A Technical Note on calculating charge state abundances in neutron star mergers

Pranav Nalamwar*
FRIB: `https://frib.msu.edu`
Facility for Rare Isotope Beams
Michigan State University
Version: 0.02

July 7, 2020

Big Picture Intro

Supernovae, or the explosions marking the deaths of heavy stars, formed the heavier elements through the rapid neutron capture process, or r-process(see Fig.1 for formed elements). Nuclei capture neutrons in a neutron-rich environment, creating a radioactive isotope that later decays into stable ones. However, a new class of astronomical event seems to be the source of the remaining r-process elements: neutron star mergers. These neutron star mergers(NSMs) are often the most neutron-rich events in the universe, so they are necessary for the formation of the rather heavy elements, most notably the lanthanides and actinides. Understanding all the details to this event will uncover why there is a large abundance of elements like gold and uranium. There have been two confirmed NSMs, most notably GW170817 in 2017. This merger also had an associated kilonova, the optical transient AT2017gfo. The kilonova light curve, which shows its brightness over time, is a direct result of r-process elements, especially the lanthanides. How bright and in what wavelengths the kilonova is in is determined by the composition and distribution of the lanthanides. By studying its optical spectra(from here on, spectra refers to optical spectra), one can deduce which elements were created in the merger event as these atoms emit light contributing to this observed spectra.

Understanding how the r-process operates in NSMs is necessary to detailing how these heavy elements form. Our goal is to look at which ionization states and isoelectronic states of which elements dominate the merger mixture at late times. Between 1 hour to 2 weeks after the merger, many of the newly formed isotopes don't have significant variations in their abundance, indicating they contribute both to the optical spectra observed and the overall abundance of the heavy elements. The kilonova light curve for GW170817 is rather detailed but due to the numerous contributing lines present, the ionization states specific to the lanthanides that are important to the late time merger material should be studied. Many NSM and kilonova codes use varying atomic databases for opacity and spectra calculations, so by uncovering which ionization states are important, we can measure their properties in a laboratory to benchmark all the atomic calculations used by these codes.

Attacking the Problem

To accurately produce the necessary charge state abundances for the elements in consideration, the Saha equation is utilized. The Saha equation relates the abundances of different charge states of one element based on the temperature, electron fraction, ionization potential, and density. Skynet (Lippuner & Roberts, 2017), a nuclear reaction network code designed to output isotopic abundances of r-process elements formed from neutron star mergers, is utilized in conjunction with the Saha equation. The derivation is provided below.

---

*nalamwar@frib.msu.edu

This sheet uses the Saha equation to find out the charge state abundances based solely on data provided by Skynet.

Constraint on elemental abundances of element Z:

$$Y_Z(t) = \sum_{I=0}^{Z} Y_{Z,I}(t)$$

where $Y_Z(t)$ is the total abundance of element Z and $Y_{Z,I}$ is the abundance of Z in ionization state I.

$$Y_{e,bound} = \sum_{Z=0}^{Z_{max}} \sum_{I=0}^{Z} (Z-I) Y_{Z,I}(t) = (1-f) Y_{e,tot}(t)$$

Note: Z_{max} is for summing over all possible elements while the second sum is summing over all possible ionization states of Z.

The value of f is introduced such that $Y_{e,free} = f Y_{e,tot}$, where $Y_{e,free}$ is the electron free fraction and $Y_{e,tot}$ is the total electron fraction in the mixture, both of which are fractions between 0 and 1. However, since f is an expensive calculation, just note f as a fraction between 0 and 1. This value will be calculated later in the Saha python class

Let's now take the Saha equation, which is converted to abundances rather than densities using the values of density:

$$\frac{Y_{Z,I+1}}{Y_{Z,I}} = \frac{2}{\rho N_A * Y_{e,tot} * f} \left(\frac{G_{Z,I+1}}{G_{Z,I}} \right) \left(\frac{m_e k_b T}{2\pi \hbar^2} \right)^{\frac{3}{2}} e^{\left(\frac{-\chi_i}{k_b T} \right)}$$

Note: ρ = density in g/cm^3, N_A = Avogadro's Number = $6.02214076 * 10^{23} \, mol^{-1}$, m_e = mass of an electron, $.511 \, MeV$, k_b = Boltzman Constant = $1.38064852 * 10^{-23} J/K$, T = temperature in GK, $\hbar$ = Reduced Planck's Constant = $6.582119569 * 10^{-16} eV \cdot s$, and χ_i = Ionization Potential in eV.

The function G is the partition function, but since most of these states are unknown, we cannot focus on it. Instead, we assume the partition function ratio is approximately 1. We will condense the entire right side of the equation into a function $g_i(p, Y_{e,tot}, T)$.

This next section will purely be calculations and equations:

$$Y_{Z,I+1} = Y_{Z,I} * g_I \implies Y_{Z,1} = Y_{Z,0} * g_0$$

$$Y_{Z,2} = Y_{Z,1} * g_1 = Y_{Z,0} * g_1 * g_0$$

such that

$$Y_{Z,I} = Y_{Z,0} \prod_{m=0}^{I-1} g_m \tag{1}$$

Put in constraint on Y_Z :

$$Y_Z = \sum_{I=0}^{Z} Y_{Z,0} \left(\prod_{m=0}^{I-1} g_m \right)$$

$$\implies Y_{Z,0} = \frac{Y_Z}{\sum_{I=0}^{Z} \prod_{m=0}^{I-1} g_m} \tag{2}$$

Note that the right hand side of the equation can be calculated purely from Skynet output and data about ionization potentials, which means you can use equation (1) to get all the $Y_{Z,I}$ for a single Z.

In total, you well have a function like this:

$$func(T, p, Y_{e,free}, [\chi_i], Y_Z)$$

Now, while these equations are correct, they do result in overflow and underflow issues when graphing the abundances towards the very beginning of the abundance calculations. Therefore, instead of altering the ionization state each time g_m is calculated, focus on just one ionization state.

Rewrite the expression $\prod_{m=0}^{I-1} g_m$ as h_I, which is a function of T, p, $Y_{e,free}$

Then, $Y_{Z,I} = h_I Y_{Z,0}$. Say another ionization state j is chosen, thus $Y_{Z,J} = h_J Y_{Z,0}$
Therefore, the relation $\frac{Y_{Z,I}}{Y_{Z,J}} = \frac{h_I}{h_J} \implies Y_{Z,I} = Y_{Z,J} \frac{h_I}{h_J}$

We can finally rewrite the elemental abundance Y_Z as $Y_Z = \sum_{I=0}^{Z} Y_{Z,I} = Y_{Z,J} \sum_{I=0}^{Z} \frac{h_I}{h_J}$.

Rewriting this gives

$$Y_{Z,J} = \frac{Y_Z}{\sum_{I=0}^{Z} \frac{h_I}{h_J}} \tag{3}$$

Equation (3) follows the same structure as equation (2). In fact, they are identical for J=0. The main difference is that equation (3) utilizes a constant h_J, indicating the calculation need not go through every element but only it's own element. This helps solve overflow and underflow issues for early times/high temperatures during the r-process.

Equation (3) is useful in finding the electron fraction and the charge state abundances for each time step. To do these calculations, only one element was chosen at first:Samarium since the data and the abundance shapes were compared to Christopher Fontes' plots (Fontes et al., 2017). Two elements were then used, Sm and Eu, in a multiple element mixture calculation. Once this was checked for validity, the abundance function, which requires user input for the number of elements they want, was constructed.

Code Breakdown

The abundance function is the culmination of the single and multiple element codes, The first part of the code uses a function called ionization generator. It requires as input an array of elements by their charge number and outputs a multi-dimensional array of each element's ionization potential based on its ionization state. The basic structure of this output is an array of arrays; the first element of the returned ionization potential array is empty, the second element contains all the potentials for hydrogen, and so on. All data is retrieved from a NIST database, which is then put into a .xlsx file in order to be read by the python program. The ionization generator function is needed to remove any odd or empty values present in the database.

Temperature values are taken directly from Skynet, which are calculated for times up to approximately 31.79 years after the merger or at $10^{-2}K$. Skynet truncates the temperature evolution for late times, we a monotonic fit, a linear relationship between temperature and time in log-log space, was used instead. To make this fit, numpy.polyfit was used for late times, thus ensuring the most accurate slope. Currently, this is not a function but rather hard-coded, so this must be adjusted to account for varied skynet parameters.
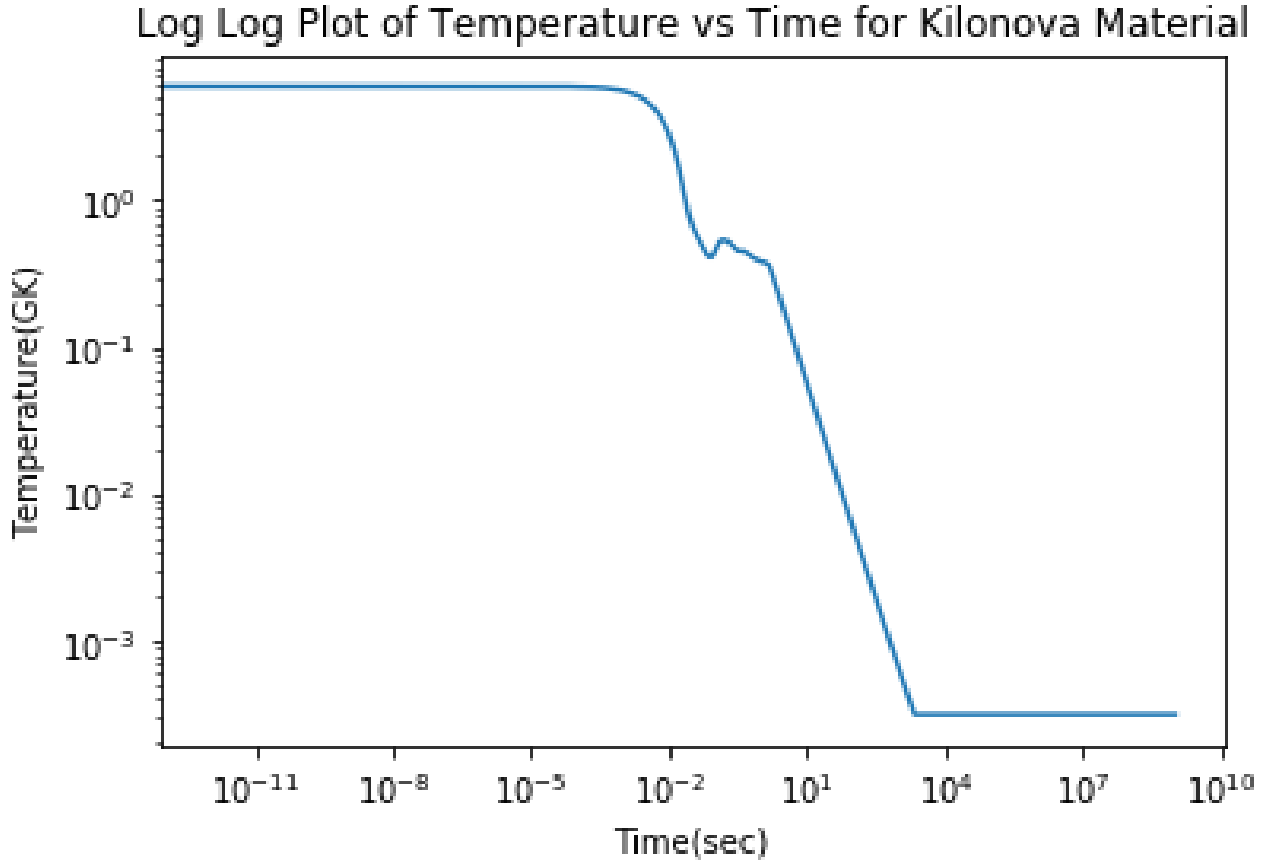
Figure 1: Plot of temperature vs time with temperature in units of GK. This is data extracted directly from Skynet. For large times, Skynet truncates any temperature calculation, resulting in this flat curve in late times. Temperature ranges from $10^1\,GK$ to $10^{-4}\,GK$.
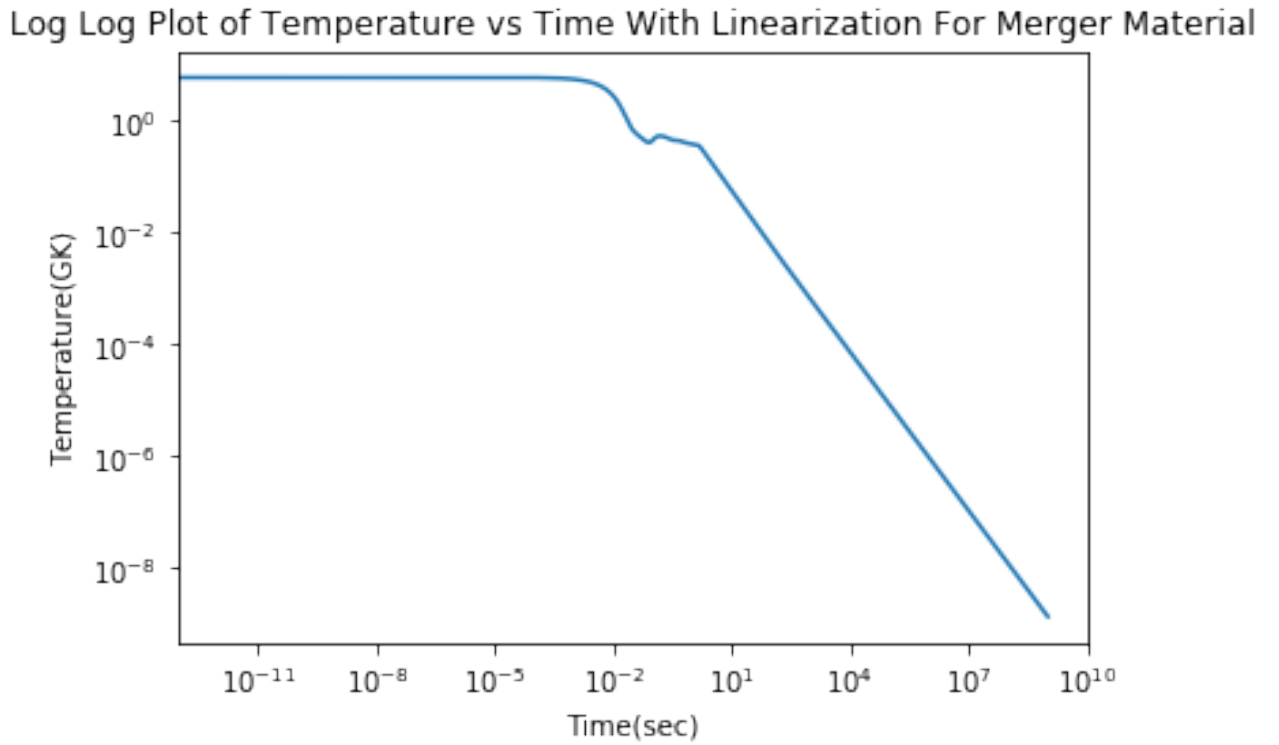

Figure 2: Plot of temperature vs time for the merger material after linearizing the plot in loglog space. Note now that the temperature ranges from $10^1\,GK$ to $10^{-9}\,GK$.

The initialization function exists solely to take all the isotopic abundances of a given element and sums it up into the total elemental abundance. For convenience, the output of the function includes the ionization potentials as well as an array of elemental abundances for all the elements in consideration across all times.

The abundance calculation function uses as input the ionization potentials, the temperature array, the density array, and the total abundances vs time for use in the saha_mult class. This will return the ionization state abundances for each element across time. Note that each element will contain Z+1 abundance arrays to account for the fully ionized state as well.

The Saha_mult class is designed to calculate the free electron fraction of the merger material across time. Skynet outputs the total electron fraction, the temperature array, and the isotopic abundances. Using this, the Saha_mult class can follow equations (1) - (3) to calculate the ionization state abundances. The most crucial step besides following the equations is finding the $Y_{e,free}$. The main assumption here is that the merger material is charge neutral, indicating the number of protons and electrons must be equal. This allows a Ye contribution to be calculated. Now, using a small and large guess for the $Y_{e,free}$, a bisection method can be used to find the midpoint between the low and high electron fractions. The idea is that this bisection method will eventually calculate a 0 crossing, which will be the actual $Y_{e,free}$. This is done for each timestep, so now the GetAbundancesFixedYef function, which is a form of equation (3), can be used to return all the desired ionization state abundances. Some pseudo code is provided below for context.

Pseudo Code
Basic Premise: Set up a way to calculate electron fraction by considering charge neutrality. Guess some low and high values for a constant Ye_free, and plug them into this function to get some range of possible Ye_free values. Then, use a bisection method to find the optimal Ye_free for when a zero crossing occurs. Using this true value for Ye_free, plug into function that returns all ionization state abundances of the elements of interest.

```
Yef Contribution Part:


def GetYefContribution(...): #This will help predict Ye values
        for each element(elements array):
                Yc = getAbundanceFixedYef(...)
                #Yc = abundance of ionization states assuming constant Ye_f
                for each ion state(element Z+1):
                        Ye_free+=Yc * ion_state
                        #This finds the number of protons = number electrons
        return Ye_free


Bisection Part: #Note all Ye guesses are in log space for convenience


Guess Yef_high = 0 #Max Ye = 1 and log(1) = 0
Guess Yef_low = neg 50 # Any significantly low lnYef will work


if (Yef_high * Yef_low < 0): #Ensure parity are different to do bisection
        for i in range(number of iterations desired):
                mid = avg(high + low)
                check mid*high and mid*low
                set new low or high depending on any sign changes
        Ye_free = avg(final_Yef_low + final_Yef_high) #gives final Yef
```

Abundance Calculation Part:

```
for each element in (element array):
        return GetAbundancesFixedYef(... Ye_free)
        #return multiple dimensional array of abundances. Each element
          contains all ionization state abundances possible, including the
          free state
```

The plotter function is designed to show 5 unique plots: the relative ionization state abundances of all elements in the mixture vs time and temperature, the relative isoelectronic state abundances(based on number of electrons the atom has), and a graphical check depicting the ratio of the sum of all charge state abundances to the total abundance present. To calculate the relative abundance values, each charge state abundance is divided by the respective elemental abundance. A shift of index by 1 to the right in the ionization state abundance arrays was necessary to form the isoelectronic state abundances.

The final graph is the total abundance of the ionization states vs time. To form this graph, each ionization state abundance is summed together across all the available elements across time. Considering the kilonova peaks around 1 hour to 2 weeks, that became the timescale of these charge state graphs. The graphed formed in the sample code focused solely on the lanthanide abundances.

Analysis and Results

The lanthanide graphs reveal many features about how the merger material evolves, especially showing the distribution of elemental and charge state abundances. The elemental abundances in the regime when the kilonova peaks indicates dysprosium as the dominant species. Also, note the fluctuations in each element's abundance. While each species' abundance doesn't change in an order of magnitude, the ranking of lowest to greatest abundance oscillates.
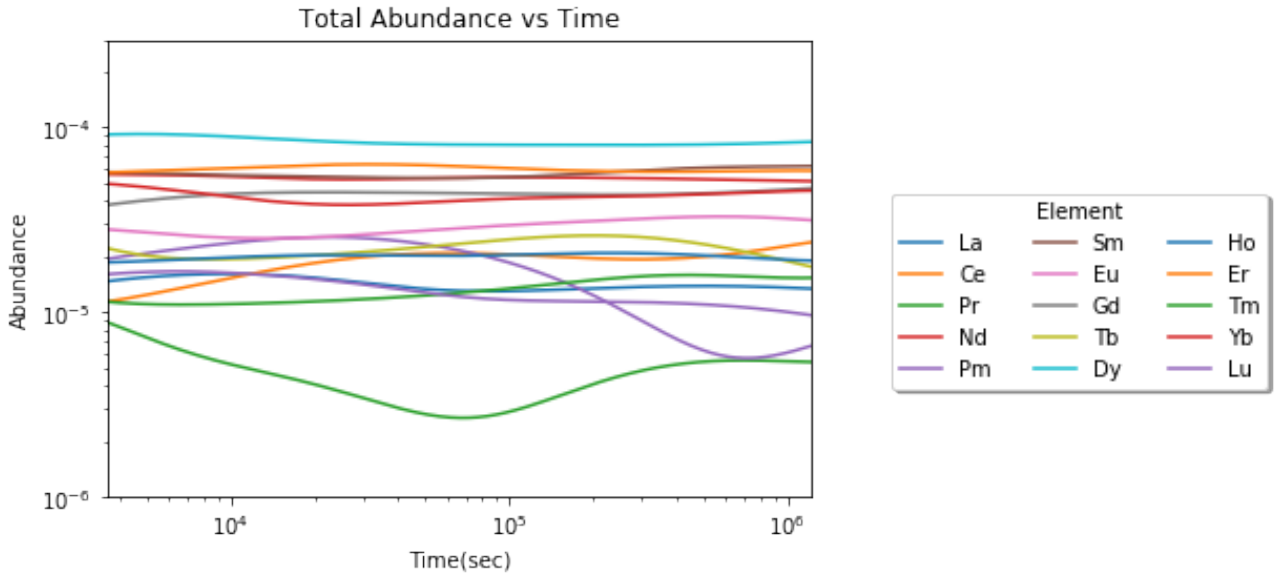

Figure 3: Plot of abundances for all lanthanides over time (left) based on output from Skynet. Note that the abundances at early times is noise considering there are values lower than $10 - 20$. Values become significant around 10^{-1} sec

The ionization state abundances can be calculated for any number of elements considered in the mixture, but they all have the same structure. For the test element samarium, it is fully ionized at high temperatures, fully neutral at low temperatures, and changes state in between with multiple peaks representing the point where that specific ionization state is the dominant species for the element.
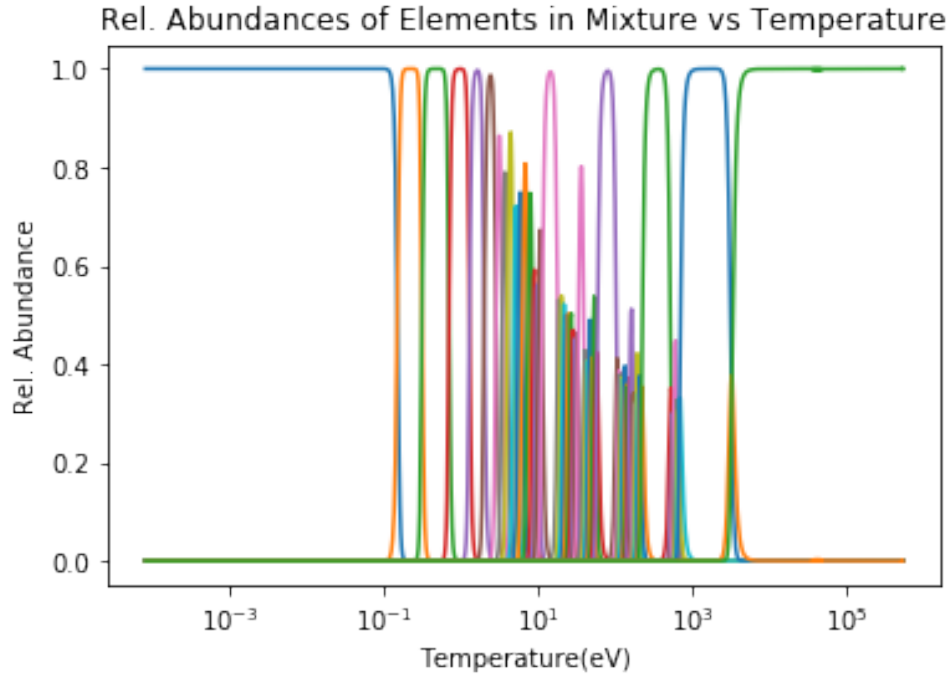
Figure 4: Relative abundance graph of samarium and 63 unique ionization states vs temperature. Note the changes in the peaks.

Once all the ionization state abundances were summed up,the plot was limited based on when the kilonova peaks(1 hour to 2 weeks). Much like the individual elements' ionization state plots, the total abundance of ionization state abundances involves numerous peaks indicating the dominant species at a given time. 17 states are considered dominant at late times.
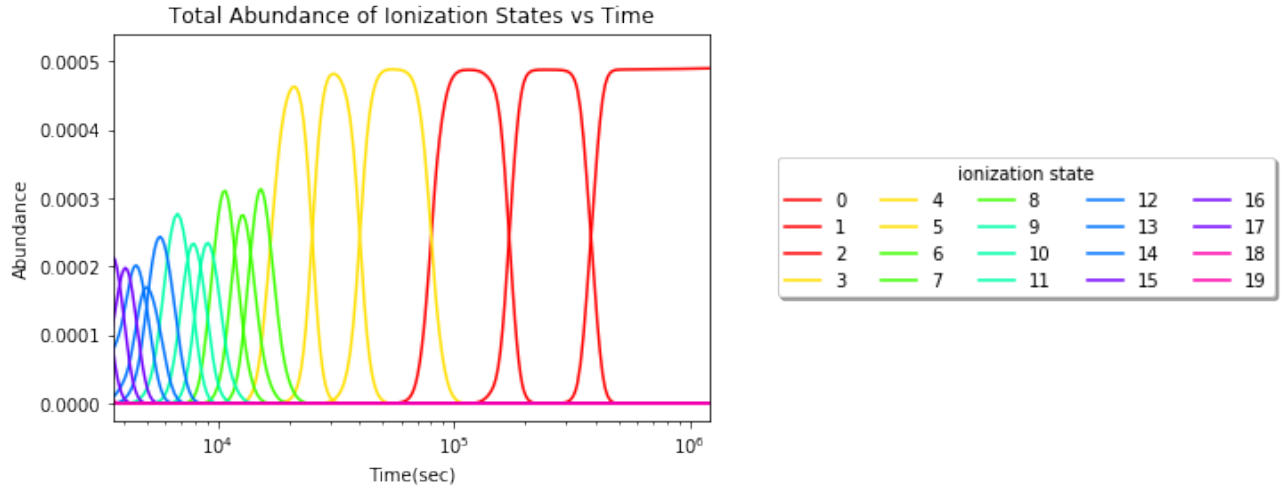

Figure 5: Relative abundance graph of all the lanthanides' ionization state abundances as a function of time in the merger material. The various colors represent the unique ionization states and the first 17 states are dominant at cool temperatures.

Conclusion

Based on the analysis of the abundance graphs, many of the lanthanides' abundances become significant after the r-process occurs. Once this happens, each element undergoes numerous atomic transitions, resulting in certain ionization states to become dominant in the merger material until the neutral state dominates for low temperatures/late times. The total ionization state abundances of the lanthanides indicate that 17 ionization states are important from one hour to two weeks, but only the neutral state and the first two ionization states are significant on the orders of days to weeks.

References

Fontes, C. J., Fryer, C. L., Hungerford, A. L., Wollaeger, R. T., Rosswog, S., & Berger, E. (2017). *A line-smeared treatment of opacities for the spectra and light curves from macronovae.*

Lippuner, J., & Roberts, L. F. (2017). *Skynet: A modular nuclear reaction network library.*