^-10, M_Pl → 2.44 10^18}, {x, 1, 50}, PlotStyle → RGBColor[1, 0, 0]]
```



- Graphics -

```
LogLinearPlot[Evaluate[y[x] /. solution2], {x, 50, 10000}, PlotRange → {700, 900}]
```



- Graphics -

Therefore, using the notation in the problem, $Y(\infty) = \frac{H(m)}{s(m)\,\sigma_0}\,y(\infty)$, and $y(\infty)$ is about $2\,x_f^{\,2}$ where x_f is when the abundance starts to deviate significantly from the equilibrium value. Because this behavior is mainly due to the exponential dropoff of Y_{eq}, it is expected to be rather insensitive to the chioce of m and σ_0. Indeed, by varying m, we find

```
Table[{10^t, Clear[yint1, yint2, solution1, solution2, solution3]; solution1 =
   NDSolve[{y'[x] == -1/x^3 (y[x]^2 - (0.192 M_Pl m σ x^(3/2) E^-x)^2), y[1] == 0.192 M_Pl m σ 1^(3/2) E^-1} /.
     {m → 10^t, g → 100, σ → 10^-10, M_Pl → 2.44 10^18}, y, {x, 1, 20}];
  yint1 = Evaluate[y[20] /. solution1[[1]]]; solution2 =
   NDSolve[{y'[x] == -1/x^3 (y[x]^2 - (0.192 M_Pl m σ x^(3/2) E^-x)^2), y[20] == yint1} /.
     {m → 10^t, g → 100, σ → 10^-10, M_Pl → 2.44 10^18}, y, {x, 20, 1000}];
  yint2 = Evaluate[y[1000] /. solution2[[1]]]; solution3 =
   NDSolve[{y'[x] == -1/x^3 (y[x]^2 - (0.192 M_Pl m σ x^(3/2) E^-x)^2), y[1000] == yint2} /.
     {m → 10^t, g → 100, σ → 10^-10, M_Pl → 2.44 10^18}, y, {x, 1000, 10000}];
  Evaluate[y[10000] /. solution3[[1]]]}, {t, 0.5, 3.5, 0.1}]
```

```
NDSolve::ndcf : Repeated convergence test failure at x == 2.6155257913357994`; unable to continue. More…

InterpolatingFunction::dmval : Input value {20} lies outside the
   range of data in the interpolating function. Extrapolation will be used. More…

{{3.16228, 24.0545}, {3.98107, 425.514}, {5.01187, 437.854},
 {6.30957, 450.385}, {7.94328, 463.108}, {10., 476.022}, {12.5893, 489.129},
 {15.8489, 502.428}, {19.9526, 515.919}, {25.1189, 529.604}, {31.6228, 543.482},
 {39.8107, 557.553}, {50.1187, 571.817}, {63.0957, 586.275}, {79.4328, 600.928},
 {100., 615.774}, {125.893, 630.815}, {158.489, 646.051}, {199.526, 661.481},
 {251.189, 677.107}, {316.228, 692.928}, {398.107, 708.944}, {501.187, 725.155},
 {630.957, 741.563}, {794.328, 758.166}, {1000., 774.966}, {1258.93, 791.962},
 {1584.89, 809.154}, {1995.26, 826.543}, {2511.89, 844.128}, {3162.28, 863.296}}
```

Only a mild variation for a very wide range of *m*.

(caution: the version of *Mathematica* I have seems to have a strange memory effect that always gives the first entry wrong. I get the reasonable results by quitting the kernel once and re-run the command.)

2. Ω_{DM}

S-wave

Once we have the yield, it is easy to convert it to the current energy density.
$\rho_{\text{DM}} = m\, Y(\infty)\, s_0 = m\, \frac{H(m)}{s(m)\,\sigma_0}\, y(\infty)\, s_0$
while $\rho_c = \frac{3 H_0^2}{8\pi G_N} = 3 H_0^2 M_{\text{Pl}}^2$, $x_f = y(\infty)$, and
$H(m)^2 = \frac{1}{3 M_{\text{Pl}}^2}\, g_*\, \frac{\pi^2}{30}\, m^4$, $s(m) = g_*\, \frac{2\pi^2}{45}\, m^3$,
and hence
$\Omega_{\text{DM}} = m \sqrt{\frac{g_*}{3 M_{\text{Pl}}^2}\, \frac{\pi^2}{30}\, m^4}\, \frac{45}{2\pi^2\, g_*\, m^3}\, \frac{x_f}{\sigma_0}\, s_0\, \frac{1}{3 H_0^2 M_{\text{Pl}}^2}$
$= 0.252\, \frac{x_f\, s_0}{g_*^{1/2} M_{\text{Pl}}^3\, H_0^2\, \sigma_0}$

```
In[50]:= N[1/3 (1/3 π^2/30)^(1/2) (2 π^2/45)^-1]

Out[50]= 0.251646
```

We found $x_f \approx 23$ for the choice of parameters.

From HW #3, we found
$s = s_\gamma + s_\nu = (1 + \frac{21}{22}) s_\gamma = \frac{43}{22}\, 2\, \frac{2\pi^2}{45}\, T_0{}^3 = 2890\,\text{cm}^{-3}$

```
(1 + 21/22) 2 (2 π^2/45) hbarc^-3 T0^3 /. {hbarc → 0.1973 10^-4, T0 → 2.725 * 8.617 10^-5}
```

```
2890.54
```

It is useful to re-express the Hubble constant in the GeV unit (sounds crazy):
$H_0 = 100\, h\, \text{km sec}^{-1}\ \text{Mpc}^{-1} = \frac{100\, h\, \text{km sec}^{-1}}{3.00\times 10^5\ \text{km sec}^{-1}} \frac{0.1973\ \text{GeV fm}}{10^6\ 3.086\ 10^{16}\ m} = 2.131\ 10^{-42}\ \text{GeV}\, h$

```
(100/(3 10^5)) (0.1973 10^-15/(10^6 3.086 10^16))
```

```
2.13113 × 10^-42
```

Therefore $\Omega_{\text{DM}}\, h^2$ is given by

```
In[77]:= 0.251646 (xf s0 hbarc^3)/(g^(1/2) MPl^3 H0^2 σ0) /.
           {xf → 23, s0 → 2890, hbarc → 0.1973 10^-13, MPl → 2.44 10^18, H0 → 2.13 10^-42, g → 100, σ0 → 10^-10}

Out[77]= 1.94925
```

This is too big.

To get the realistic value $\Omega_{\text{DM}}\, h^2 = 0.12$, we want $\sigma_0 = 1.6 \times 10^{-9}\ \text{GeV}^{-2}$. This is the range of cross sections of our interest.

```
In[52]:= 1.9492533091097788` / 0.12

Out[52]= 16.2438
```

P-wave

This time, $y(\infty)$ is enhanced to 775. Therefore $\Omega_{\text{DM}}\, h^2$ is given by

```
In[83]:= 0.251646 (y s0 hbarc^3)/(g^(1/2) MPl^3 H0^2 σ0) /.
           {y → 775, s0 → 2890, hbarc → 0.1973 10^-13, MPl → 2.44 10^18, H0 → 2.13 10^-42, g → 100, σ0 → 10^-10}

Out[83]= 65.6814
```

This is too big.

To get the realistic value $\Omega_{\text{DM}}\, h^2 = 0.12$, we want $\sigma_0 = 5.5 \times 10^{-8}\ \text{GeV}^{-2}$ would do much better. This is the range of cross sections of our interest.

```
In[84]:= 65.6813615026121` / 0.12

Out[84]= 547.345
```

In either case, the cross section is of the order of electroweak scale, $\sigma_0 \sim \frac{\pi \alpha^2}{m^2}$ with $m \sim$ TeV.

Optional

To be followed.