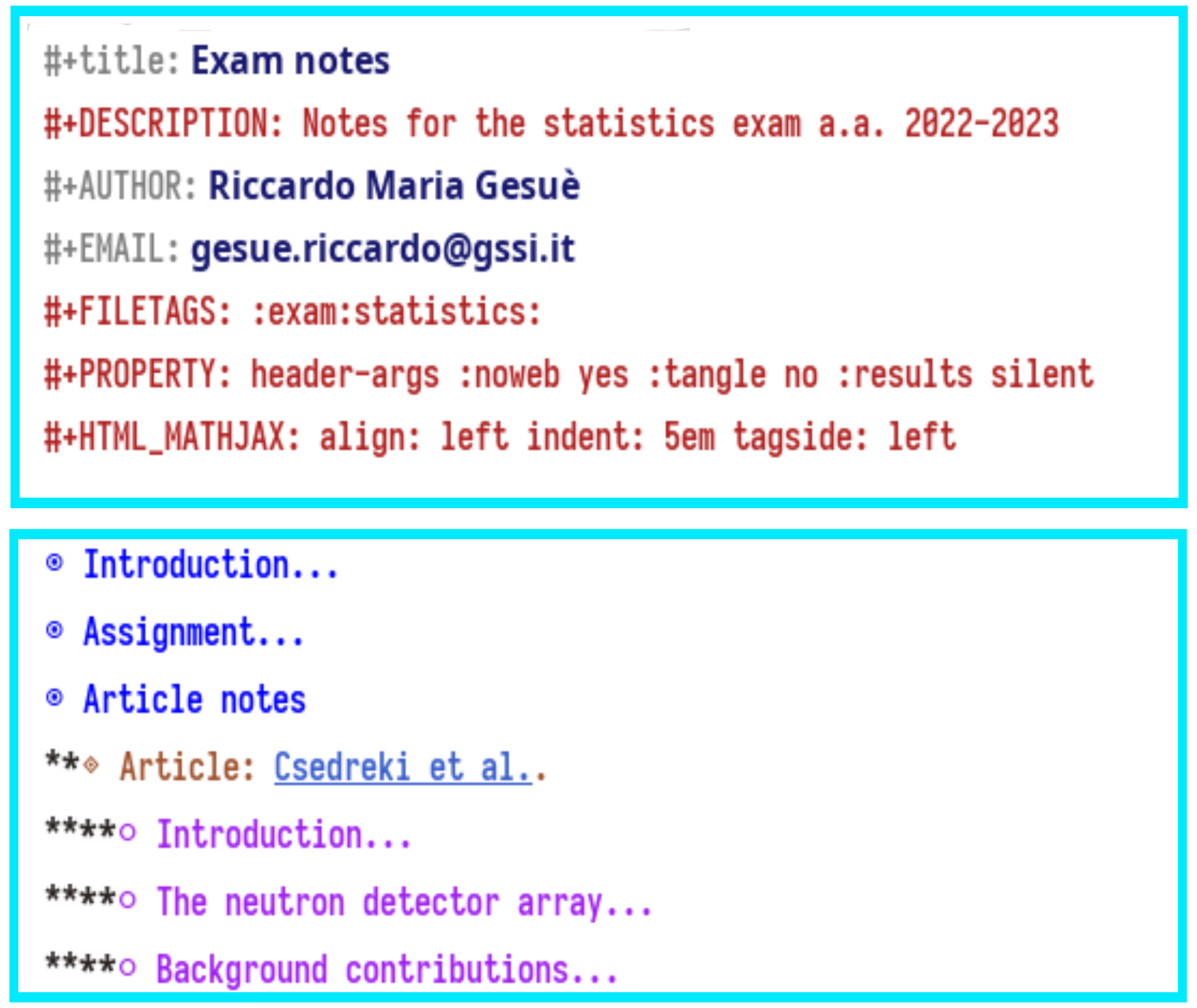
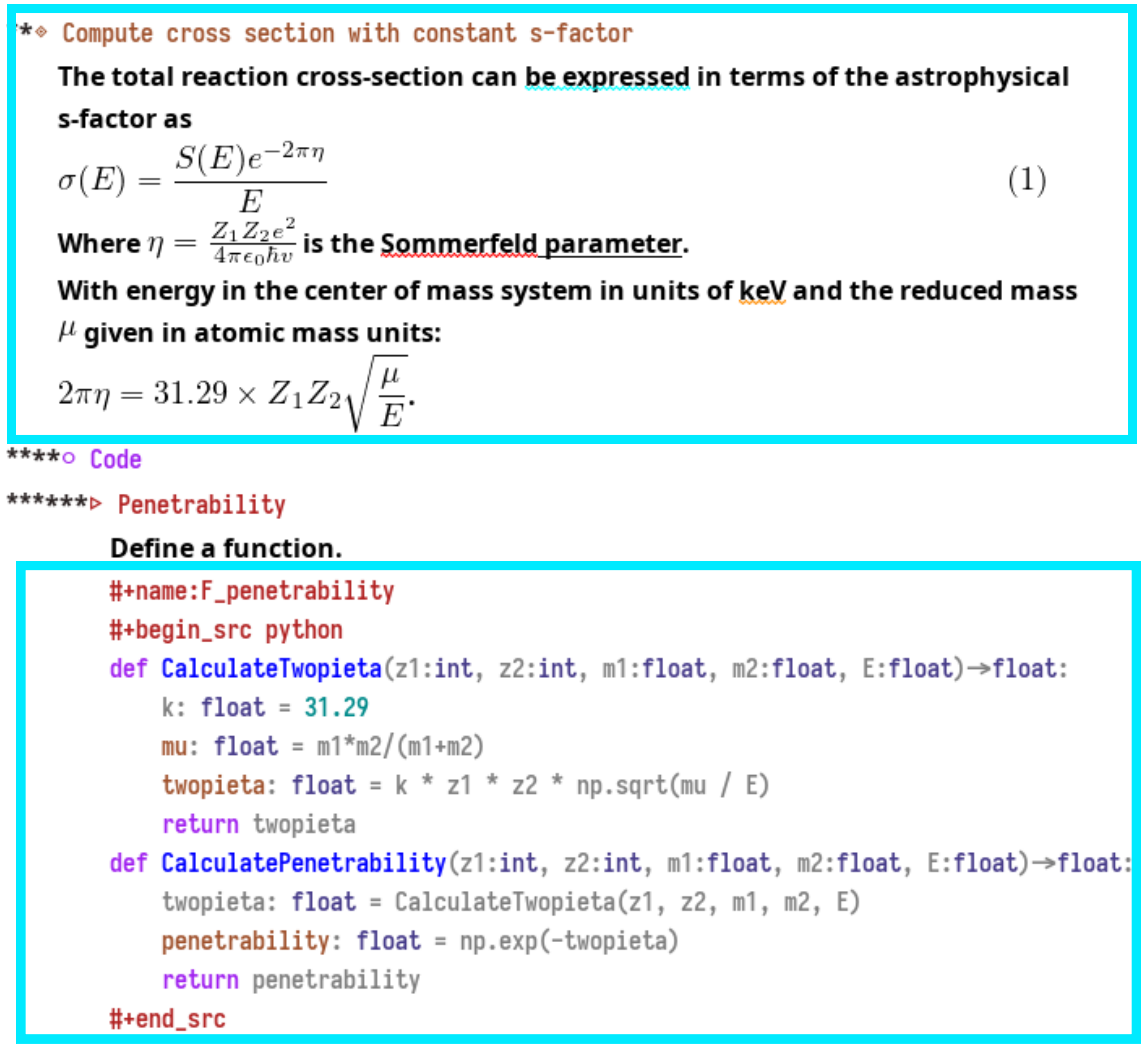


```
#+title: Exam notes
#+DESCRIPTION: Notes for the statistics exam a.a. 2022-2023
#+AUTHOR: Riccardo Maria Gesuè
#+EMAIL: gesue.riccardo@gssi.it
#+FILETAGS: :exam:statistics:
#+PROPERTY: header-args :noweb yes :tangle no :results silent
#+HTML_MATHJAX: align: left indent: 5em tagside: left
```

```
⊙ Introduction...
⊙ Assignment...
⊙ Article notes
**◦ Article: Csedreki et al..
****○ Introduction...
****○ The neutron detector array...
****○ Background contributions...
```

Metadata

Sections



*◦ Compute cross section with constant s-factor

The total reaction cross-section can be expressed in terms of the astrophysical s-factor as

$$\sigma(E) = \frac{S(E)e^{-2\pi\eta}}{E} \qquad (1)$$

Where $\eta = \frac{Z_1 Z_2 e^2}{4\pi\epsilon_0 \hbar v}$ is the Sommerfeld parameter.

With energy in the center of mass system in units of keV and the reduced mass μ given in atomic mass units:

$$2\pi\eta = 31.29 \times Z_1 Z_2 \sqrt{\frac{\mu}{E}}.$$

****○ Code

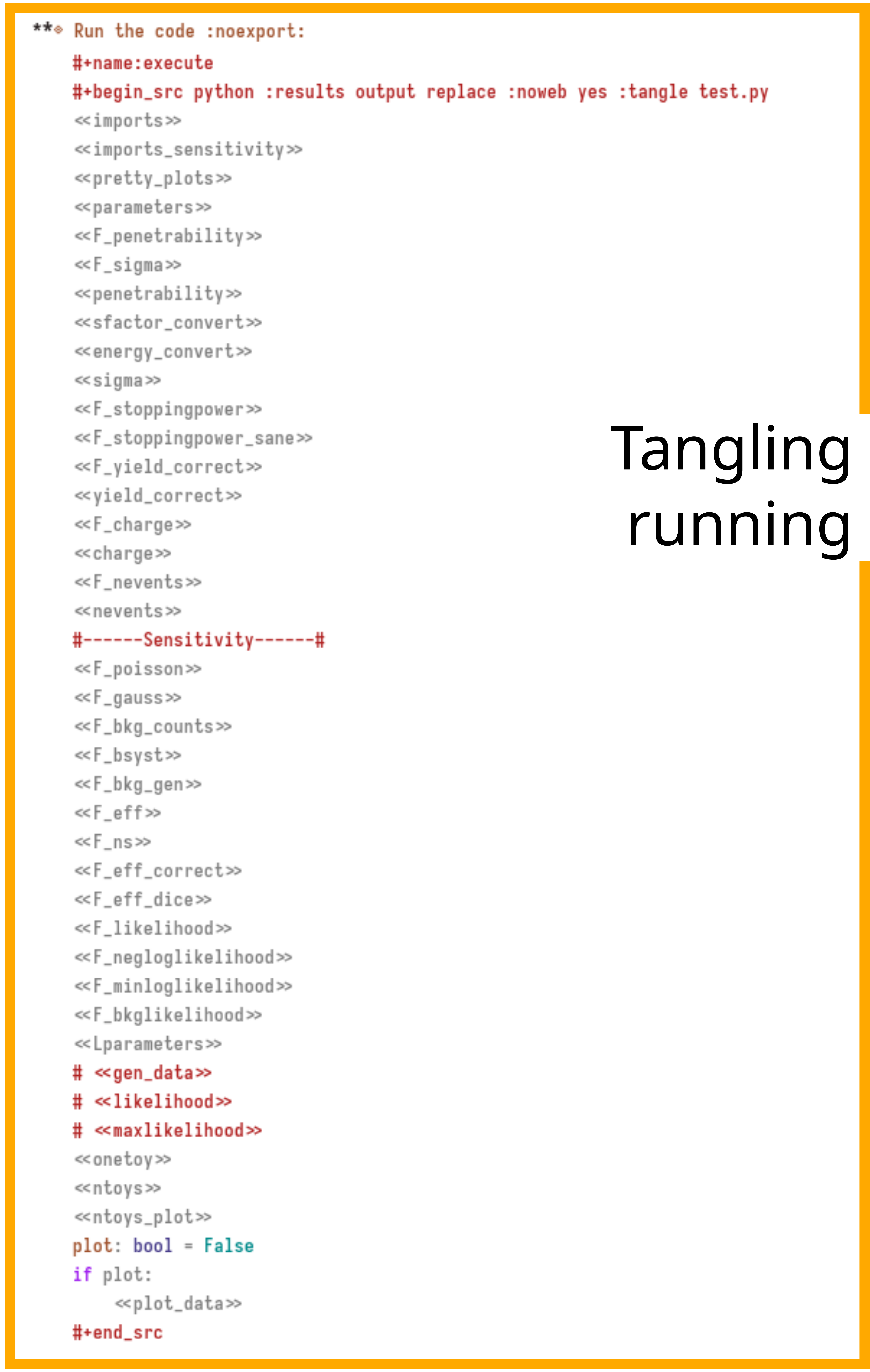
******▷ Penetrability

Define a function.

```
#+name:F_penetrability
#+begin_src python
def CalculateTwopieta(z1:int, z2:int, m1:float, m2:float, E:float)→float:
    k: float = 31.29
    mu: float = m1*m2/(m1+m2)
    twopieta: float = k * z1 * z2 * np.sqrt(mu / E)
    return twopieta
def CalculatePenetrability(z1:int, z2:int, m1:float, m2:float, E:float)→float:
    twopieta: float = CalculateTwopieta(z1, z2, m1, m2, E)
    penetrability: float = np.exp(-twopieta)
    return penetrability
#+end_src
```

Text with LaTeX formatting

Code snippets



```
**◦ Run the code :noexport:
   #+name:execute
   #+begin_src python :results output replace :noweb yes :tangle test.py
   <<imports>>
   <<imports_sensitivity>>
   <<pretty_plots>>
   <<parameters>>
   <<F_penetrability>>
   <<F_sigma>>
   <<penetrability>>
   <<sfactor_convert>>
   <<energy_convert>>
   <<sigma>>
   <<F_stoppingpower>>
   <<F_stoppingpower_sane>>
   <<F_yield_correct>>
   <<yield_correct>>
   <<F_charge>>
   <<charge>>
   <<F_nevents>>
   <<nevents>>
   #------Sensitivity------#
   <<F_poisson>>
   <<F_gauss>>
   <<F_bkg_counts>>
   <<F_bsyst>>
   <<F_bkg_gen>>
   <<F_eff>>
   <<F_ns>>
   <<F_eff_correct>>
   <<F_eff_dice>>
   <<F_likelihood>>
   <<F_negloglikelihood>>
   <<F_minloglikelihood>>
   <<F_bkglikelihood>>
   <<Lparameters>>
   # <<gen_data>>
   # <<likelihood>>
   # <<maxlikelihood>>
   <<onetoy>>
   <<ntoys>>
   <<ntoys_plot>>
   plot: bool = False
   if plot:
       <<plot_data>>
   #+end_src
```

Tangling and running the code