The Exercises at the HAP Workshop 2014 - Wednesday

Gernot Maier,
Moritz Hütten (DESY Zeuthen)

HELMHOLTZ ASSOCIATION
Alliance for Astroparticle Physics

HUMBOLDT-UNIVERSITÄT ZU BERLIN

Overview: What we want (you) to do

1) **Tuesday (Mar 11)**
   - **Calculation of annihilation rates**
   - **Inspection of supersymmetric models and annihilation spectra**
   - **Responsible: Ulli Schwanke, Louise Oakes**
2) **Today (Mar 12)**
   - **Calculation of astrophysical factors (analytically and using existing tools)**
   - **Calculation of fluxes**
   - **Responsible: Gernot Maier, Moritz Hütten**
3) **Thursday (Mar 13)**
   - **Calculation of event statistics, background rates**
   - **Statistical tests**
   - **Responsible: Rolf Bühler, Markus Ackermann**

Note: The exercises of each day will use the same toy model and the same (not yet existing) detector

Today's Exercises

- all exercises are available on the Indico page
- all necessary files are already stored in the HAPworkshop_Day3_clumpy/ folder
- There is still a wrong Dark Matter density value for the toy model inside the toytarget.txt file and the HAPworkshop.toytarget.py script:

  Either find it and correct it or download a newer version of the scripts!

Task 0 - Answers

Units conversion:

In[1]:= $pc = 10^{-3}\ kpc;$

$meter = \frac{pc}{3.0857 \times 10^{16}};$

$cm = \frac{meter}{100}$

Out[3]= $3.24076 \times 10^{-22}\ kpc$

In[8]:= $kg = \frac{msun}{1.9891 \times 10^{30}};$

$eV = 1.782662 \times 10^{-36}\ kg;$

$TeV = 10^{12}\ eV$

Out[10]= $8.96215 \times 10^{-55}\ msun$

In[11]:= $\frac{TeV^2}{cm^5}$

Out[11]= $\frac{0.224695\ msun^2}{kpc^5}$

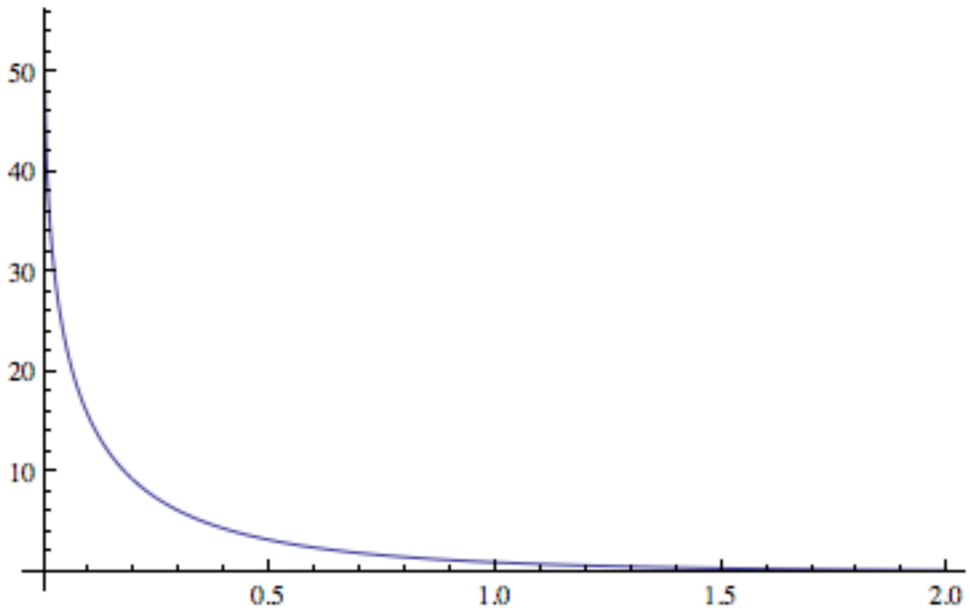
In[12]:= $0.15\ \frac{TeV}{cm^3}$

Out[12]= $\frac{3.94971 \times 10^9\ msun}{kpc^3}$

= rho_0

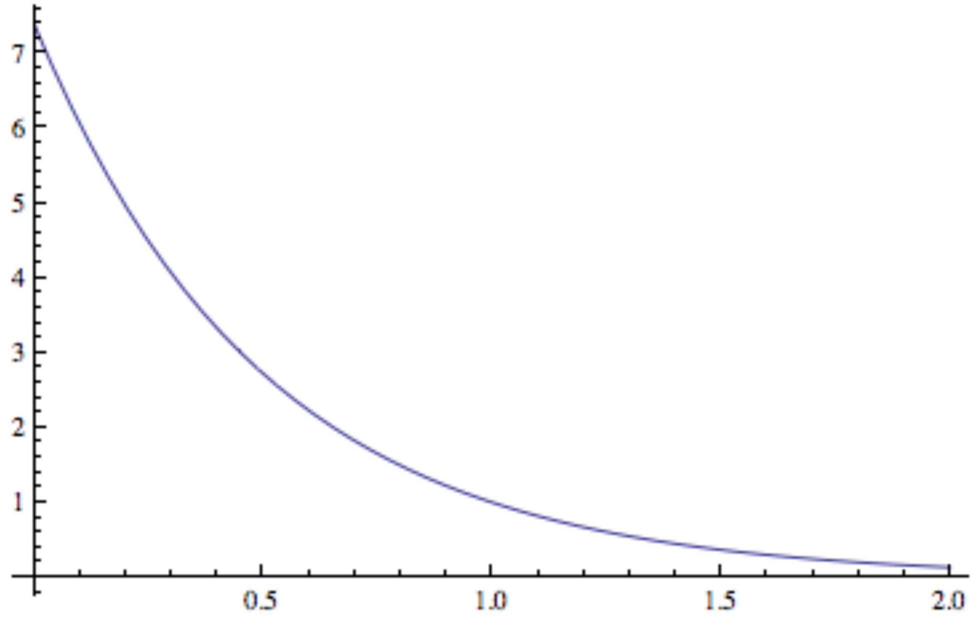
Task 1a - Answers

```
rhoE[r_] := rhos * Exp[- 2/alpha ((r/rs)^alpha - 1)]

rhos = 1;
rs = 1;
alpha = 0.5;

Plot[rhoE[x], {x, 0, 2 * rs}, PlotRange → All]
```



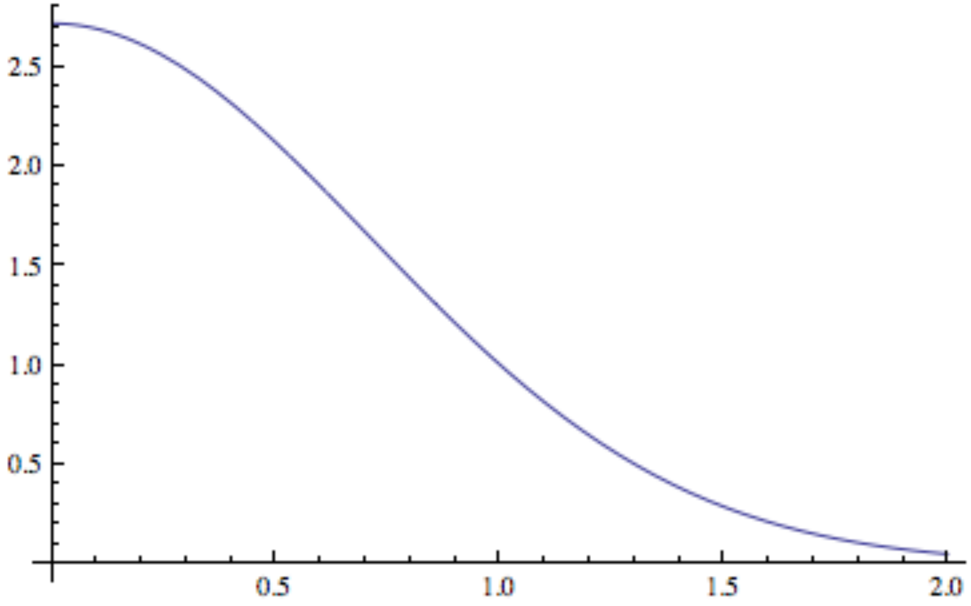
Task 1a - Answers

```
rhoE[r_] := rhos * Exp[- 2/alpha ((r/rs)^alpha - 1)]

rhos = 1;
rs = 1;
alpha = 1;

Plot[rhoE[x], {x, 0, 2*rs}, PlotRange → All]
```



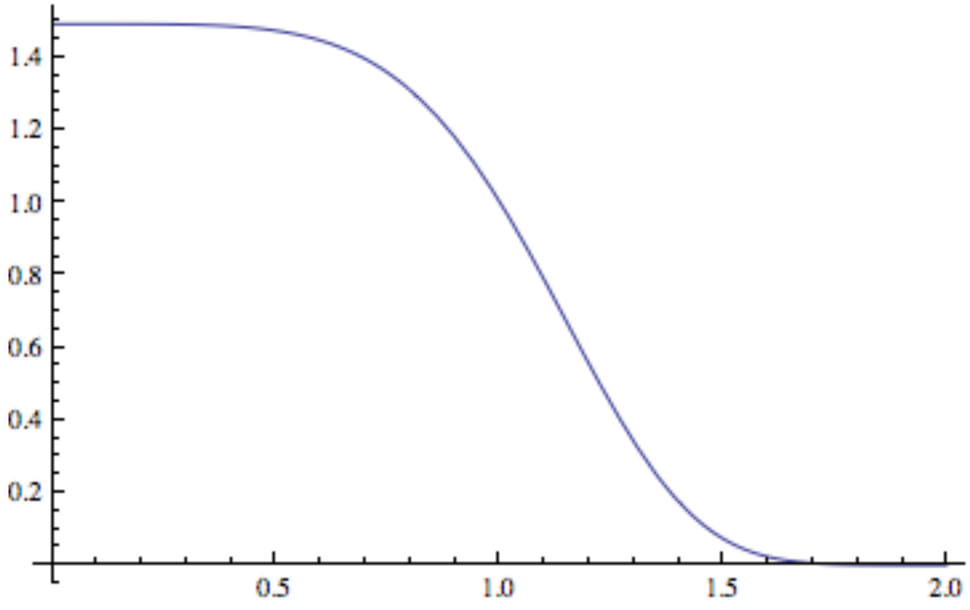
Task 1a - Answers

```
rhoE[r_] := rhos * Exp[- 2/alpha ((r/rs)^alpha - 1)]

rhos = 1;
rs = 1;
alpha = 2;

Plot[rhoE[x], {x, 0, 2*rs}, PlotRange → All]
```



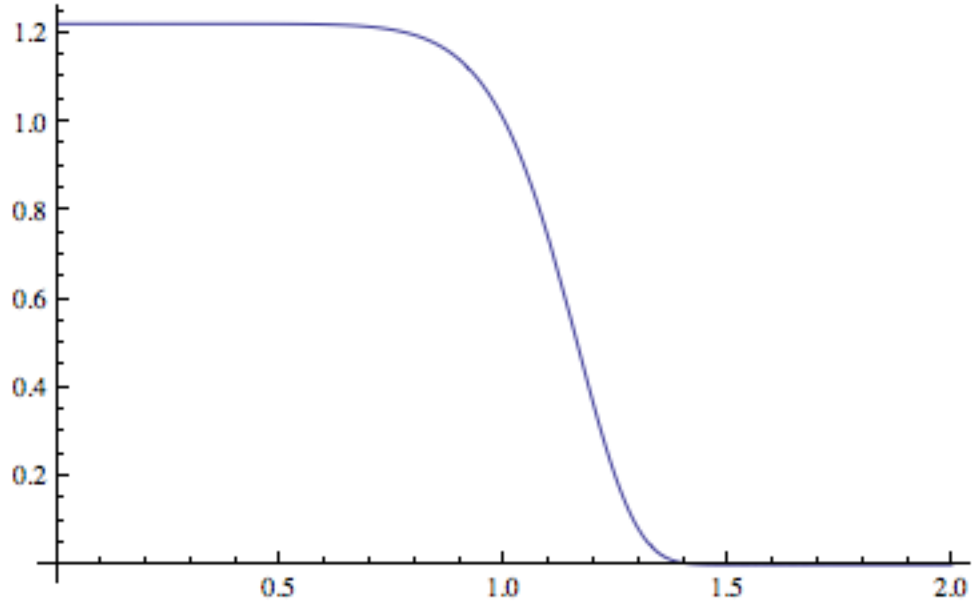
Task 1a - Answers

```
rhoE[r_] := rhos * Exp[-2/alpha ((r/rs)^alpha - 1)]

rhos = 1;
rs = 1;
alpha = 5;

Plot[rhoE[x], {x, 0, 2*rs}, PlotRange → All]
```



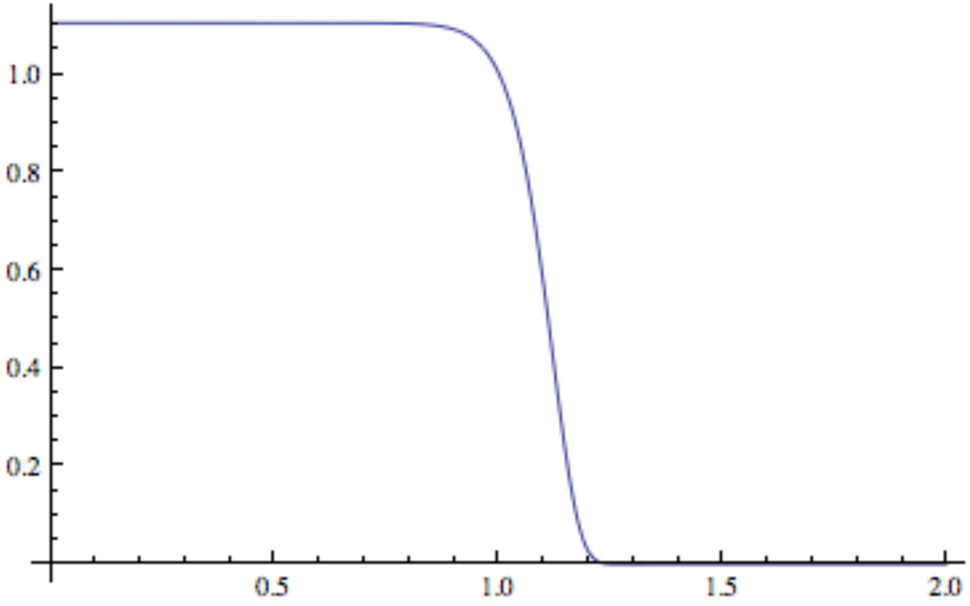
Task 1a - Answers

```
rhoE[r_] := rhos * Exp[-2/alpha ((r/rs)^alpha - 1)]

rhos = 1;
rs = 1;
alpha = 10;

Plot[rhoE[x], {x, 0, 2*rs}, PlotRange → All]
```



Task 1a - Answers

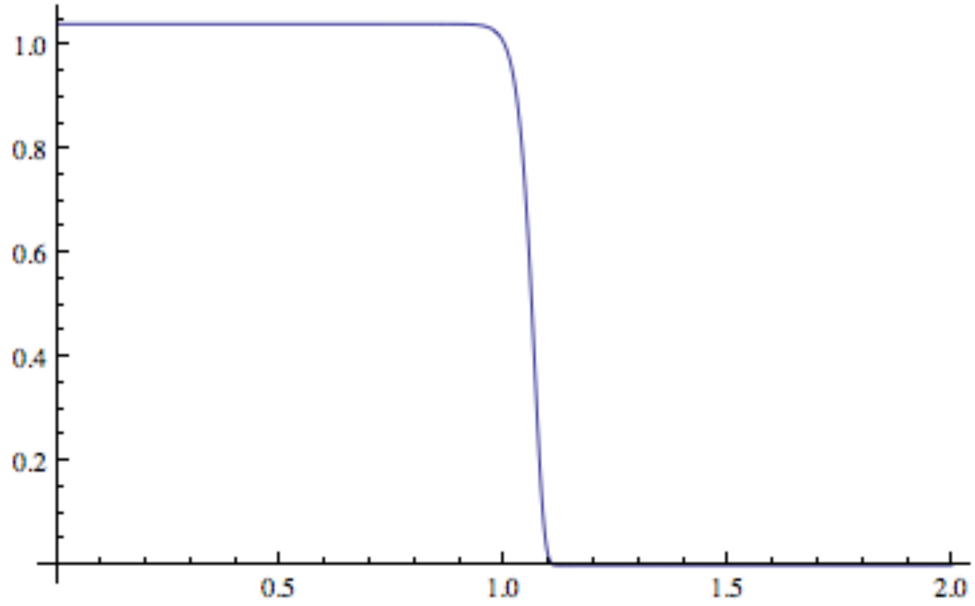
```
rhoE[r_] := rhos * Exp[- 2/alpha ((r/rs)^alpha - 1)]

rhos = 1;
rs = 1;
alpha = 20;

Plot[rhoE[x], {x, 0, 2*rs}, PlotRange → All]
```



Task 1a - Answers

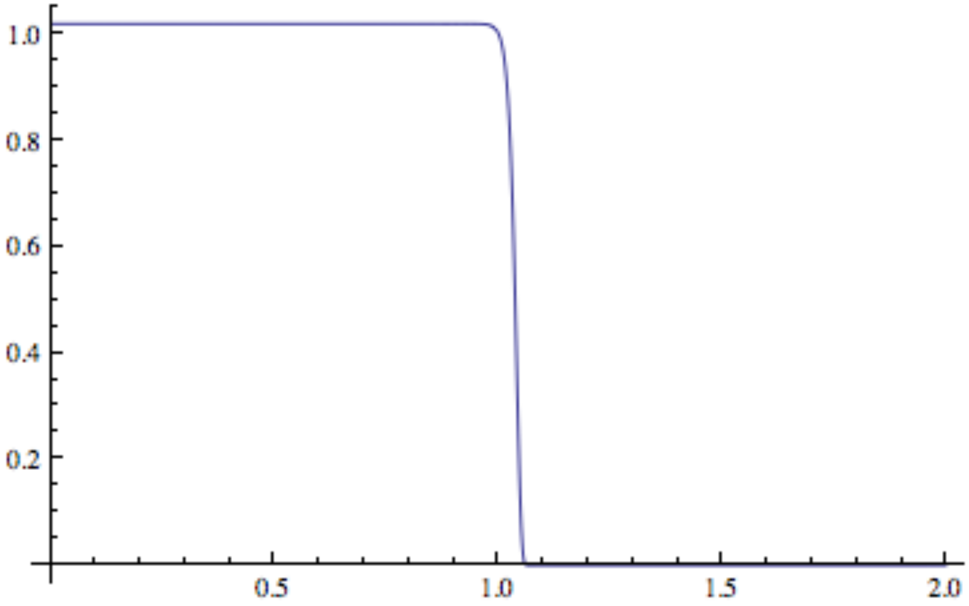
$$\text{rhoE}[r\_] := \text{rhos} * \text{Exp}\left[-\frac{2}{\text{alpha}}\left(\left(\frac{r}{\text{rs}}\right)^{\text{alpha}} - 1\right)\right]$$

```
rhos = 1;
rs = 1;
alpha = 50;

Plot[rhoE[x], {x, 0, 2*rs}, PlotRange → All]
```



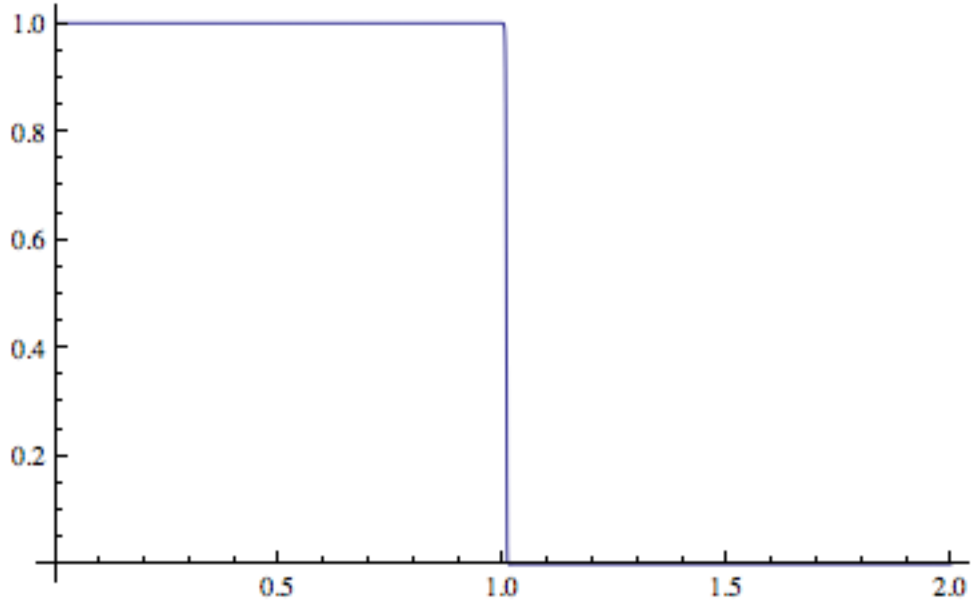
Task 1a - Answers

```
rhoE[r_] := rhos * Exp[-2/alpha ((r/rs)^alpha - 1)]

rhos = 1;
rs = 1;
alpha = 100;

Plot[rhoE[x], {x, 0, 2*rs}, PlotRange → All]
```



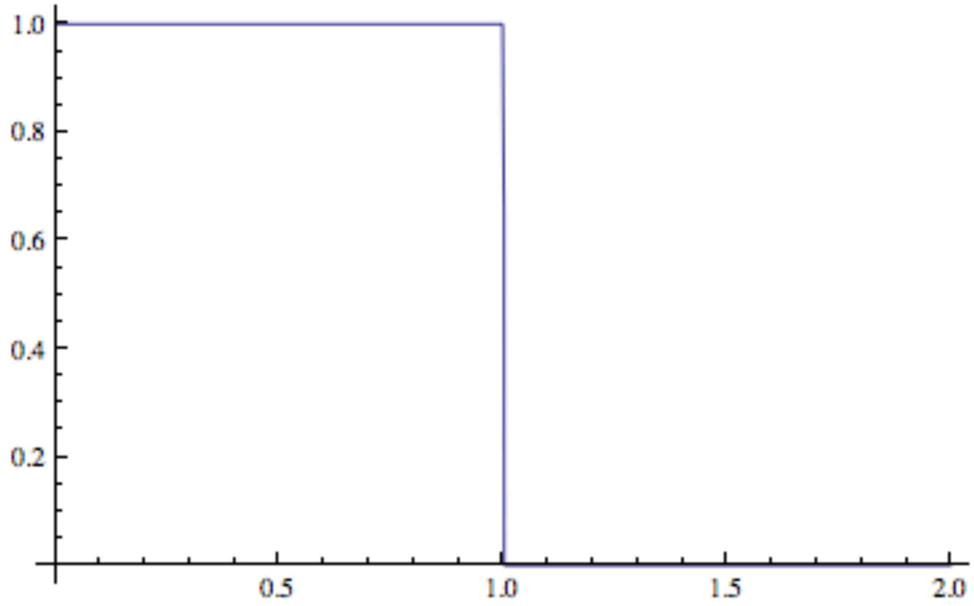
Task 1a - Answers

```
rhoE[r_] := rhos * Exp[- 2/alpha ((r/rs)^alpha - 1)]

rhos = 1;
rs = 1;
alpha = 1000;

Plot[rhoE[x], {x, 0, 2*rs}, PlotRange → All]
```



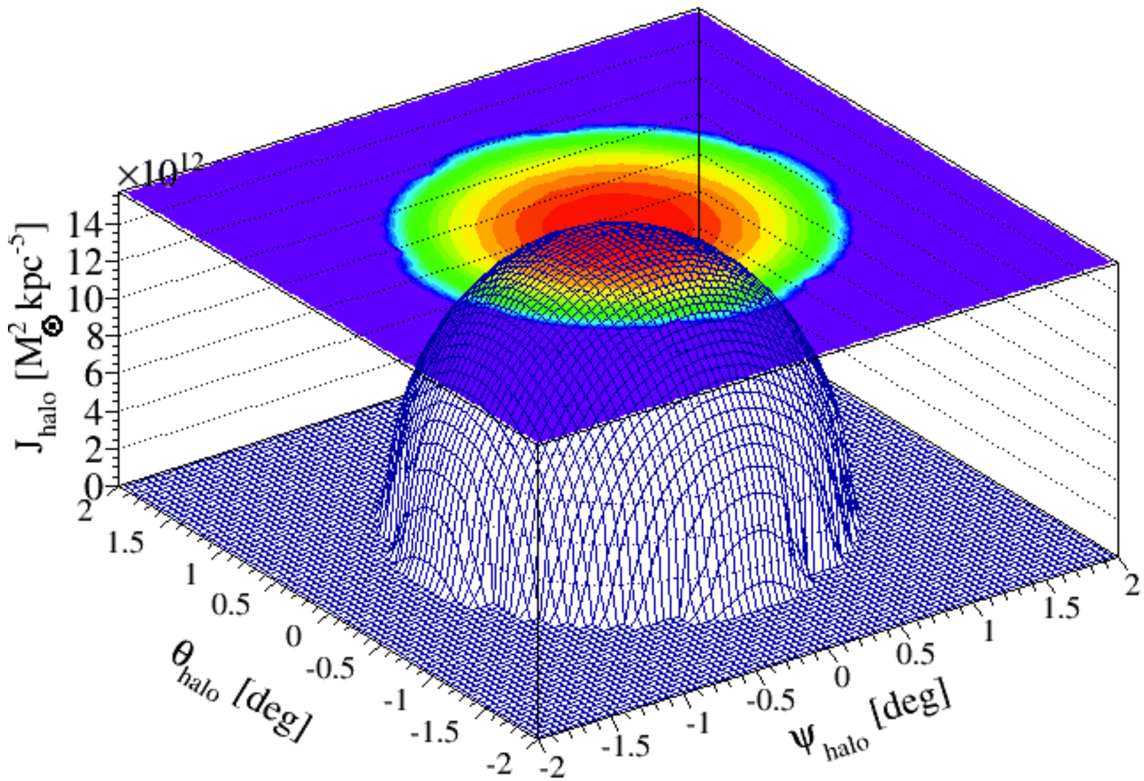
Task 1a - Answers

```
rhoE[r_] := rhos * Exp[-2/alpha ((r/rs)^alpha - 1)]

rhos = 1;
rs = 1;
alpha = 10000;

Plot[rhoE[x], {x, 0, 2*rs}, PlotRange → All]
```

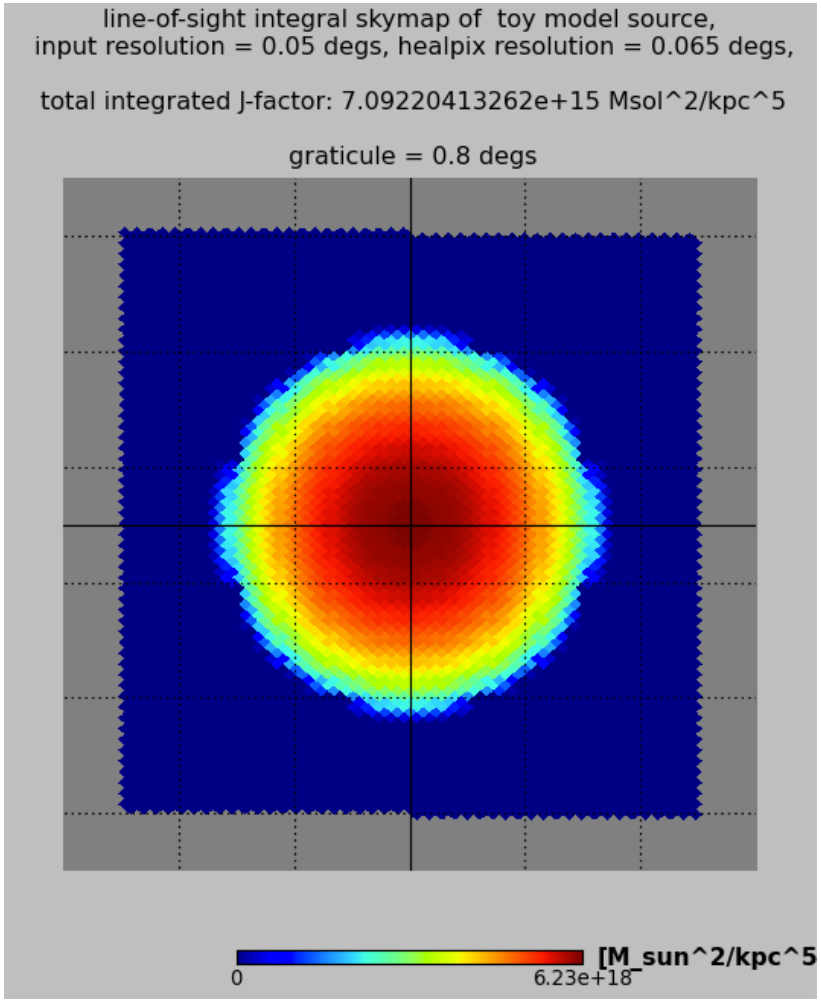


Task 1b - Answers

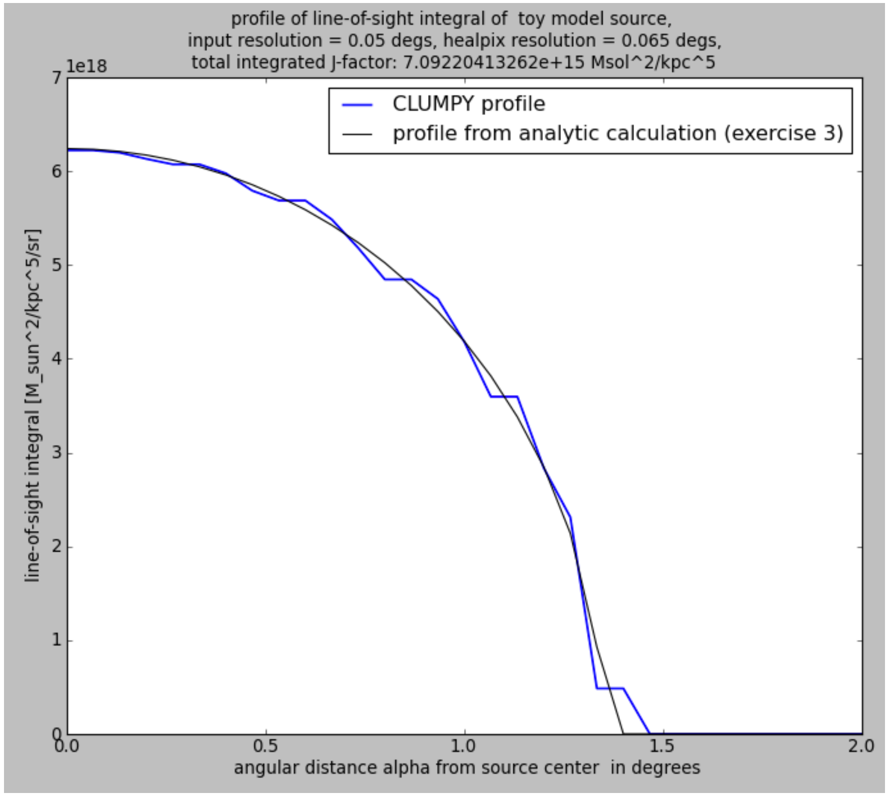
J-factor skymap plot by clumpy (ROOT):



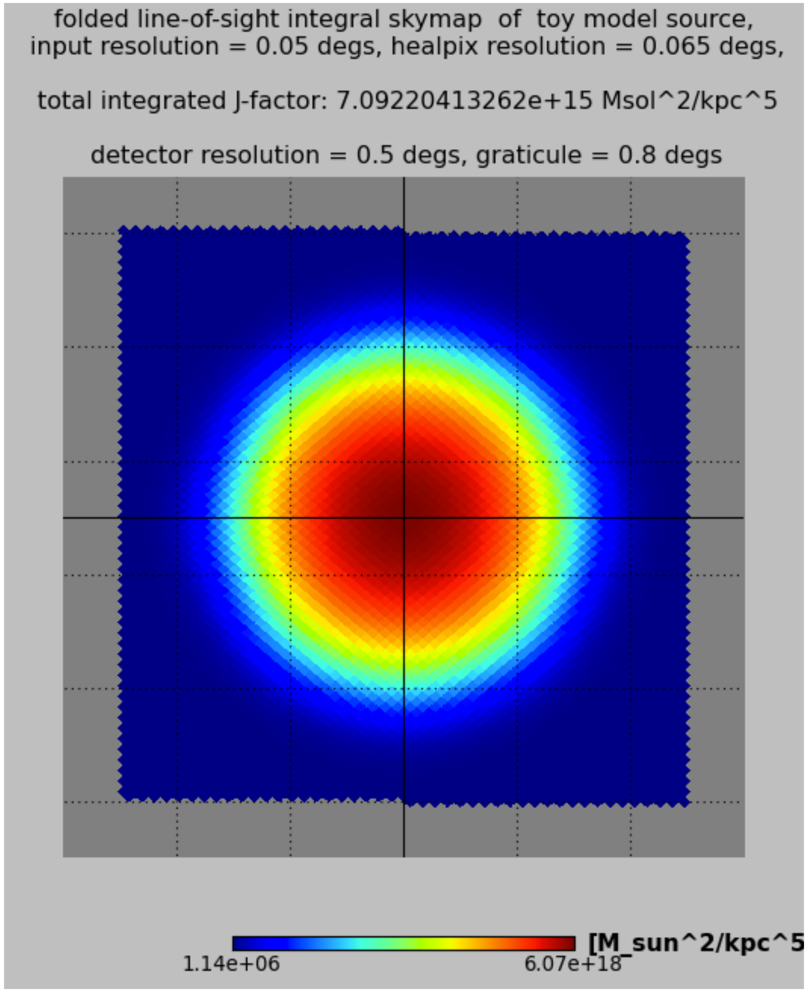
Task 1b - Answers



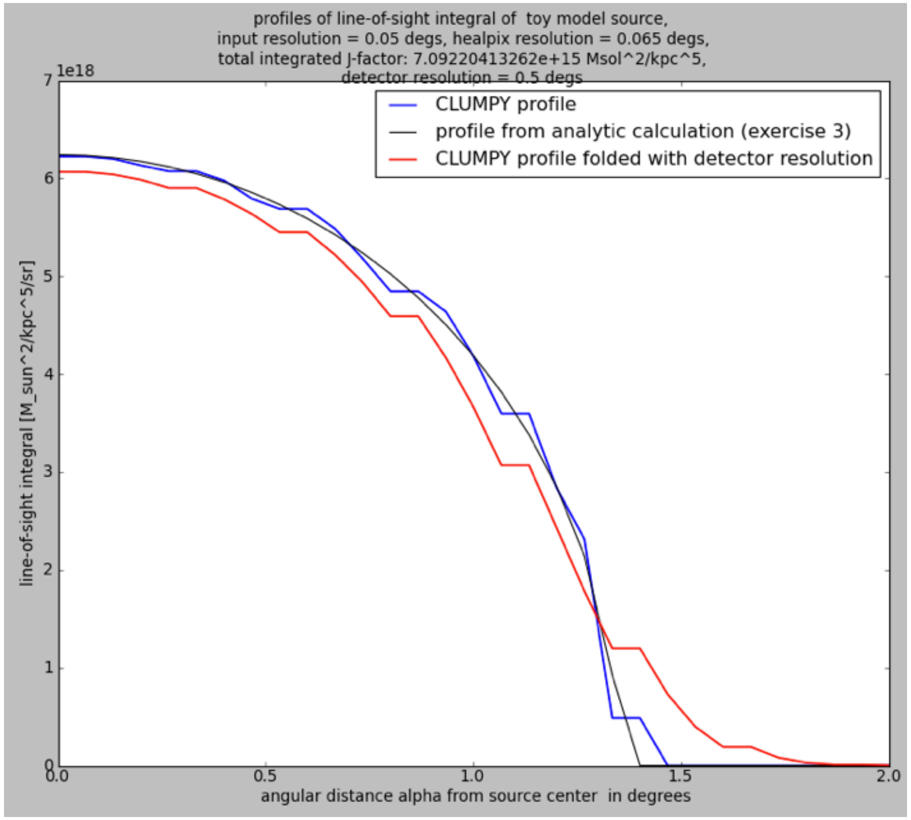
Plots with healpy (Python) & integrated J-factor:



Task 1b - Answers

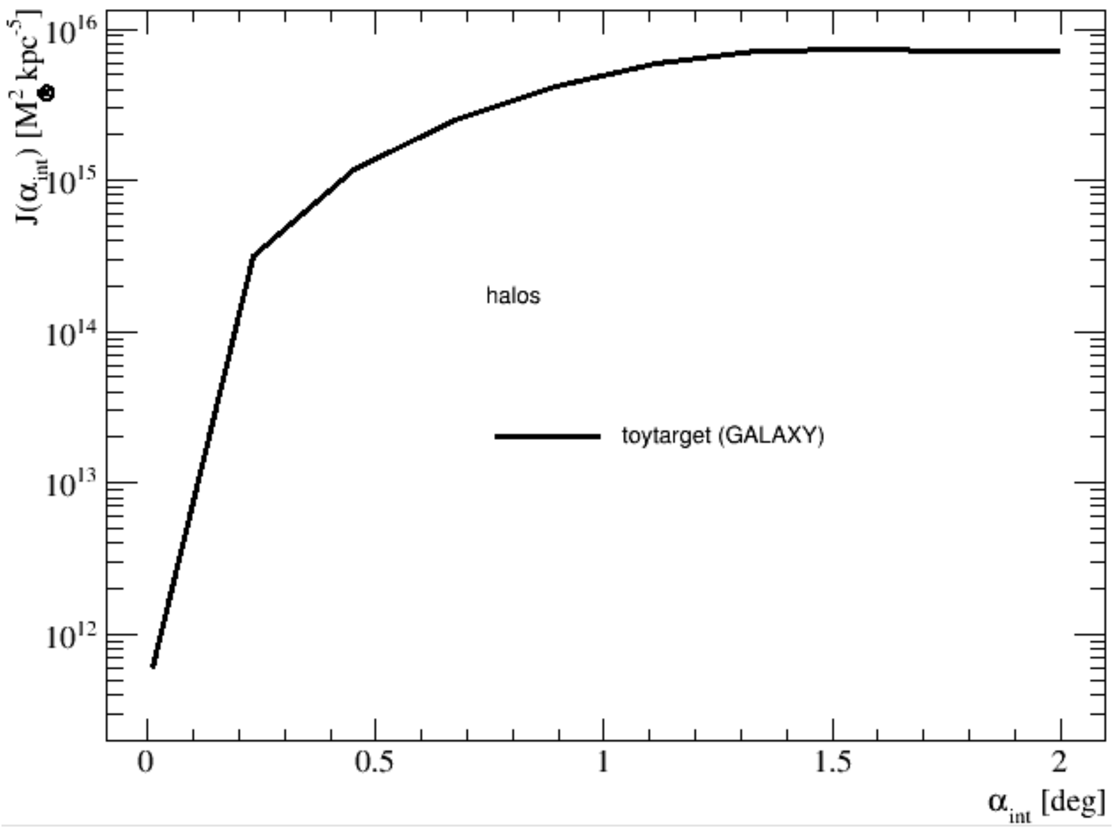


line-of-sight sky map folded with detector resolution:



Task 1b - Answers

clumpy h2 mode result (precision 1.e-3):



Task 1b - Answers

Precision of the numerical result:

CLUMPY precision e-2 (h5 mode): J = 7.(1) Msol^2/kpc^5
CLUMPY precision e-3 (h2 mode): J = 7.2(4) Msol^2/kpc^5
CLUMPY precision e-4 (h2 mode): J = 7.24(0) Msol^2/kpc^5

Analytic result: J = 7.2366 Msol^2/kpc^5

Keep in mind that:

- The constant-density sphere is approximated by an Einasto profile (although sufficiently well for the demanded precision)
- The non-steady density boundary causes numerical problems

Task 2 - Answers

Get $\frac{d\Phi}{dE}$ from $\frac{dN}{dE}$ and J-factor :

In[642]:= $J = 7.09\times10^{15}\,\frac{\text{msun}^2}{\text{kpc}^5};$

$\sigma v = 3.0\times10^{-26}\,\frac{\text{cm}^3}{\text{s}};$

$m = 0.5\ \text{TeV};$

In[660]:= $\text{kg} = \frac{\text{msun}}{1.9891\times10^{30}};$

$\text{eV} = 1.782662\times10^{-36}\ \text{kg};$

$\text{TeV} = 10^{12}\ \text{eV};$

$\text{kpc} = 3.0857\times10^{21}\ \text{cm};$

$\text{kg} = 1;$

$\text{cm} = 1;$

$\text{s} = 1;$

Task 2 - Answers

Fornengo (2004) spectrum:

Annihilation into u, u_bar/d,d_bar, m_chi = 500 GeV:

```
In[667]:= a = -1.5;
          b = 0.047;
          c = -8.7;
          d = 9.14;
          e = -10.3;
```

```
In[679]:= dNdx[x_] := x^a * Exp[b + c * x + d * x^2 + e * x^3]
```

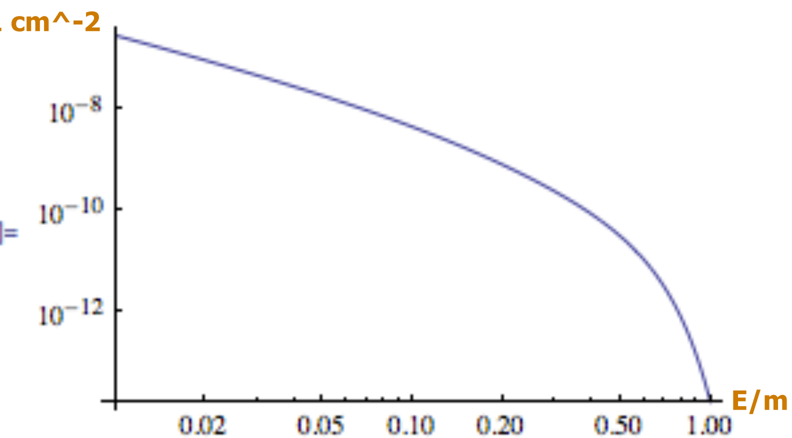
$$\text{factor} = \frac{1}{8\,\text{Pi}} \times \frac{\sigma v}{m^2} \times J$$

Out[680]= 1.50659×10^{-10} s^-1 cm^-2

```
In[44]:= dΦdx[x_] := factor × dNdx[x];
```

$$\text{LogLogPlot}\left[\frac{\text{TeV}}{m} \times d\Phi dx[x], \{x, 0.01, 1\}\right]$$

Out[45]=




```
In[46]:= Φ = NIntegrate[dΦdx[x], {x, 0.2, Infinity}]
```

Out[46]= 3.48552×10^{-11} **photons s^-1 cm^-2**

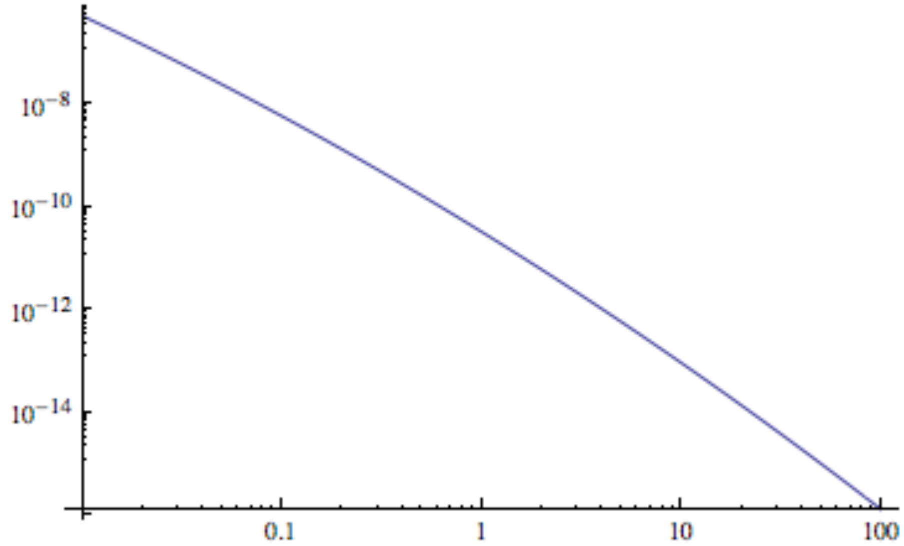
Task 2c - Answers

Compare with Crab nebula flux:

MAGIC (2011):

```
In[684]:= dPhidx[x_] := 3.27 * 10^(-11) x^(-2.4 - 0.15 * Log10[x])
          LogLogPlot[dPhidx[x], {x, 0.01, 100}]
```

Out[685]=




```
In[687]:= PhiCrab = NIntegrate[dPhidx[x], {x, 0.1, Infinity}]
```

Out[687]= 4.84182×10^{-10} **photons s^-1 cm^-2**

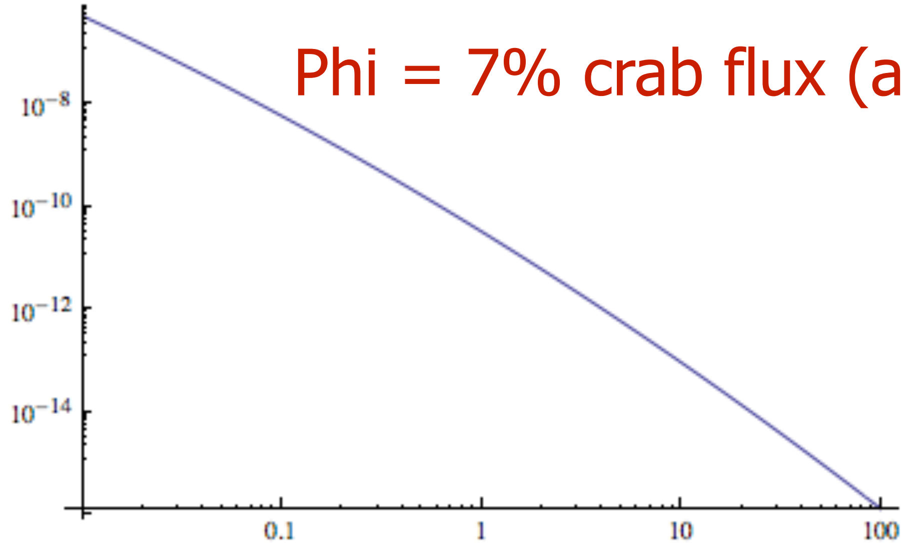
Task 2c - Answers

Compare with Crab nebula flux:

MAGIC (2011):

In[684]:=
```
dPhidx[x_] := 3.27*10^(-11) x^(-2.4 - 0.15*Log10[x])
LogLogPlot[dPhidx[x], {x, 0.01, 100}]
```

Out[685]=



Phi = 7% crab flux (above 100 GeV)

In[687]:=
```
PhiCrab = NIntegrate[dPhidx[x], {x, 0.1, Infinity}]
```

Out[687]= 4.84182×10^{-10} **photons s^-1 cm^-2**

The End

That`s it.