

Introduction

- pynucastro is an open-source python library with the main goal of providing visualization of rates and nuclei properties, given a particular set of rates or the nuclei involved in the reaction network of interest, and/or its composition.

- In addition, pynucastro provides an interface of the first-order stiff differential system of equations, generated by the selected reaction network, to C++ and python. The C++ interface also provides GPU offloading techniques, making it suitable for Hydro+Reactive source calculations.

$$\frac{dY_i}{dt} = \rho N_A \langle \sigma v \rangle_{ijk\dots} Y_i Y_j Y_k \dots$$

- In this new version of pynucastro, we have implemented high temperature burning support for $\sim 10^9$ K environments, including the nuclear partition function in baryonic inverse rates calculations, the nuclear statistical equilibrium (NSE) composition determination, and implementations for classical electron screening.

Inverse Rates

- In pynucastro we use the REACLIB database from which all the important rates are selected. In pynucastro, for each baryonic forward reaction (exothermic), we compute its inverse rate (endothermic) by the use of detailed balance:

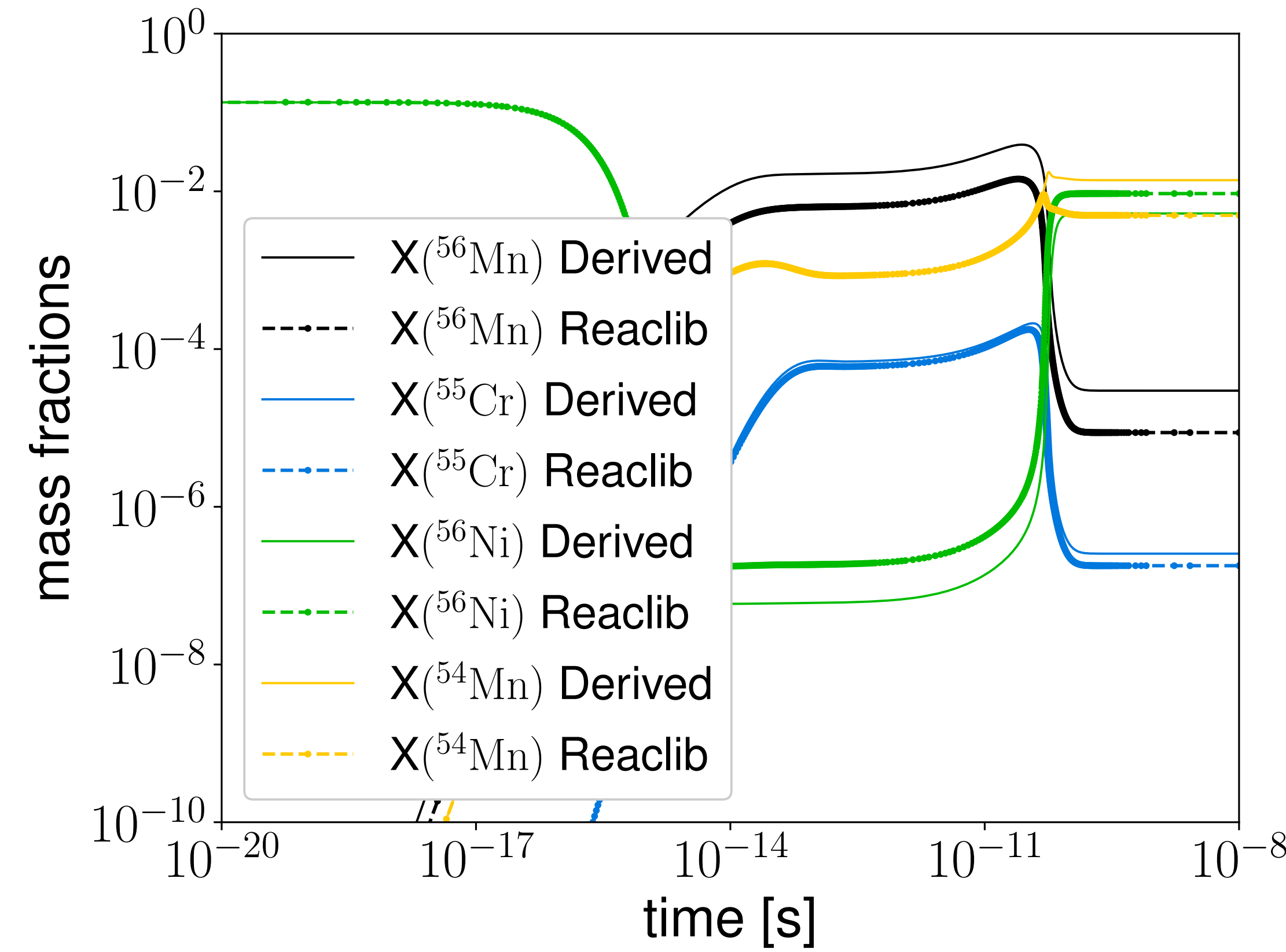
$$N_a^{p-1} \langle \sigma v \rangle_{B_1, B_2, \dots, B_p} = \left(\frac{1}{N_a} \right)^{r-p} \frac{(2J_{C_1} + 1) \dots (2J_{C_r} + 1)}{(2J_{B_1} + 1) \dots (2J_{B_p} + 1)} \times \left(\frac{A_{C_1} \dots A_{C_r}}{A_{B_1} \dots A_{B_p}} \right)^{3/2} \left(\frac{m_u}{2\pi\hbar^2} \right)^{\frac{3}{2}(r-p)} T^{\frac{3}{2}(r-p)} \times \frac{G_{C_1} \dots G_{C_r}(T) \prod_{B_i} c_{B_i}!}{G_{B_1} \dots G_{B_p}(T) \prod_{C_i} c_{C_i}!} e^{-Q/T} \times N_a^{r-1} \langle \sigma v \rangle_{C_1, C_2, \dots, C_r}$$

where r and p are the number of reactants and products respectively, G_{N_i} are the nuclear partition functions of the nuclei N_i and the remaining quantities are just temperature independent.

References

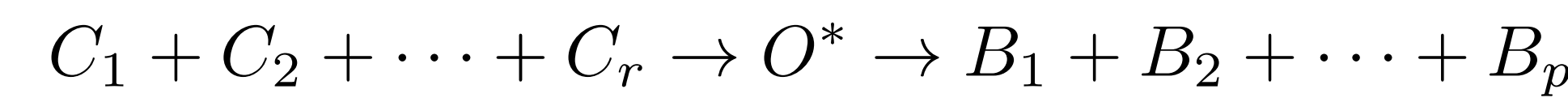
- <https://github.com/pynucastro/pynucastro>
- <https://iopscience.iop.org/article/10.3847/1538-4357/acbaff>
- https://github.com/aisclark91/poster/blob/main/pynucastro_astronum.pdf

- The comparison between REACLIB inverse rates and the inverse rates effects obtained by pynucastro is depicted as follows:



Coulomb Screening

- Under the assumption of linear mixing rule (Dewitt et.al 1973), the classical enhancement factor of the reaction rate,



is given by

$$f_{C_1 \dots C_r \rightarrow O^*}^{\text{cl}} = \exp \left[\frac{\mu^C(Z_{C_1}) + \dots + \mu^C(Z_{C_r})}{T} - \frac{\mu^C(Z_{C_1} + \dots + Z_{C_r})}{T} \right]$$

where $\mu^C(Z_{C_i})$ is the free-energy excess per particle of the one component plasma (OCP) made of a negative charge background and ions of atomic number Z_{C_i} .

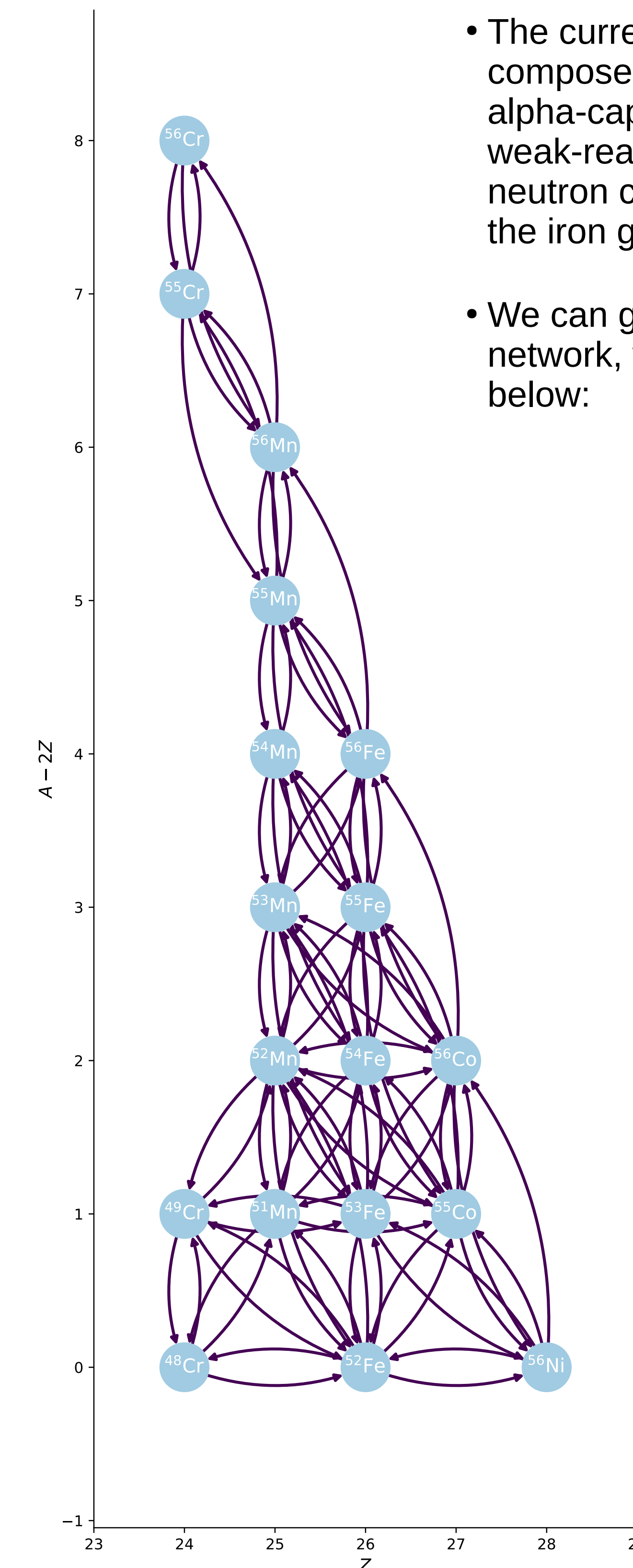
- Currently, pynucastro have support for two screening routines for $\mu^C(Z_{C_i})$ (Potekhin et.al 1998, Chugunov et.al. 2007, 2009). However, there are plans to include more routines, including quantum effects (Kushnir 2019, Chugunov 2021).

Download Us



Writing a Network

- As an example of pynucastro, we can write the following network:



- The current network is composed by successive alpha-capture reactions, weak-reactions and neutron captures near the iron group

- We can generate this network, with the snippet below:

```
1 import pynucastro as pyna
2
3 rl = pyna.ReacLibLibrary()
4 fwd = rl.derived_forward()
5
6 nuclei = ["n", "p", "he4",
7           "cr48", "cr49", "cr55",
8           "cr56", "mn51", "mn52",
9           "mn53", "mn54", "mn55",
10          "mn56", "fe52", "fe53",
11          "fe54", "fe55", "fe56",
12          "co55", "co56", "ni56"]
13
14 frates = fwd.linking_nuclei(nuclist=nuclei,
15                             with_reverse=False)
16
17 derived = []
18 for r in frates.get_rates():
19     d = pyna.DerivedRate(rate=r, compute_Q=False,
20                           use_pf=True)
21     derived.append(d)
22
23 der_rates = pyna.Library(rates=derived)
24 full_library = frates + der_rates
25
26 pynet = pyna.PythonNetwork(libraries=full_library,
27                             do_screening=True)
```

NSE Composition

- pynucastro provides a method to compute the mass fraction composition of a network once the nuclear statistical equilibrium is reached:

$$X_i = \frac{m_i}{\rho} (2J_i + 1) G(T) \left(\frac{m_i T}{2\pi\hbar^2} \right)^{3/2} \times \exp \left[\frac{Z_i \mu_p^{\text{id}} + N_i \mu_n^{\text{id}} + Q_i}{T} + \frac{Z_i \mu_p^C - \mu_i^C}{T} \right]$$

- The difference between computing the mass fraction NSE composition + screening and by integrating the network, can be seen as follows:

