12-2010

Libstatmech and Applications to Astrophysics

Tianhong Yu
Clemson University, tyu@clemson.edu

Libstatmech and Applications to Astrophysics

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Sciense
Physics

---

by
Tianhong Yu
December 2010

Accepted by:
Dr. Bradley S. Meyer, Committee Chair
Dr. Mark D. Leising
Dr. Murray S. Daw

Abstract

In this work an introduction to Libstatmech is presented and applications especially to astrophysics are discussed. Libstatmech is a C toolkit for computing the statistical mechanics of fermions and bosons, written on top of libxml and gsl(GNU Scientific Library). Calculations of Thomas-Fermi Screening model and Bose-Einstein Condensate based on libstatmech demonstrate the expected results. For astrophysics application, a simple Type Ia Supernovae model is established to run the network calculation with weak reactions, in which libstatmech contributes to compute the electron chemical potential and allows the weak reverse rates to be calculated from detailed balance. Starting with pure ^{12}C and $T_9 = 1.8$, we find that at high initial density ($\rho \sim 9 \times 10^9 g/cm^3$) there are relatively large abundances of neutron-rich iron-group isotopes (e.g. ^{66}Ni, ^{50}Ti, ^{48}Ca) produced during the explosion, and Y_e can drop to ~ 0.4, which indicates that the rare, high density Type Ia supernovae may help to explain the ^{48}Ca and ^{50}Ti effect in FUN CAIs.

Dedication

I dedicate my thesis to the loved ones.

Rong,

With your support I am finishing my masters work. But without staying with you everyday I would finish it a long time ago! Just kidding. It's worth to be with you. Research is part of life. Family is also part of life.

To my parents,

Thanks for your reminding me about my masters each time on the phone. Now I am about to finish it finally! Now it's time for PhD.

Acknowledgments

I would like to thank Dr. Brad Meyer, my adviser for the opportunity to pursue this research. Dr. Meyer spent a lot of time with me to help develop programming skills and narrate me on key ideas of this project. His kind guidance led me into the gate of "coding the physics", which is really fun. I would also like to thank the members of my committee: Dr. Mark Leising gave me valuable advice after LG talks. Dr. Murray Daw taught me a different way to think about statistical mechanics. And I thank Dr. Lih-sin The and Dr. Jason Brown for their helpful input and support.

Table of Contents

List of Figures

Chapter 1

Introduction

The most neutron-rich stable iron-group isotopes (namely, ^{48}Ca, ^{50}Ti, ^{54}Cr, ^{58}Fe, and ^{62,64}Ni) present an important challenge for nucleosynthesis theory, Galactic chemical evolution studies, and for our understanding of the birth of the Solar System. With the exception of ^{48}Ca, they are significantly produced by slow neutron capture in massive stars (the weak *s*-process) (e.g., [13]). Such processing occurs primarily in the large helium-burning convective core that follows core hydrogen burning. As the star converts ^{4}He into ^{12}C and ^{16}O, abundant ^{14}N left over from hydrogen burning is converted to ^{22}Ne by the reaction sequence $^{14}\mathrm{N}+\alpha \rightarrow {}^{18}\mathrm{F}+\gamma$, $^{18}\mathrm{F} \rightarrow {}^{18}\mathrm{O}+e^{+}+\nu_{e}$, and $^{18}\mathrm{O}+\alpha \rightarrow {}^{22}\mathrm{Ne}+\gamma$. The ^{22}Ne thus produced then serves as a neutron source: $^{22}\mathrm{Ne}+\alpha \rightarrow {}^{25}\mathrm{Mg}+n$. Pre-existing nuclei (from previous generations of stars whose ashes were incorporated into the star under consideration) then capture these neutrons to make heavier elements. Such *s*-processing also occurs in helium burning environments in low-mass stars during their asymptotic-giant branch phases, but chemical evolution studies show that massive stars dominate the *s*-process contribution to the iron group (e.g., [13]).

The short beta-decay half of ^{45}Ca (162.2 days) and even shorter half of ^{47}Ca (4.536 days) prevent ^{48}Ca from being produced in the same environment. The site of ^{48}Ca synthesis is suspected to be a subset of thermonuclear supernovae (type Ia supernovae) that achieve high enough densities to have significant electron capture during the explosion (e.g., [10, 14]). Because the matter in these explosions has low entropy per nucleon (typically less than 0.1 k_B, where k_B is Boltzmann's constant, per nucleon), the nuclear abundances are characterized by an overabundance of heavy nuclei relative to nuclear statistical equilibrium. This permits production of abundant ^{48}Ca when

nuclear statistical equilibrium would require even larger production of ^{66}Ni, an isotope with nearly the same degree of neutron richness [6].

The type Ia supernovae that make ^{48}Ca also make ^{50}Ti, ^{54}Cr, and the other neutron-rich iron-group isotopes. The yields depend in large measure on the degree of neutron-richness achieved during the explosion. This, in turn, depends on the weak interaction rates on the nuclei. The interesting challenge for nucleosynthesis theory, then, is to understand the complex dynamics of the nuclear abundances as they shift due to both weak interactions and changing temperature and density.

Because the neutron-rich iron-group isotopes are made in at least two different sites, they also present a challenge to Galactic chemical evolution studies. Massive stars live and die quickly on astronomical timescales. For example, a star 25 times as massive as the Sun may be born, live, and die in a core-collapse supernova event over a time span of only 7 million years (e.g., [13]). A type Ia supernova is thought to be the explosion of a white dwarf star that has accreted enough mass to trigger a thermonuclear runaway that disrupts the entire star. The white dwarf formed from a lower mass star (less than roughly eight times the mass of the Sun) and then had to accrete matter from a companion star. The timescale for this to happen is hundreds of millions of years. The build up of abundance of an isotope like ^{50}Ti in the Galaxy is coming from two different sources operating on two different timescales while the build up of the ^{48}Ca abundance is apparently happening only on the longer timescale. The interesting challenge for Galactic chemical evolution is, then, to follow the relative build up of these isotopes over time, especially ^{48}Ca compared to its sister neutron-rich iron-group species.

These differing formation scenarios and different production timescales for the neutron-rich isotopes may have significant implications for the very early history of the Solar System. Calcium-aluminum-rich inclusions (CAIs), are refractory mineral inclusions found in primitive meteorites. They appear to be the first solid condensates (that is, the oldest "rocks") in the Solar System (e.g., [1, 3]). A small subset of CAIs are known as FUN (Fractionated and Unknown Nuclear effects) CAIs. FUN CAIs (and smaller grain hibonites) show roughly correlated anomalies in ^{48}Ca and ^{50}Ti [9, 11]. The anomalies are roughly correlated in the sense that CAIs or hibonites that show excesses (relative to average Solar composition) of ^{48}Ca tend also to show excesses of ^{50}Ti while those that show deficits of ^{48}Ca also show deficits of ^{50}Ti. A possible explanation for this is that the precursor dust of the Solar System had a heterogeneous (but correlated) distribution of theses isotopes. As

this precursor dust gathered to form the FUN CAIs, some sampled an excess of the dust with large quantities of the neutron-rich iron-group isotopes while others sampled a deficit of that dust. The challenge for Solar System studies is to characterize the precursor dust and mixing in the early Solar nebula to understand the origin of the FUN CAIs.

These elements may all be related. The rareness of the type Ia supernova events may lead to a heterogeneous distribution of the neutron-rich iron-group isotopes in the dust in the Galaxy. Processing of that dust in the interstellar medium will tend to homogenize the isotopes as supernova shocks sputter atoms from the dust which then reaccrete onto other dust grains. If that homogenization process is not too efficient, however, the precursor dust in the Solar System might indeed have been quite isotopically heterogeneous so that the anomalies in the FUN CAIs could be understood.

Unraveling the details of the history of the neutron-rich iron-group isotopes is clearly a large problem that will require years, if not decades, of work. In this thesis I contribute to this grand problem by focusing on the issue of production of these isotopes in type Ia supernovae. In particular, I developed some tools to study the nucleosynthesis in low-entropy explosive environments in which electron capture is changing the degree of neutron richness.

To carry out this project, I had to first develop a set of tools to compute the statistical mechanics of fermions. The result was libstatmech, which I describe in §2. With libstatmech, Prof. Bradley Meyer (my advisor) and I were able to develop libnuceq, a library for computing a variety of nuclear statistical equilibria. With libnuceq, I could then compute weak nuclear interaction rates from detailed balance. This was important because it allowed me to evolve nuclear reaction networks into weak nuclear statistical equilibrium, as described in §3. Finally, I applied the network code to a simple (zero dimensional) model of a type Ia supernova and followed the nucleosynthesis in §4. My hope is that the tools and insights I have developed can be applied to studies of nucleosynthesis in more detailed models of type Ia supernova, for example, those computed at the FLASH Center [7].

Chapter 2

libstatmech

2.1 Introduction to Libstatmech

Libstatmech is a library of C codes for computing the statistical mechanics of fermion and boson gases. It is written on top of libxml (GNOME C xml toolkit) and gsl (GNU Scientific Library).

It was released Aug 2009 at http://www.webnucleo.org/home/modules/libstatmech. The authors are Tianhong Yu and Bradley S. Meyer.

With libstatmech users can create fermions and bosons and calculate their thermodynamic quantities numerically with either default integrands (fully relativistic, non-interacting particles) or user-supplied ones or with user-defined functions (estimate Coulomb interaction with thermal motion somewhere, like star). Users can also define their own thermodynamic quantities as functions or integrands. A well-documented API allows users to incorporate libstatmech into their own codes, and examples in the libstatmech distribution demonstrate the API.

Here is a simple example to show how libstatmech works. Given electron's rest mass, its multiplicity (2 for electron), and its charge, the pressure of an electron gas can be calculated with input electron number density and temperature. The calculation takes both relativistic and degenerate situation into account.

2.2 Default Integrands

This section presents the default integrands used in libstatmech. They are for non-interacting, fully relativistic fermions and bosons (see A). The key parameters are m, the particle rest mass, g, the particle multiplicity, T, the Temperature, $\hbar$, Planck's constant divided by 2π, c, the speed of light in vacuum, and k, Boltzmann's constant. We use GSL defined values of these constants (see the GSL documentation) in all cases. We choose the cgs system of units. The other parameters are α and γ. These are defined as follows:

$$\alpha = \frac{\mu - mc^2}{kT} \equiv \frac{\mu'}{kT}, \tag{2.1}$$

where μ is the full chemical potential and μ' is the chemical potential less the particle's rest mass energy, and

$$\gamma = \frac{mc^2}{kT}. \tag{2.2}$$

2.2.1 Fermions

In deriving our integrands, we considered fermions and anti-fermions and assumed them to be in annihilation equilibrium with the photon field. Thus, for example, we assumed the reaction $e^+ + e^- \rightleftharpoons \gamma + \gamma$, where the γ represents a photon. Because the photon has zero chemical potential, the equilibrium implies

$$\mu_{e^+} = -\mu_{e^-}. \tag{2.3}$$

We may then write in terms of chemical potentials less the rest mass:

$$\mu'_{e^+} = -\mu'_{e^-} - 2mc^2, \tag{2.4}$$

where m is the electron rest mass. On division by kT, this then becomes

$$\alpha_{e^+} = -\alpha_{e^-} - 2\gamma. \tag{2.5}$$

The resulting integrands are the following (refer to Jason Brown's thesis [2]):

Number Density:

$$n_{e^+e^-} = \frac{(mc^2)^3 g}{2\pi^2(\hbar c)^3\gamma^3}\int_0^\infty (x+\gamma)\sqrt{x^2+2\gamma x}\left[\frac{1}{1+\exp(x-\alpha)} - \frac{1}{1+\exp(x+2\gamma+\alpha)}\right] dx \quad (2.6)$$

Pressure:

$$P_{e^+e^-} = \frac{(mc^2)^4 g}{2\pi^2(\hbar c)^3\gamma^4}\int_0^\infty (x+\gamma)\sqrt{x^2+2\gamma x}\left\{\ln(1+\exp[\alpha-x]) + \ln(1+\exp[-x-2\gamma-\alpha])\right\} dx \quad (2.7)$$

Energy Density:

$$\epsilon_{e^+e^-} = \frac{(mc^2)^4 g}{2\pi^2(\hbar c)^3\gamma^4}\int_0^\infty (x+\gamma)^2\sqrt{x^2+2\gamma x}\left[\frac{1}{1+\exp(x-\alpha)} + \frac{1}{1+\exp(x+2\gamma+\alpha)}\right] dx \quad (2.8)$$

Entropy Density:

$$s_{e^+e^-} = \frac{k(mc^2)^3 g}{2\pi^2(\hbar c)^3\gamma^3}\int_0^\infty (x+\gamma)\sqrt{x^2+2\gamma x} \quad (2.9)$$

$$\left[\frac{x-\alpha}{1+\exp(x-\alpha)} + \ln[1+\exp(\alpha-x)] + \ln[1+\exp(-x-2\gamma-\alpha)] + \frac{x+2\gamma+\alpha}{1+\exp(x+2\gamma+\alpha)}\right] dx \quad (2.10)$$

In the equations above, $n_{e^+e^-}$ denotes the net electron number density ($n_{e^-} - n_{e^+}$). Because we will get electron number density from $\rho N_A Y_e$, where ρ is the mass density of materials, N_A is the Avogadro constant and Y_e is electron numbers per nucleon in the system of interest, which will be calculated by net positive charge in the system as we will see later.

All other quantities with subscript e^+e^- indicate the sum of contribution from electrons and positrons.

2.2.2 Bosons

In the absence of evidence for a conserved boson number, we neglected the anti-bosons. The resulting integrands are the following (derived in B):

Number Density:

$$n = \frac{(kT)^3 g}{2\pi^2(\hbar c)^3}\int_0^\infty \frac{(x+\gamma)\sqrt{x^2+2\gamma x}}{\exp(x-\alpha)-1}\, dx \quad (2.11)$$

Pressure:

$$P = -\frac{(kT)^4 g}{2\pi^2(\hbar c)^3}\int_0^\infty (x+\gamma)\sqrt{x^2+2\gamma x}\ln[1-\exp(\alpha-x)]\, dx \quad (2.12)$$

Energy Density:

$$\epsilon = \frac{(kT)^4 g}{2\pi^2(\hbar c)^3} \int_0^\infty \frac{(x+\gamma)^2\sqrt{x^2+2\gamma x}}{\exp(x-\alpha)-1}\, dx \tag{2.13}$$

Entropy Density:

$$s = -\frac{(kT)^3 g}{2\pi^2(\hbar c)^3} \int_0^\infty (x+\gamma)\sqrt{x^2+2\gamma x}\left\{\ln[1-\exp(\alpha-x)] + \frac{\alpha-x}{\exp(x-\alpha)-1}\right\} dx \tag{2.14}$$

As we will see later, we will do a root finding to get chemical potential with input number density and temperature. And then use the calculated chemical potential to get all other quantities. The functions in libstatmech to calculate a quantity are described in C.

2.3 Numerical Calculation of the Integrals

The integration variable for quantity integrands is, in effect, the magnitude of the particle momentum converted to energy in units of kT; thus, the full range of integration is from zero to ∞. Users may reset the integral lower limit with the updateIntegralLowerLimit API routines. The integrals are computed with Gnu Scientific Library (GSL) routines. It is efficient, when the $\mu'/kT > 0$, to integrate from the lower limit to μ'/kT and then from μ'/kT to ∞. The former integral is done with the GSL routine **gsl_integration_qags** while the latter is done with **gsl_integration_qagiu**. When $\mu'/kT \leq 0$, the full integration is done with **gsl_integration_qagiu**.

Numerical integration of a quantity continues until the approximation to the integral satisfies the absolute tolerance ϵ_{abs} and relative tolerance ϵ_{rel}. By default these are both 10^{-8} but the user may update them with the updateQuantityIntegralAccuracy() API routines. The default values provide an excellent compromise between accuracy and speed, and most users should probably not need to change them. Users who need the routines to be fast (up to ~ 3 times faster than the default calculations) and who can sacrifice some accuracy may wish to increase the tolerance numbers (tolerances of 10^{-2} lead to results that are still typically accurate up to about four decimal places).

2.4 Inversion of the Number Density Integral

It is often the case that one knows the temperature and number density for a gas of fermions or bosons and seeks the chemical potential. To do this, one must invert the number density integral; that is, given T and n, the temperature and number density, one must find the μ'/kT such that

the integral of the number density agrees with n. libstatmech does this by finding the root of the function

$$f = n_0 - n(T, \mu'/kT, X), \tag{2.15}$$

where n_0 is the given number density, n is the result of the integration [or, more generally, the function plus integral that gives the number density as computed from computeQuantity()], and X represents other parameters to the number density. The root is μ'/kT. To find this root, libstatmech uses the Brent solver in GSL. Iteration proceeds until the root achieves a relative tolerance of 10^{-12}. This tolerance is set by the parameter D_EPS_ROOT in Libstatmech.h. Our experience is that the chosen value works well.

2.5 Temperature Derivatives of Thermodynamic Quantities

Users may compute temperature derivatives of thermodynamic quantities with the computeTemperatureDerivatives() API routines. The temperature derivatives are computed with the GSL routine **gsl_deriv_central**. The step size for the differentiation is a fraction 0.001 of the temperature at which the derivative is desired. This fraction may be changed by setting D_STEP_FRACTION in Libstatmech.h to a different value.

2.6 Example: Screening

This example demonstrates how to add a user-defined potential to the integrand.

The Thomas-Fermi model is a quantum mechanical theory for the electronic structure of many-body systems. For simplicity, we will just apply a Yukawa potential to the electrons in an atom to describe the screening effect.

The Yukawa potential is a function of radius away from a charge:

$$\phi(r) = \frac{Ze}{r} e^{-r/r_0} \tag{2.16}$$

where Z is the atomic number, e is the proton charge, r_0 is a parameter to describe how the potential drops with radius. Because the potential contributes to the energy of an electron, the number density integral for electrons in the presence of the charge becomes a function of the radius (compare to

eq(2.6)):

$$n(r) = \frac{(mc^2)^3 g}{2\pi^2(\hbar c)^3\gamma^3}\int_0^\infty (x+\gamma)\sqrt{x^2+2\gamma x}\left[\frac{1}{1+\exp(x-\mu'/kT-e\phi(r)/kT)}\right]dx \tag{2.17}$$

Here the minus sign in the potential energy means that the interaction with positive charged nucleus lowers the electron's energy level.

To find the chemical potential, we consider that the nuclei are separated by a distance $2R$ and thus associate a sphere of radius R with each nucleus. Because of charge neutrality, we then have the constraint

$$\int_0^R n(r)e\,dr = Z. \tag{2.18}$$

We root find on this equation to get the chemical potential (μ'/kT in the integral).

We use $e^2 = \alpha\hbar c$ in the example, where α is the fine-structure constant, $\hbar$ is Planck's constant divided by 2π, and c is the speed of light in vacuum. We use the GNU Scientific Library values for these constants (in cgs units).

Figure 2.1 shows how the electron number density drops with radius under the Yukawa potential. Electrons tend to stay where the potential energy is lower. The curves with different r_0 shows that the electrons penetrate deeper with smaller r_0 (the percentage of electrons near the nucleus is higher).

2.7 Example: Bose-Einstein Condensate

This example describes an libstatmech application to boson systems.

2.7.1 Bose-Einstein Condensate

At low temperature or high number density, a boson gas tends to collapse into the lowest quantum state (ground state), which is a Bose-Einstein condensate. To calculate thermodynamic quantities in this situation, we need to add the ground state function to the integral and set the integral lower limit to the first excited state energy. These examples demonstrate how to add an user-supplied thermodynamic function and how to set a new integral lower limit.

For our system, we consider ideal bosons in a box with sides of equal length L and volume $V = L^3$. The single-particle states for the bosons are plane waves with momentum $\vec{p} = 2\pi\hbar\vec{n}/L$,

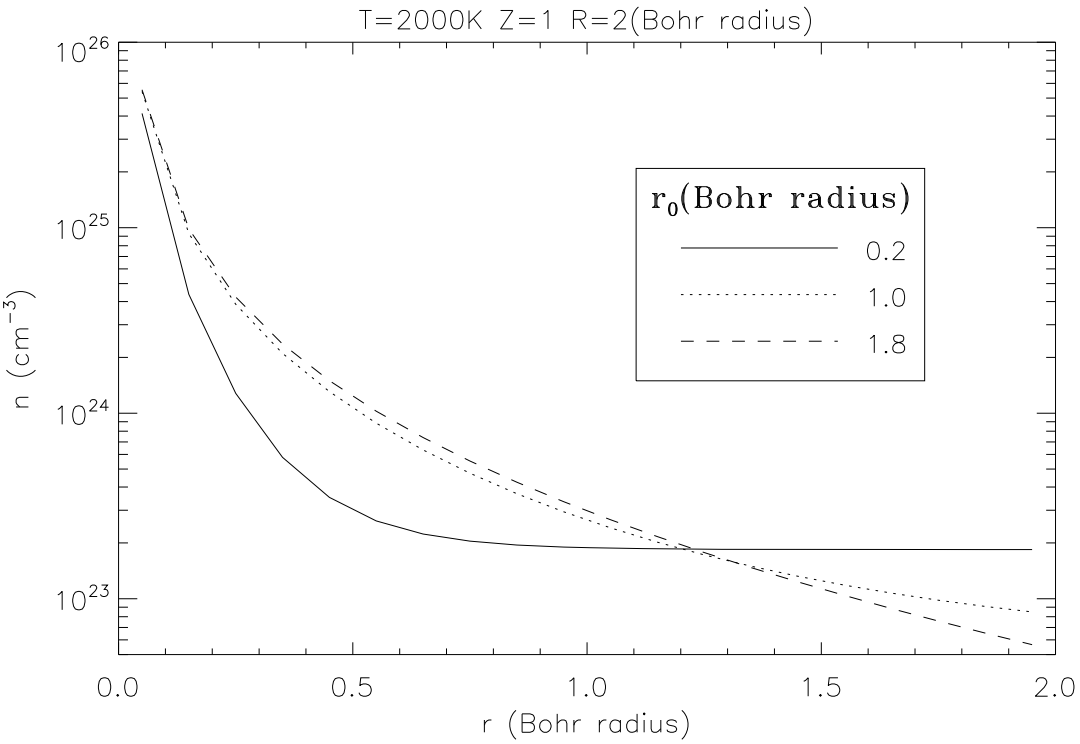

Figure 2.1: Number densities vs radius. For atomic number = 1 and atomic radius = 2 (Bohr radius), the electron densities get larger at smaller radius.

where $\vec{n}$ is a vector whose components n_x, n_y, and n_z are 0 or ±integers. The ground state thus has momentum zero, and we consider the energy to be that less the rest mass energy. We may derive the thermodynamic quantities from the grand canonical potential (see, for example, [12])

$$\Omega = k_B T \sum_r \ln\left(1 - e^{\beta(\mu' - \epsilon'_r)}\right), \tag{2.19}$$

where k_B is Boltzmann's constant, T is the temperature, $\beta = 1/k_B T$, μ' is the chemical potential less the rest mass, ϵ'_r is the energy of the single-particle state r less the rest mass, and the sum runs over all single-particle states r. Since we assume uniform density over volume V, the number density is given by

$$n = -\frac{1}{V}\frac{\partial \Omega}{\partial \mu'}. \tag{2.20}$$

By considering all single-particle states r such that $\epsilon_r = 0$, we thus find the ground-state number density:

$$n_0 = \frac{g}{V}\frac{1}{\exp(-\alpha) - 1} \tag{2.21}$$

where g is the boson particle's multiplicity and $\alpha = \mu'/kT$, which is always negative. The entropy density is given by

$$s = -\frac{1}{V}\frac{\partial \Omega}{\partial T}; \tag{2.22}$$

hence, the ground-state entropy density is

$$s_0 = -k_B \frac{g}{V}\left[\ln\left(1 - e^{\alpha}\right) + \frac{\alpha}{e^{-\alpha} - 1}\right]. \tag{2.23}$$

Finally, the pressure is

$$P = -\Omega/V; \tag{2.24}$$

hence, the ground state pressure is

$$P_0 = -\frac{k_B T}{V} g \ln\left(1 - e^{\alpha}\right). \tag{2.25}$$

Since the ground state single-particle energy is zero and the particles are considered to be non-interacting, the energy density of the ground state is zero.

The boson condensate examples in the libstatmech distribution include these functions,

which are then set by the API routine Libstatmech__Boson__updateQuantity(). Once the functions are set, they are applied during quantity calculations.

2.7.2 Integral Lower Limit

The non-condensate part of thermodynamic quantity calculations are still computed with the default integrands. Since the ground state quantities are now calculated separately, however, we need to set the integral lower limit to the first excited state energy over $k_B T$. We do this by considering the particle in this state to be non-relativistic:

$$E_1 = \frac{p^2}{2m} = \frac{\hbar^2 (2\pi/\lambda)^2}{2m} = \frac{2(\hbar\pi)^2}{mV^{2/3}} \tag{2.26}$$

$$x_1 = E_1/kT = \frac{2(\hbar\pi)^2}{mV^{2/3}kT} \tag{2.27}$$

The new integral lower limit is set in the examples with the API routine Libstatmech__Boson__updateIntegralLowerLimit().

2.7.3 Results

We may use the boson example codes in the libstatmech distribution to explore Bose-Einstein condensation in some detail. The examples use both the ground-state boson functions defined above and the default integrands with the lower integral limit x_1. For example, with the function and the integral together the number density is:

$$n = \frac{g}{V}\frac{1}{\exp(-\alpha) - 1} + \frac{(kT)^3 g}{2\pi^2(\hbar c)^3}\int_{x_1}^{\infty} \frac{(x+\gamma)\sqrt{x^2 + 2\gamma x}}{\exp(x-\alpha) - 1}\, dx \tag{2.28}$$

When α (the chemical potential less rest mass over kT) is a large negative number, the function term is small compared to the integral and is negligible. When α is increasing toward zero, the function term starts to dominate. At this point almost all the particles will collapse into the ground state. This happens at low temperatures or high number densities. Figures 2.2 and 2.3 below show results from the distribution examples for a system volume $V = 1$ cm for a boson with multiplicity $g = 3$ and a rest mass of $mc^2 = 1$ MeV. Figure 2.2 shows the ratio of the number of particles N_0 in the ground state relative to the total number of particles N as a function of the total

number density of particles in the box of volume V. Since the temperature and volume are fixed, it is clear that adding particles to the system eventually causes condensation in which most of the particles are in the ground state. The number density at which condensation occurs increases for higher temperature because the probability to excite a given boson to the first excited state is higher for higher T.

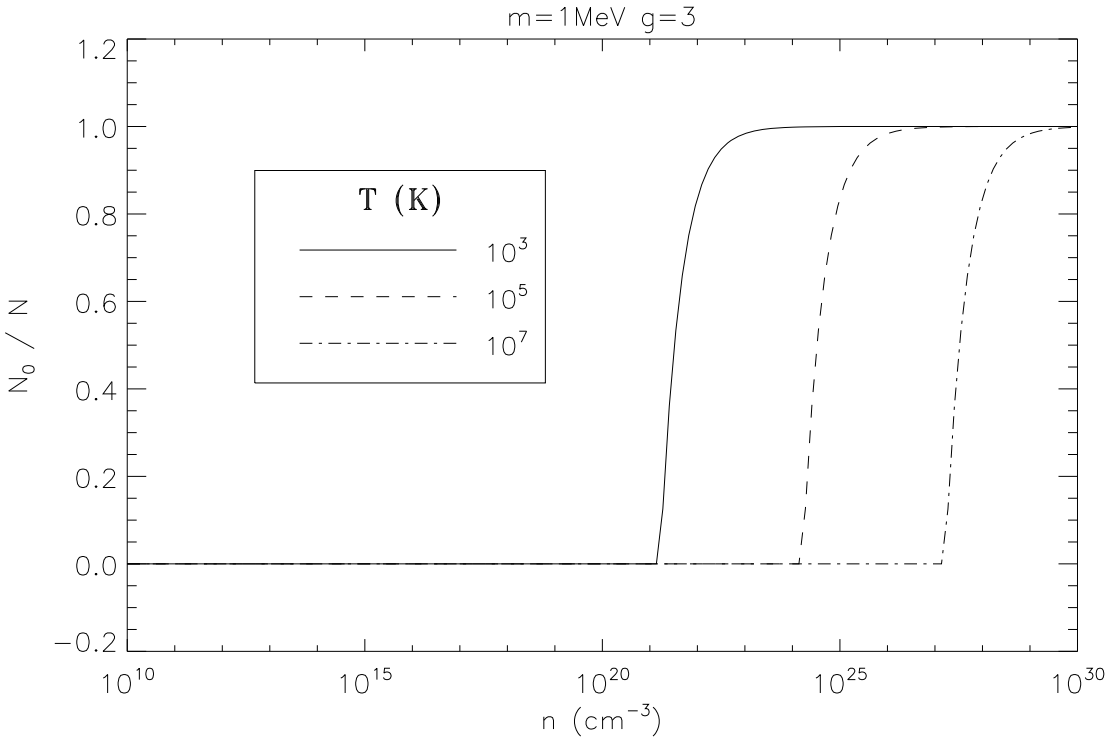

Figure 2.2: Ground state fraction vs number density at different temperatures. N_0 is the number of particles in the ground state, while N is the total particle number.

Condensation also occurs when the temperature is lowered for fixed number density. Figure 2.3 shows this for several number densities. As the temperature is lowered, a temperature is reached at which the fraction of particles in the ground state rises quickly. This is the phase transition to the condensate.

The critical temperature T_c at which the phase transition occurs may be computed for non-relativistic bosons to be

$$T_c = \frac{2\pi\hbar^2}{mk_B}\left(\frac{n}{g\zeta(3/2}\right)^{2/3}, \tag{2.29}$$

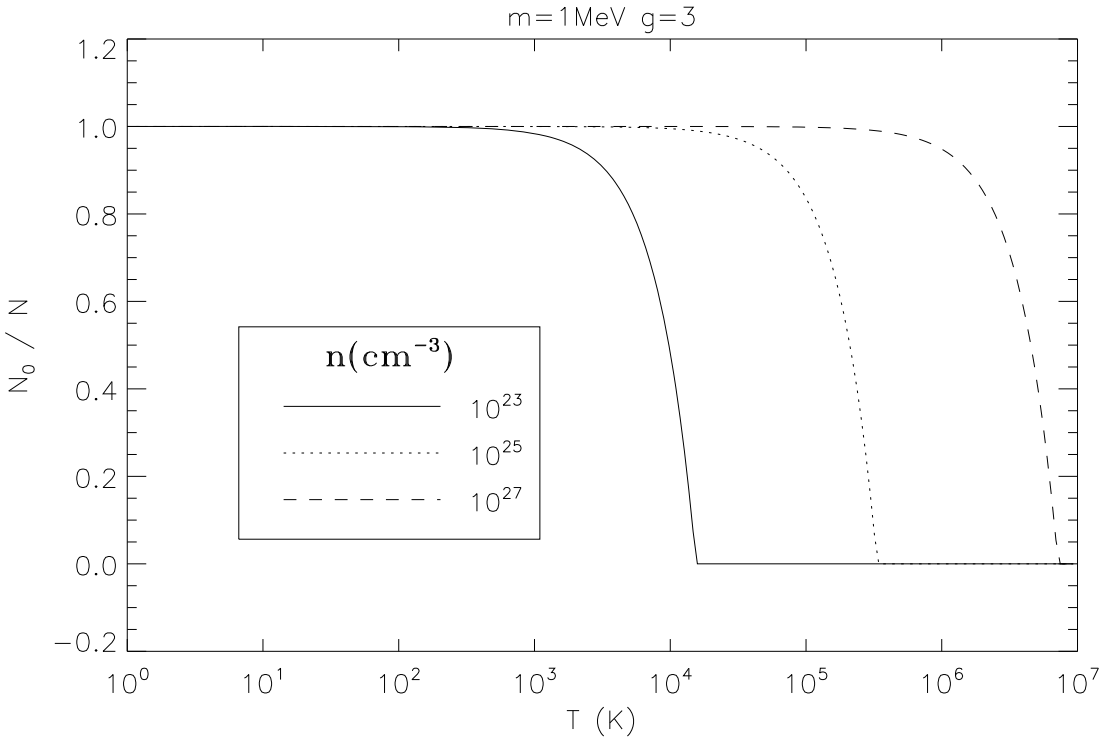

Figure 2.3: Ground state fraction vs temperature at different number densities. N_0 is the number of particles in the ground state, while N is the total particle number.

where $\zeta(3/2)$ is the Riemann zeta function of argument 3/2 [12]. Figure 2.4 shows N_0/N as a function of T/T_c as computed with a libstatmech example code. It is clear that condensation does indeed occur at T_c. Figure 2.5 shows the specific heat capacity as a function of T/T_c as computed from a libstatmech example code with $g = 3$ and $mc^2 = 100$ MeV. The cusp in the curve at $T = T_c$ is the signal of the phase transition. As T increases above T_c, the specific heat capacity settles down towards $3k_B/2$, as expected for a non-relativistic, ideal gas.

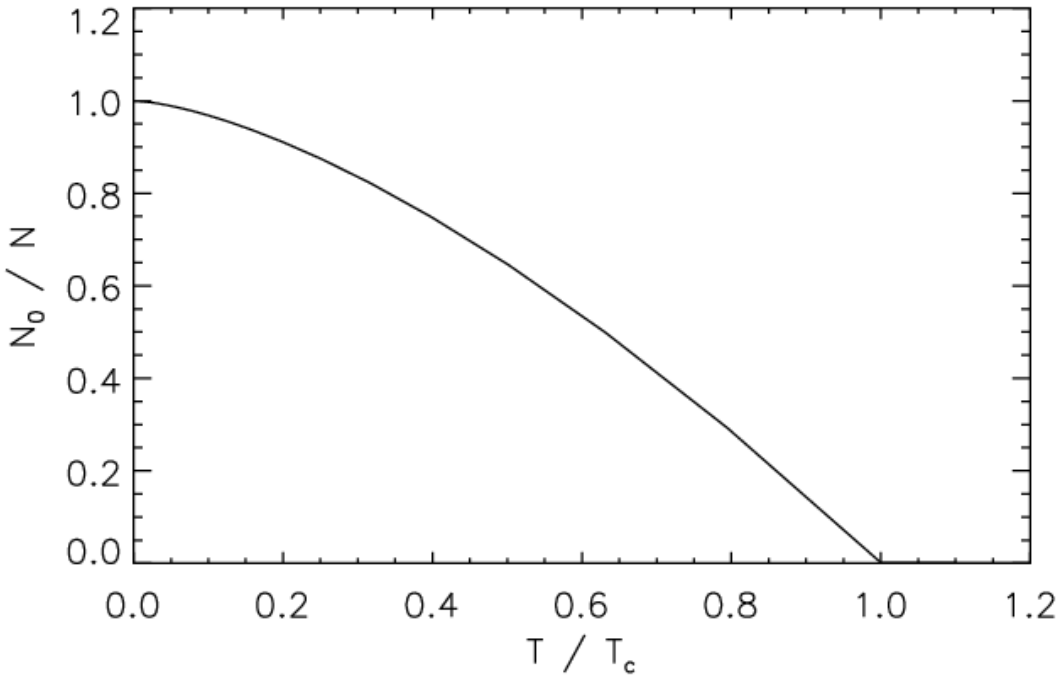

Figure 2.4: Ground state fraction as a function of the temperature relative to the critical temperature for an ideal boson gas with $mc^2 = 100$ MeV and $g = 3$.

Figure 2.6 shows the pressure of an ideal boson gas with $mc^2 = 100$ MeV and $g = 3$ as a function of the number density for a fixed temperature of 100 K. Below the condensation, the pressure is proportional to the number density, as expected for an ideal, non-relativistic gas. Once the number density exceeds the critical number density, the pressure becomes constant as a function of number density. This surprising result is due to the fact that, as the number of particles increases towards infinity, the number of particles not in the ground state becomes a constant. To understand this, consider a two-state system with a probability p for a single particle not to be in the ground

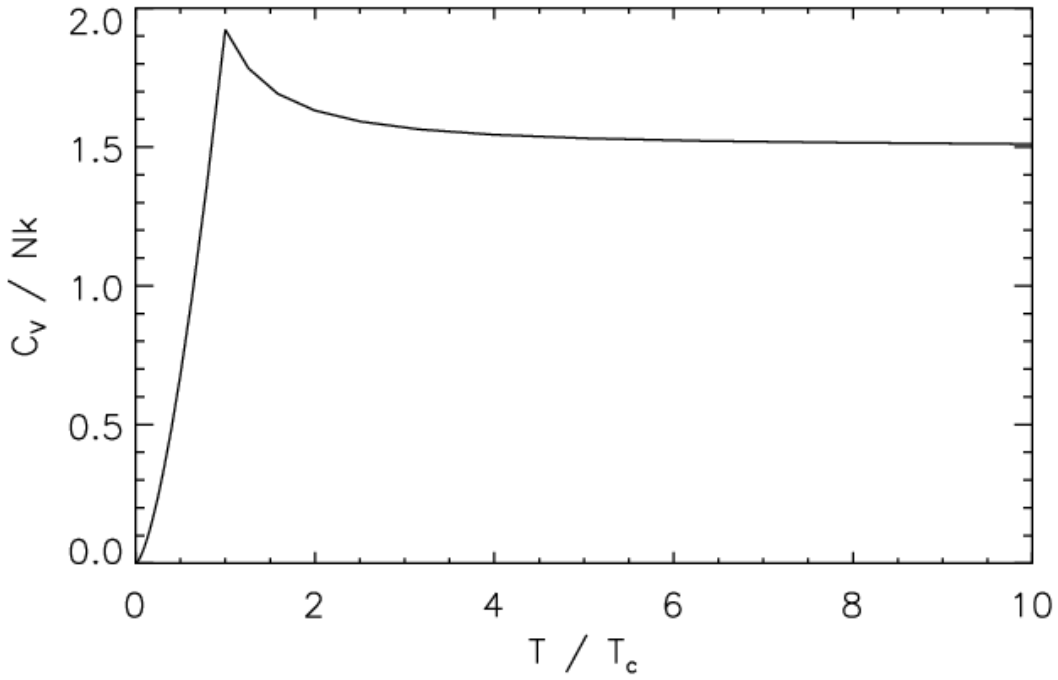

Figure 2.5: Specific heat capacity as a function of temperature for the ideal boson gas.

state. The particles are indistinguishable; therefore, the probability to have m out of a total of N particles not in the ground state is $P(m) = \mathcal{N}p^m$, where $\mathcal{N}$ is a normalization constant. This means that

$$\mathcal{N}\sum_{m=0}^{N} p^m = 1. \tag{2.30}$$

If $N \to \infty$, then $\mathcal{N} = 1 - p$. The total number of particles not in the ground state for large N is thus

$$\sum_{m>0}(1-p)mp^m = (1-p)p\frac{d}{dp}\sum_{m=0} p^m = (1-p)p/(1-p)^2 = p/(1-p). \tag{2.31}$$

This becomes a negligible fraction of N as $N \to \infty$. Nevertheless, it is these particles that carry the energy, pressure, and entropy; hence, these quantities become constant as a function of the number density when the condensation occurs.

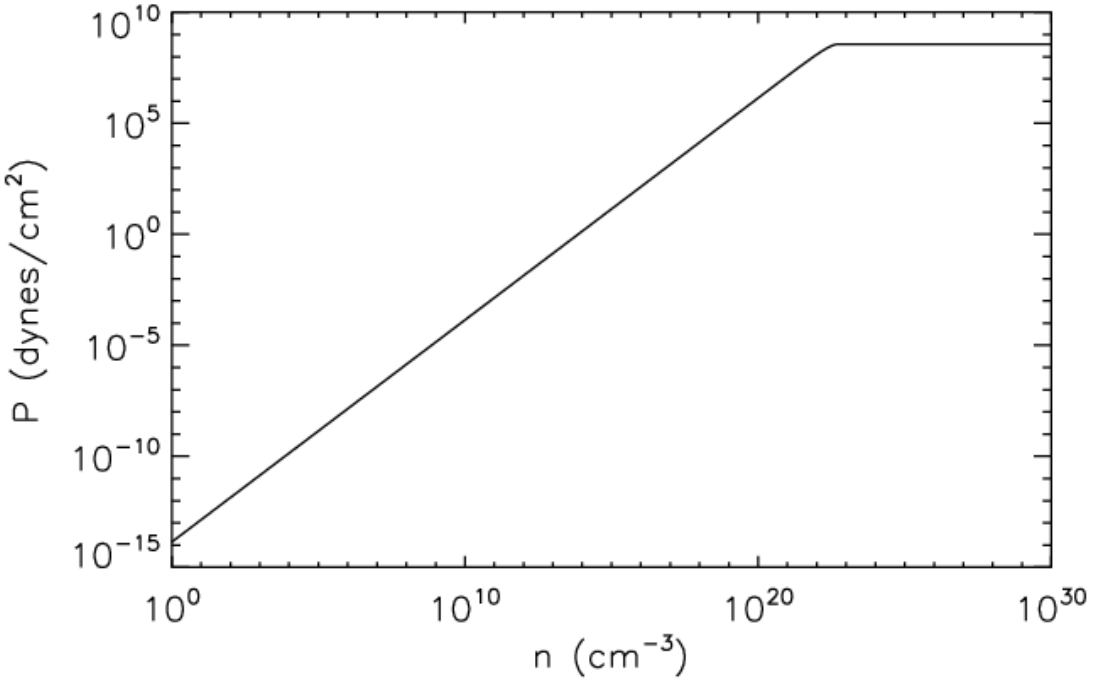

Figure 2.6: Pressure as a function of number density for an ideal gas of bosons with mass $mc^2 = 100$ MeV and $g = 3$ at $T = 100$ K. Once condensation occurs, the pressure is independent of the number density.

Chapter 3

Weak Statistical Equilibrium

For a typical weak nuclear reaction β-decay $i \rightarrow j + e^- + \bar{\nu_e}$ we may get the forward reaction rate $\lambda_{forward}$ from laboratory or theoretical nuclear physics models. The reverse rates $\lambda_{reverse}$ (e.g. electron capture rate $e^- + j \rightarrow i + \nu_e$) are generally endothermic in the laboratory and have to be calculated from nuclear models as well (e.g., [5]). While the models are in general quite good, there is no guarantee that the forward and reverse rates are related in such a way that a nuclear reaction network will evolve, given sufficient time, into weak statistical equilibrium (WSE). An equilibrium means that after a sufficient time of evolution, the abundances of all species in the system present get their stable values which depend only on temperature, density, electron abundance and nuclear properties of each species (mass, multiplicity).

With libstatmech and libnuceq (see Appendix D), we can compute reverse weak rates on nuclei from detailed balance, as we do with the strong and electromagnetic rates. This permits us to evolve a reaction network dynamically into the full weak equilibrium. To demonstrate how to apply detailed balance, let's consider the β-decay above. At some certain temerature and density if the two species i and j are in weak statistical equilibrium, we should have

$$\lambda_{forward} Y_i^{WSE} = \lambda_{reverse} Y_i^{WSE} \tag{3.1}$$

In the equation the superscript WSE stands for the value in weak statistical equilibrium, and Y_i is

defined as number of species i per nucleon:

$$Y_i \equiv \frac{N_i}{N_{nucleon}} \tag{3.2}$$

which is refered to "abundance" and is a useful quantity in such studies. So eq(3.1) can be interpreted as the consumption rate of species i equals the production rate at equilibrium. And if we can calculate both Y^{WSE}, the reverse rate then can be computed from the forward one.

To get Y_i from other quantities, we start from the chemical potential of classical particles (nuclei can be considered as classical particles as the temperature won't get up to nuclei mass level, $T_9 \sim 1, kT \sim 0.1\text{MeV}$):

$$\mu_i = m_i c^2 + kT \ln\left(\frac{Y_i}{Y_{Qi}}\right) \tag{3.3}$$

where Y_{Qi} is the quantum abundance per nucleon and is given by

$$Y_{Qi} = G_i \frac{1/\lambda^3_{thermal,i}}{n_{nucleon}} = \frac{G_i}{\rho N_A}\left(\frac{m_i kT}{2\pi\hbar^2}\right)^{3/2} \tag{3.4}$$

In the equation above, $n_{nucleon} = \rho N_A$ is nucleon number density (i.e. baryon number density). $\lambda_{thermal}$ denotes the thermal de Broglie wavelength of a free ideal gas, and $1/\lambda^3_{thermal,i}$ is like number density of species i when the average interparticle spacing is the thermal de Broglie wavelength. So the quantum abundance Y_{Qi} is like the number of species i per nucleon when quantum effects start to become important, and it is just a function of temperature, density, mass and multiplicity of species i. Comparing Y_i and Y_{Qi} helps us to tell whether the gas is ideal or not.

For WSE condition we Y_i^{WSE} for eq(3.3):

$$Y_i^{WSE} = Y_{Qi} e^{\frac{\mu_i^{WSE} - m_i c^2}{kT}} \tag{3.5}$$

With the result derived in appendix D

$$\mu_i^{WSE} = A_i \mu_n^{WSE} + Z_i \mu_e^{WSE} \tag{3.6}$$

and the fact $A_i = A_j$ we have

$$\frac{\lambda_{reverse}}{\lambda_{forward}} = \frac{Y_i^{WSE}}{Y_j^{WSE}} = \frac{Y_{Qi}}{Y_{Qj}} e^{(Z_j - Z_i)\mu_e^{WSE}/kT} e^{-Q/kT} \tag{3.7}$$

where $Q = m_i c^2 - m_j c^2$ is just the Q-value (mass difference) of the reaction. With libstatmech we can easily compute the chemical potential of electron at weak statistical equilibrium and thereby compute reverse rates from forward ones.

A crucial point concerns how the weak rates are computed. These are typically given as two-dimensional tables [4, 8] in terms of $T_9 = T/10^9$ K and ρY_e, where ρ is the mass density in (g/cc) and Y_e is the electron fraction

$$Y_e = \sum_{i,nuc} Z_i Y_i. \tag{3.8}$$

To compute the forward rate, one supplies T_9 and ρYe and interpolates the rate. To compute the reverse rate, we need the WSE at the same T and ρY_e instead of at a particular T_9 and ρ, as is normally done. Once the proper WSE is computed, we can compute the reverse rate from the forward one.

The importance of using detailed balance can be seen in Figure 3.1. Here we evolved a nuclear reaction network at fixed temperature and density using rates from the tables in [4]. The solid curve shows the evolution of Y_e when reverse rates are computed from detailed balance while the dashed curve shows what happens if both forward and reverse rates are taken from the tables. The calculation with detailed balance evolves to the correct WSE value for Y_e while the calculation using the tables for both forward and reverse rates does not. This indicates a slight imbalance in the tables between the forward and reverse rates. That such an imbalance is present is not surprising. The rates are computed from incomplete nuclear data. Furthermore, the beta-decay rates, for example, must properly account for Pauli blocking of the final electron state. This is difficult to compute accurately.

An example to demonstrate the role of WSE in network calculation is given here. Let's put pure ^{12}C in a box with initial temperature of 6×10^9K and let it evolve. At such a high temperature carbon burning will happen and the system will evolve to equilibrium among all species.

Figure 3.2 shows that the number of heavy nuclei drops, i.e. more heavier nuclei were formed to reduce the total number. Y_h is big at the beginning that means it tends to have fewer heavy

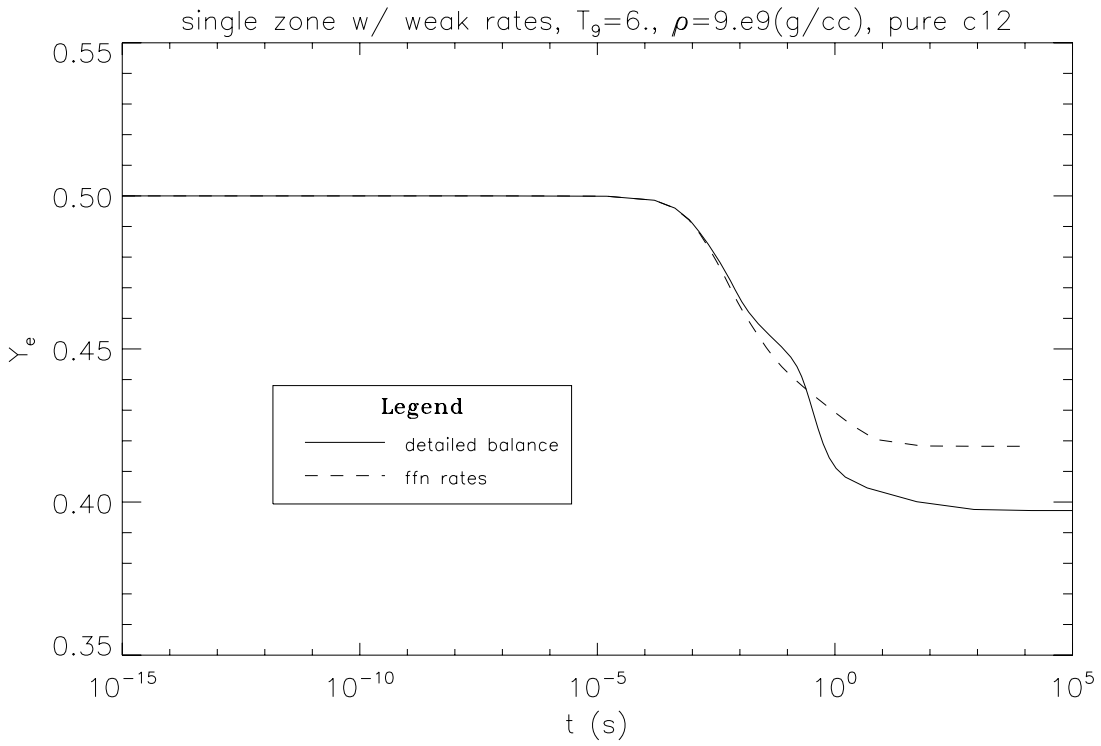

Figure 3.1: Evolution of a nuclear reaction network starting with pure ^{12}C at fixed $T_9 = T/10^9$ K $= 6$ and mass density $\rho = 9 \times 10^9$ g/cc with weak interaction rates on nuclei from the tables in [4]. The solid curve shows the evolution of Y_e when reverse rates are computed from detailed balance while the dashed curve shows the evolution when reverse rates are simply taken from the tables. The network with the reverse rates computed from detailed balance evolves to the correct WSE while the one with both forward and reverse rates taken from the tables does not.

nuclei. And it gets to zero at the end meaning the system is in equilibrium then.

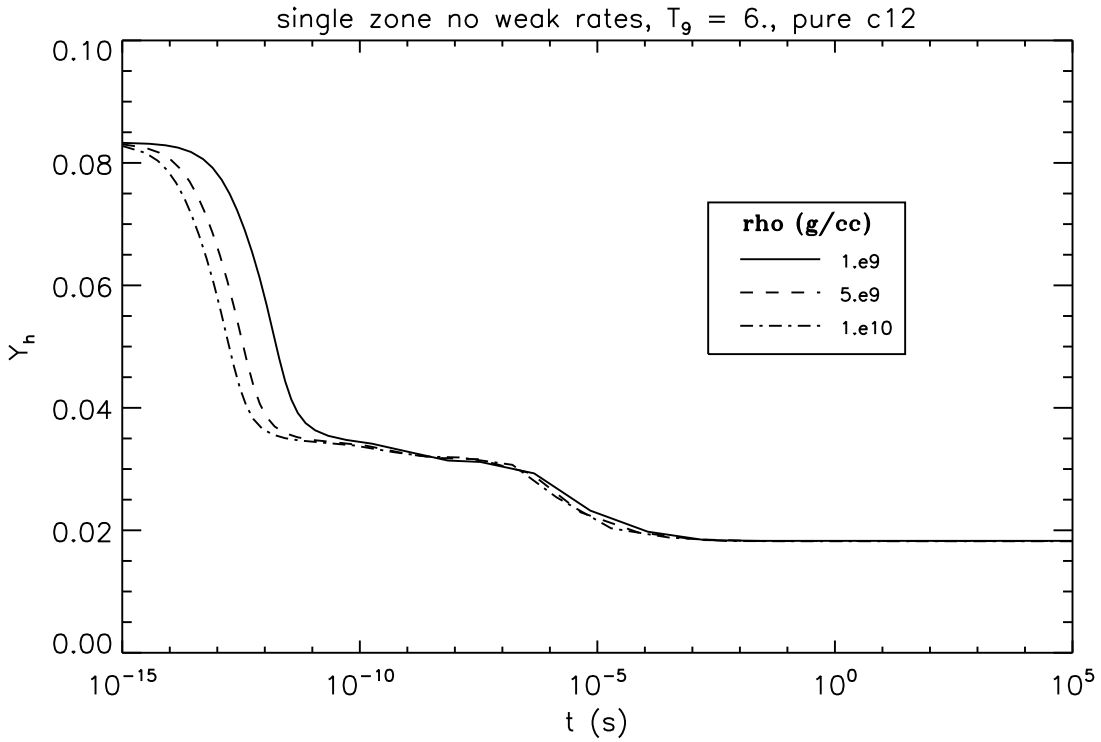

Figure 3.2: Abundance of heavy nuclei (Y_h) vs time for a single zone evolution starting with pure ^{12}C and without weak interaction (Y_e not changing). Y_h drops from 0.833 (1/12 for ^{12}C) to 0.0179 (1/56 for ^{56}Ni) to make more heavier species, i.e. to reduce the number of heavy nuclei (atomic number is greater or equal to 6). ^{56}Ni dominates the final abundances since it is the most stable species with $Y_e = 0.5$.

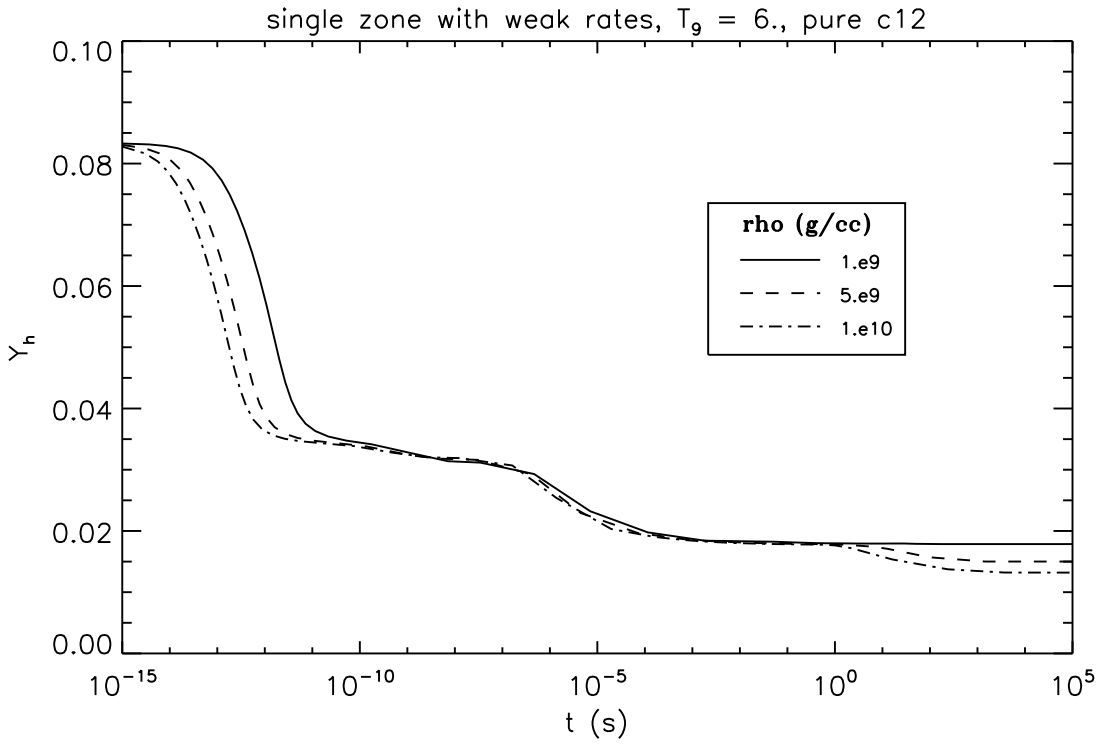

Figure 3.3: Abundance of heavy nuclei (Y_h) vs time for a single zone evolution starting with pure ^{12}C and with weak interaction (Y_e can change). Comparing with Figure(3.2), it is shown that at high density Y_h can drop even more to make heavier species (e.g. ^{70}Ni). That's because at high density electron capture rates get higher and thus make more neutron-rich isotopes.

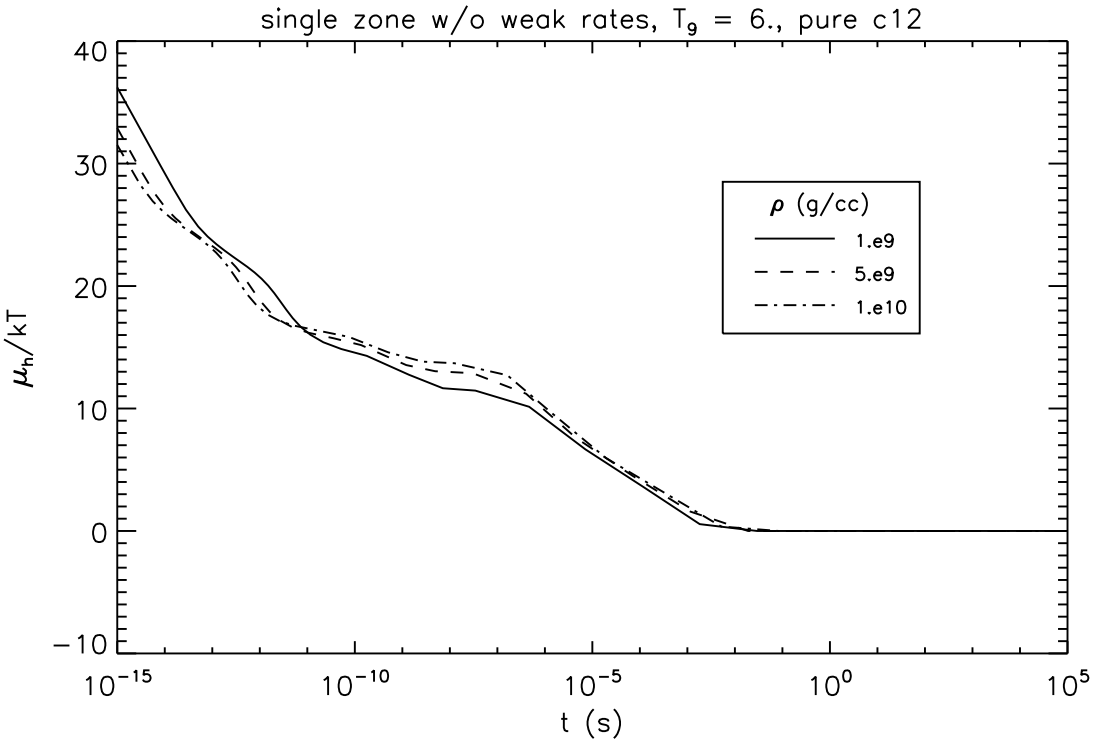

Figure 3.4: Chemical potential of heavy nuclei vs time of evolution of a single zone without weak interaction starting with pure ^{12}C. It starts at a high value, meaning that the system has too many heavy nuclei and tends to combine some together to reduce the total number of heavy nuclei. After this procedure, μ_h gets to zero, indicating that the system is in equilibrium.

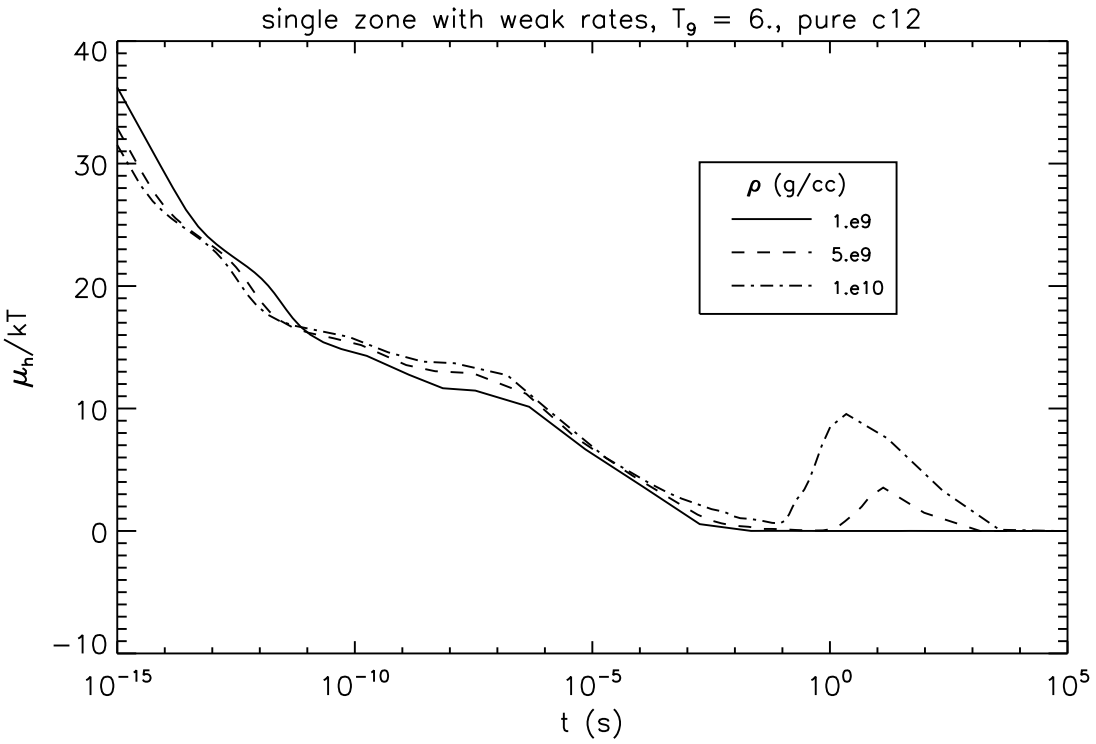

Figure 3.5: Chemical potential of heavy nuclei vs time of evolution of a single zone with weak interaction. The interesting behavior is that at high density, μ_h gets near to zero first and bump up away from equilibrium. At the end it comes back to zero to achieve equilibrium. Details of this process will be explored in the following figures.

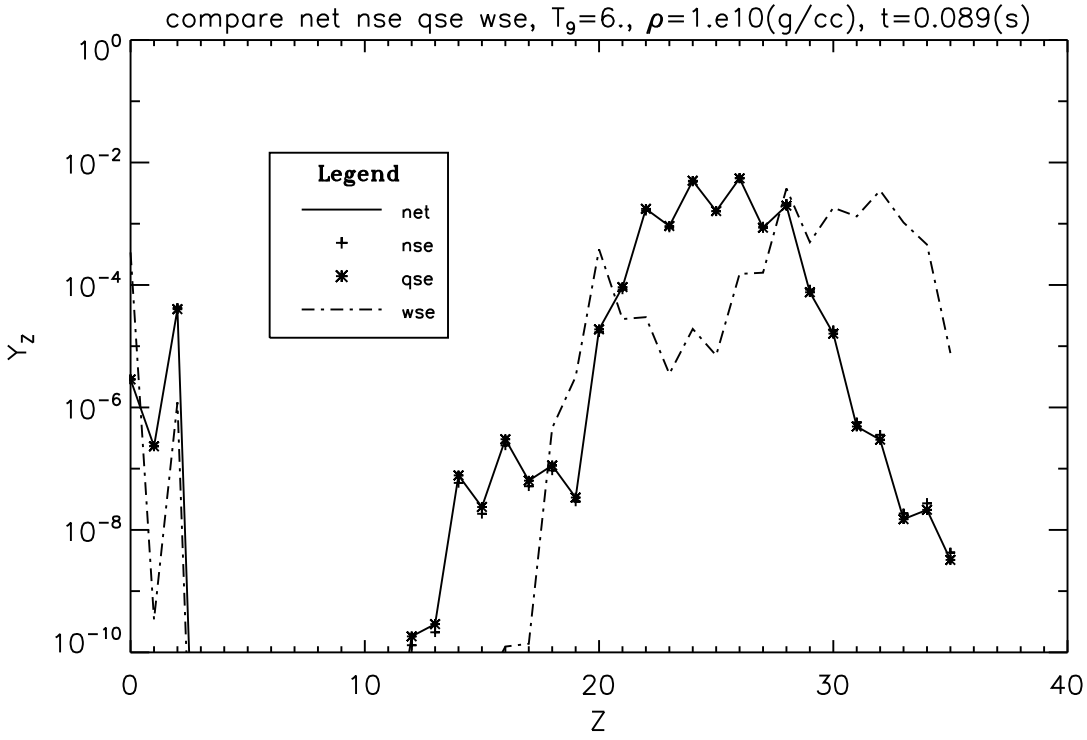

Figure 3.6: Y_Z vs Z at t=0.089s of the evolution in Figure(3.5). It is almost right before the bump. We can see that the network is in QSE and already NSE, but far from WSE meaning that the weak interactions have not started much yet.

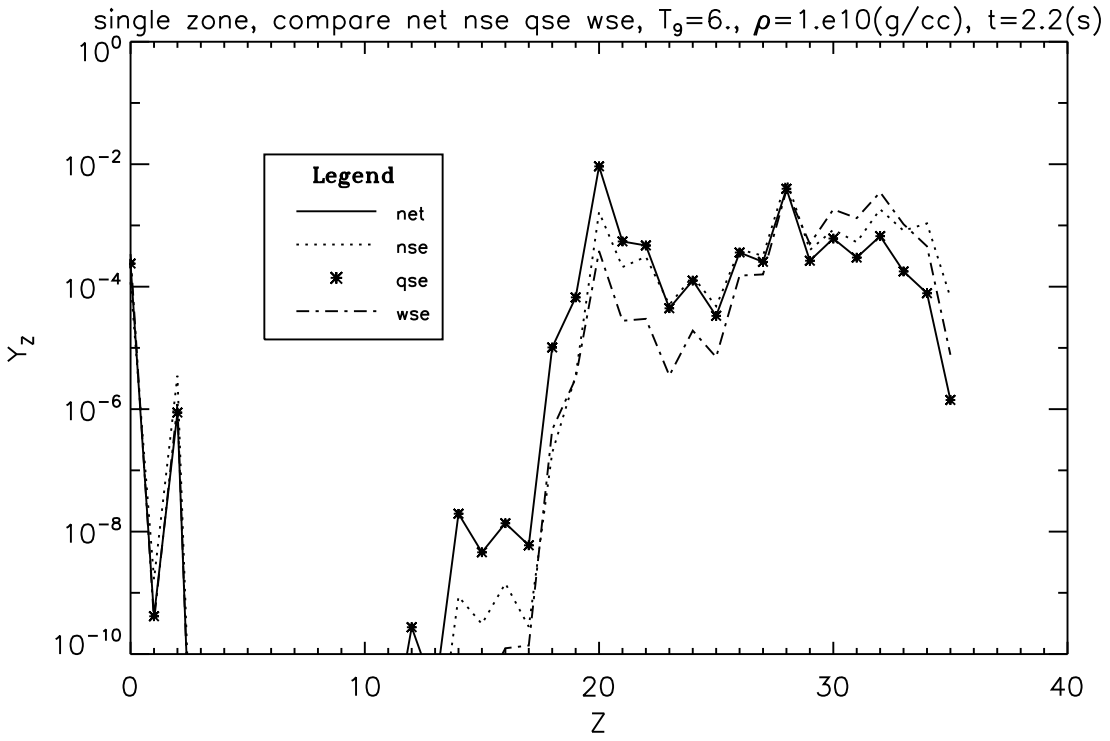

Figure 3.7: Y_Z vs Z at the peak of the bump. The net work is still in QSE, but starts to move away from NSE and move closer to WSE. In this stage, weak reactions start to make a significant contribution. The electron abundance is reduced and thus the nuclear equilibrium needs to change its distribution and that needs time, which has comparable time scale with weak reactions. So the network evolves to NSE and WSE at the same time.

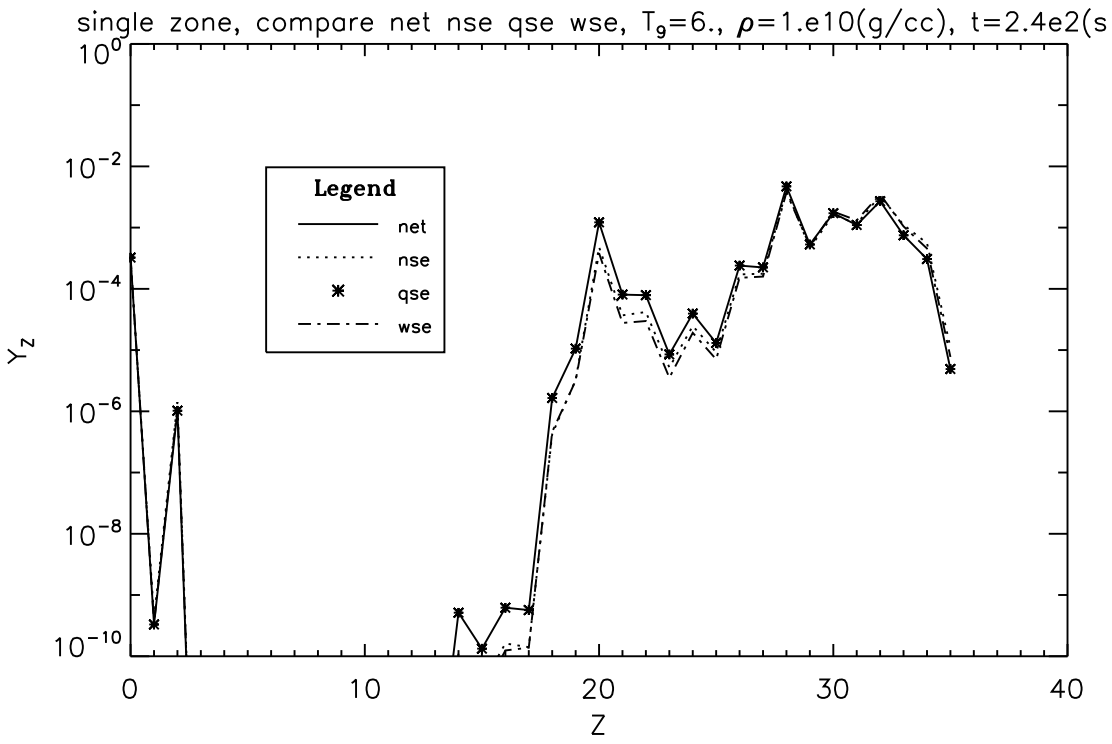

Figure 3.8: Y_Z vs Z dwon the hill. Both NSE and WSE get closer to network abundances.

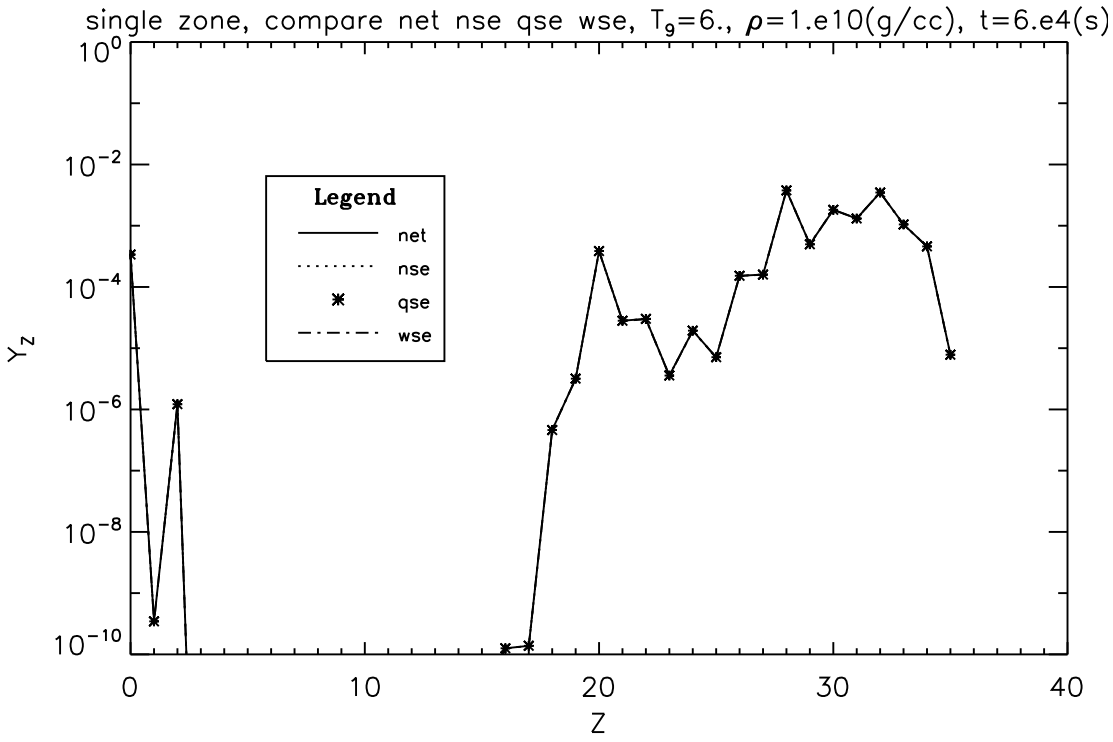

Figure 3.9: Y_Z vs Z at the end. The network evolves to all the equilibria. The time scale of this process is about 10^4 seconds. Although the network is under a competition of NSE and WSE, it seems to always fit QSE.

These figures show that during the evolution to equilibrium, the network experiences a time that the weak reactions start to become important and nuclear equilibrium does not have enough time to respond to the change of Y_e. This is really an interesting phonomenon.

Chapter 4

Simple SNIa Model

In this chapter I will develop a simple Type Ia supernovae model and apply libstatmech to calculate the reverse weak reaction rates during the explosion.

4.1 Hydrodynamics

The simple SNIa model is a 1-zone, 0-D model. We let the white dwarf (to be exploded) be a uniform, isotropic sphere with radius R. Thus the temperature and density do not depend on the radius of the white dwarf.

Start from the momentum equation:

$$\frac{dv}{dt} = -\frac{1}{\rho}\frac{dP}{dr} - \frac{GM(r)}{r^2} \tag{4.1}$$

where v is the velocity at radius r and $M(r)$ is the mass inside radius r.

For simplification, we assume that the pressure gradient $\frac{dP}{dr}$ does not change much from the center to the surface. So

$$\frac{dP}{dr} \approx \frac{P_{surface} - P_{center}}{R} \approx -\frac{P}{R} \tag{4.2}$$

where we assume the pressure at the surface $P_{surface}$ is 0 and we will use the pressure at the center P_{center} as a the pressure of whole star. Although it seems a contradiction that a uniform pressure star has pressure gradient, it is OK for a 1-zone model which could be expanded into multi-zone in the future. In fact, even a 1-zone model the error is just with a factor of order 1.

For the initial conditions, let's make the acceleration small ($dv/dt \approx 0$) at the beginning. Then from eq(4.1) and eq(4.2) we have the relationship:

$$\frac{GM(R)}{R_0^2} = -\frac{1}{\rho_0}\frac{dP}{dr} = \frac{1}{\rho_0}\frac{P_0}{R_0} \tag{4.3}$$

where the subscript 0 stands for the initial values ($t = 0$). Then also for the surface, plug $r = R$ and eq(4.3) into eq(4.1), leading the acceleration to

$$\frac{dv}{dr} = \frac{R^3}{\rho_0 R_0^3}\frac{P}{R} - \frac{P_0 R_0}{\rho_0 R^2} \tag{4.4}$$

where $\rho = \rho_0 \frac{R_0^3}{R^3}$ was applied. For easier calculation later we define

$$x \equiv \frac{R}{R_0} \tag{4.5}$$

and

$$y \equiv \frac{v}{R_0} \tag{4.6}$$

Then we get the two first order ordinary differential equations for hydrodynamics:

$$\frac{dx}{dt} = y \tag{4.7}$$

$$\frac{dy}{dt} = \frac{P_0}{\rho_0 R_0^2}\left[\frac{P}{P_0}x^2 - \frac{1}{x^2}\right] \tag{4.8}$$

4.2 Thermodynamics

Now for thermodynamic part, we start from the differential form of free energy $f(T, v, \{Y_i\})$ and entropy per nucleon $s(T, v, \{Y_i\})$:

$$df = -sdT - Pdv + \sum_i \mu_i dY_i \tag{4.9}$$

$$ds = \frac{\partial s}{\partial T}dT + \frac{\partial s}{\partial v}dv + \sum_i \frac{\partial s}{\partial Y_i}dY_i \tag{4.10}$$

Here the extensive properties f, s, v(volume per nucleon, inverse of nucleon number density ρN_A) are normalized to per nucleon. The sum is over all species (including electron) in the system.

Multiply T to eq(4.10) and apply the Maxwell relations below,

$$\frac{\partial^2 f}{\partial v \partial T} = -\left(\frac{\partial s}{\partial v}\right)_{T,\{Y_i\}} = -\left(\frac{\partial P}{\partial T}\right)_{v,\{Y_i\}} \tag{4.11}$$

$$\frac{\partial^2 f}{\partial Y_i \partial T} = -\left(\frac{\partial s}{\partial Y_i}\right)_{v,T} = \left(\frac{\partial \mu_i}{\partial T}\right)_{v,\{Y_i\}} \tag{4.12}$$

we have

$$Tds = T\frac{\partial P}{\partial T}dv - \sum_i T\frac{\partial \mu_i}{\partial T}dY_i + c_V dT \tag{4.13}$$

where

$$c_V = T\frac{\partial s}{\partial T} \tag{4.14}$$

is heat capacity per nucleon. Then we take time derivative of eq(4.13) and apply $Q = Tds + \sum_i \mu_i dY_i$ and $dv = d(1/n_{nucleon}) = d\left(\frac{1}{\rho N_A}\right) = \frac{2x^2}{\rho_0 N_A}dx$, leading to

$$\frac{dT}{dt} = \frac{1}{c_V}\left[\frac{dQ}{dt} - \frac{3x^2 T}{\rho_0 N_A}\left(\frac{\partial P}{\partial T}\right)y + \sum_i \left(T\frac{\mu_i}{\partial T} - \mu_i\right)\frac{dY_i}{dt}\right] \tag{4.15}$$

This is the first order ordinary differential equation for temperature, together with eq(4.7) and eq(4.8) forming a full series of equations that determine the behavior of the white dwarf.

There are three terms in the bracket on the rhs of eq(4.15) that determine how temperature changes. The first term is the heat loss mostly by neutrinos which we do not take into account in this simple model. The second term is basically the pressure work (e.g. for classical particles $T\frac{\partial P}{\partial T} = P$). The third term is the energy generation part, which will be discussed in detail later in §4.4.

4.3 Oscillation Without Energy Generation

Now let's consider first a white dwarf without energy generation. We can imagine that the white dwarf may oscillate to some amplitude around its equilibrium point. Until we turn on the energy generation (e.g. carbon burning) the oscillation would be stable. We can even simplify the

problem here with a polytrope model (see appendix E) that let

$$P = P_0 x^{-5} \tag{4.16}$$

then we will have a simple-harmonic-like oscillation with frequency $\omega = \sqrt{\frac{P_0}{\rho_0 R_0^2}}$.

Figure(4.1) shows the oscillation of a polytrope white dwarf model that consist of pure ^{12}C. In the figure, $T9_0$ indicates the initial temperature in 10^9K, which is 1 billion Kelvin. Initial density is $10^8 g/cm^3$. Initial radius is $10^8 cm$. And we give the white dwarf a small inward kick, i.e. initial velocity (y) is -0.02. And it behaves well with the predicted oscillating time period $\sim 1.32s$.

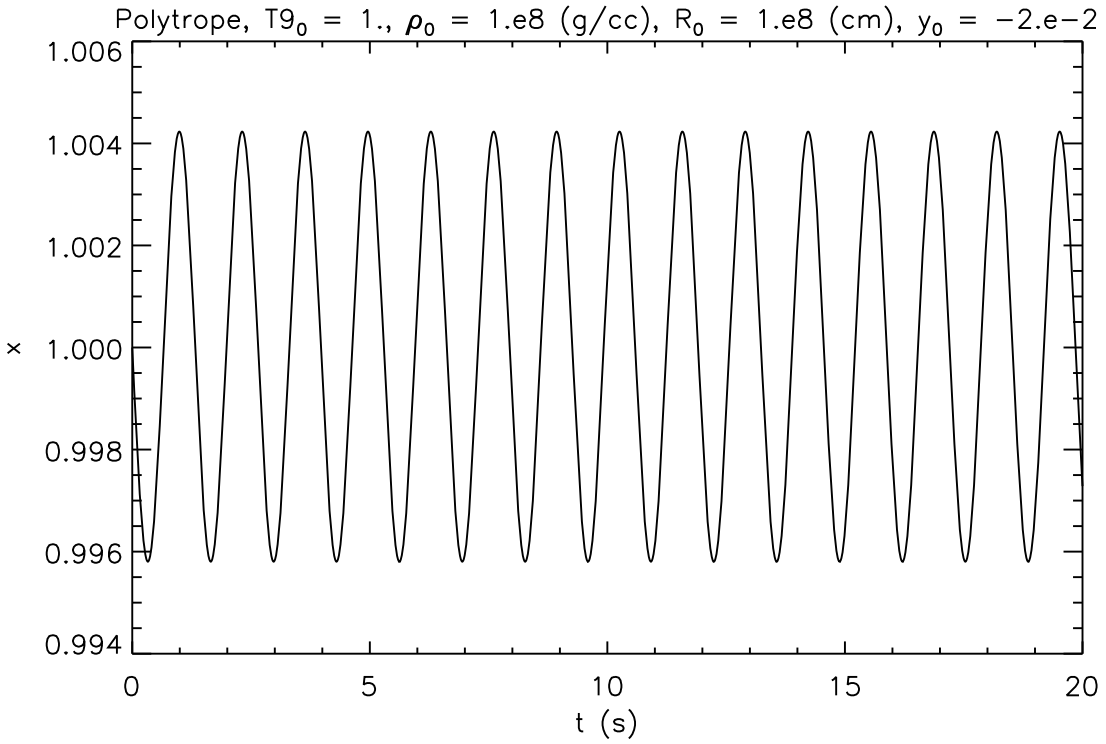

Figure 4.1: x vs time of polytrope model with pure ^{12}C. The star behaves like a stable simple harmonic oscillation with frequency $\omega = \sqrt{P_0/(\rho_0 R_0^2)} \approx 4.75(rad/s)$, ($P_0$ is computed to be $2.254e25(dynes/cm^2)$), corresponding to period of $\sim 1.32(s)$, which is about right.

Figure(4.2) shows the oscillation with a real pressure calculated by libstatmech, but without energy generation. The white dwarf oscillates well except that the frequency seems small compared to the polytrope one. That's because a polytrope is not quite accurate.

A white dwarf is supported by mainly the degenerate pressure of electrons, which only depends on the its number density. When we turn the carbon burning on (start energy generation), the temperature will increase, however the pressure won't respond much and the white dwarf won't expand and cool down back. Thus the heat will accumulate until the temperature gets high enough to ignite intense carbon burning (the reaction rate of carbon burning is quite sensitive to temperature, $\sim T^{40}$), which explodes the whole star. And we will see the scenario in the following section.

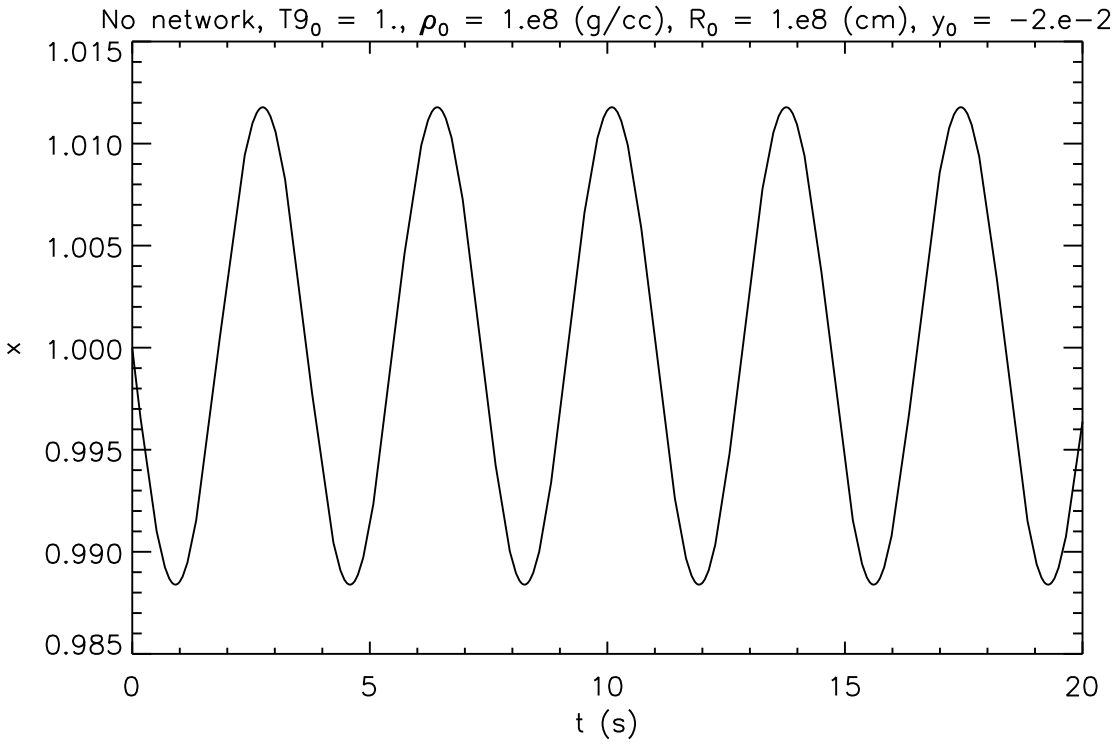

Figure 4.2: x vs time of simple SNIa model with pure ^{12}C. The calculation here is not running the network, i.e. no energy generation. The time period is longer than that of Figure(4.1).

4.4 Energy Generation

Now we are going to add energy generation to the SNIa model.

Let's look at the energy generation part in eq(4.15).

$$\epsilon = \sum_i \left(T\frac{\mu_i}{\partial T} - \mu_i\right)\frac{dY_i}{dt} = \sum_{i\in nuc}\left(T\frac{\mu_i}{\partial T} - \mu_i\right)\frac{dY_i}{dt} + \left(T\frac{\mu_e}{\partial T} - \mu_e\right)\frac{dY_e}{dt} \tag{4.17}$$

From eq(3.3) we can write the temperature derivative of the species i's chemical potential as

$$\frac{\partial \mu_i}{\partial T} = k \ln\left(\frac{Y_i}{Y_{Qi}}\right) - kT\frac{\partial \ln Y_{Qi}}{\partial T} = \frac{\mu_i - m_i c^2}{T} - \frac{3}{2}k - kT\frac{\partial \ln G_i}{\partial T} \tag{4.18}$$

Then the nuclear part in eq(4.17) then becomes

$$\epsilon_{nuclear} = \sum_{i \in nuc}\left(-m_i c^2 - \frac{3}{2}kT - kT^2\frac{\partial \ln G_i}{\partial T}\right)\frac{dY_i}{dt} \tag{4.19}$$

So the energy changes in baryon include mass change, kinetic energy change and nuclear energy change (see appendix).

Let's write mass in the form of mass excess Δ (which is commonly used in nuclear physics):

$$m_i c^2 = A_i m_u c^2 + \Delta_i \tag{4.20}$$

where m_u is atomic mass unit ($m_u c^2 \approx 931.494$ MeV). To keep the mass conserved we need

$$\sum_i A_i \frac{dY_i}{dt} = \frac{d}{dt}\sum_i A_i Y_i = 0 \tag{4.21}$$

The total nucleon number does not change with time ($\sum_i A_i Y_i = 1$) and only the mass excess terms left in the summation in eq(4.19).

Now let's look at the electron part in eq(4.17),

$$\frac{dY_e}{dt} = \frac{d}{dt}\sum_i Z_i Y_i = \sum_i Z_i \frac{dY_i}{dt} \tag{4.22}$$

where $Y_e \equiv Y_{e^-} - Y_{e^+} = \sum_i Z_i Y_i$ is the net electron number per nucleon. Then the energy generation ϵ becomes

$$\epsilon = \sum_i \left[-\Delta_i - \frac{3}{2}kT - kT^2\frac{\partial \ln G_i}{\partial T} + Z_i\left(T\frac{\partial \mu_e}{\partial T} - \mu_e\right)\right]\frac{dY_i}{dt} \tag{4.23}$$

With the help of libstatmech, we can easily calculation $\frac{\partial \mu_e}{\partial T}$ and μ_e. Thus we are able to compute the energy generation during a finite time Δt.

The energy released will heat up the white dwarf slowly. However if the temperature reached to some certain value, the nuclear burnings will become enormously strong and take over, blowing the entire white dwarf.

4.5 Explosion and Nucleosynthesis

To solve the three first order ordinary differential equations (4.7, 4.8, 4.15), which determine the evolution of size, velocity and temperature, we used **gsl_odeiv_step_rk4**, the 4th order (classical) Runga-Kutta methods.

For each timestep, we evolve the the three equations first. And the run the network to get the energy generation and abundance changes. We use both x, y, T and species abundances to control the timestep.

Here is a typical calculation. In figure(4.3) we can see that the white dwarf starts to explode at about 1 second after we give an initial kick. The time is just around the oscillation tiem period.

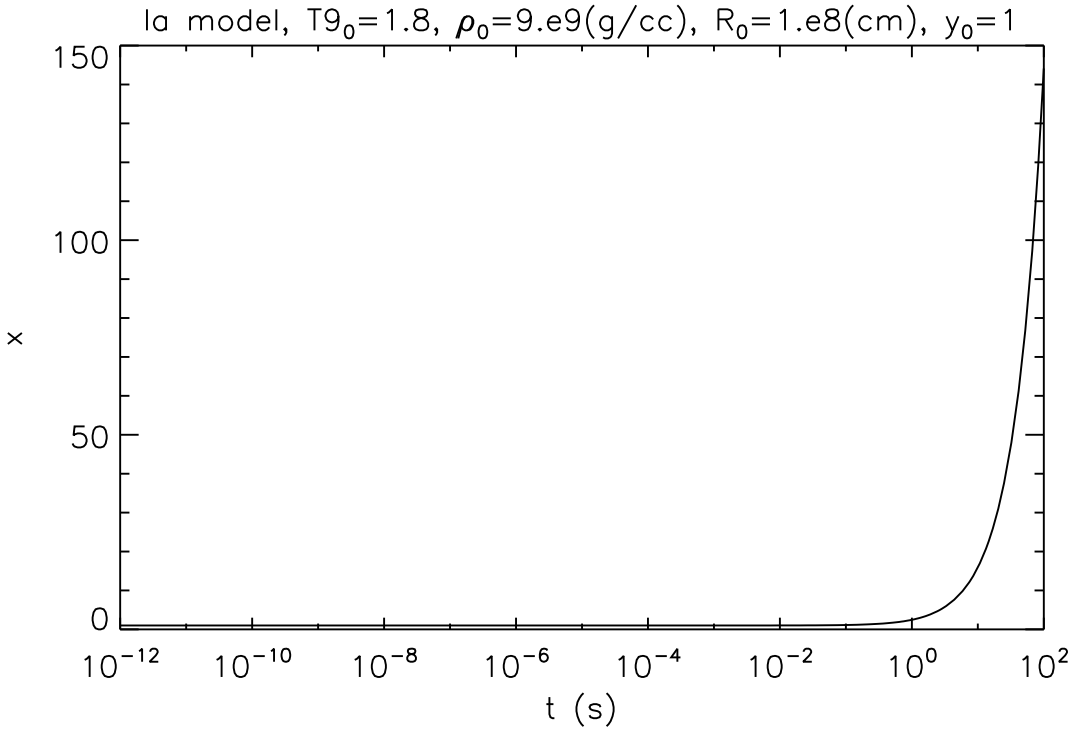

Figure 4.3: Radius vs time, Ia model. White dwarf starts to explode around 1 second.

In figure(4.5) we can see that the neutron-rich iron-group isotopes dominate even before the stage of the explosion. The interesting result is that ^{50}Ti and ^{48}Ca get mass fraction values above or around 10 percent of the total, which is really a large amount of yield. The result shows that with more accurate weak rate, we can get lower Y_e and more neutron-rich isotopes, which may be

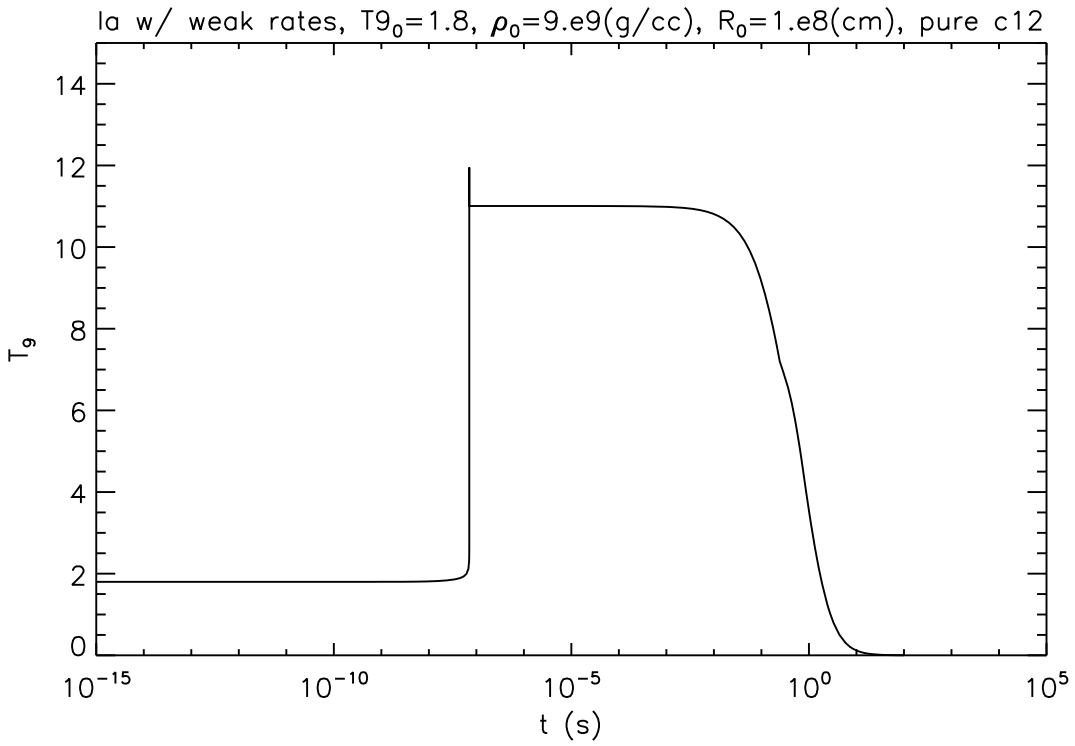

Figure 4.4: Temperature (T_9) vs time, Ia model. Temperature shoots up at about 10^{-7} second due to energy release from intense nuclear reactions.

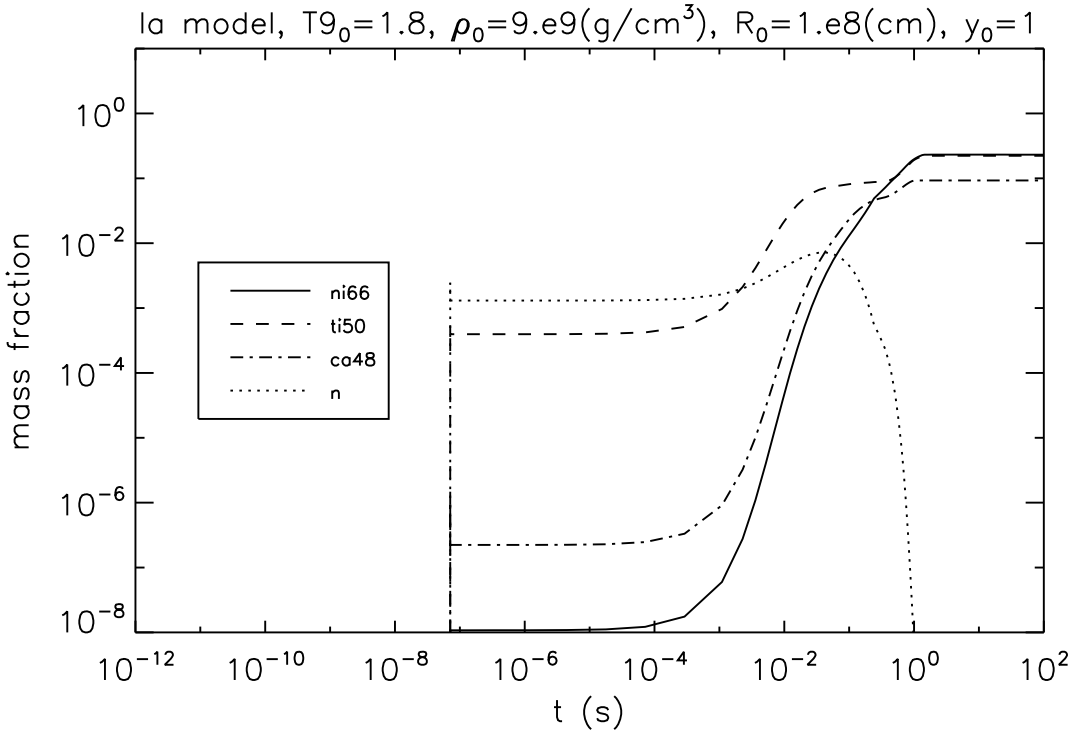

Figure 4.5: Nucleosynthesis of Ia model, neutron-rich iron-group isotopes. Y_e drops to ~ 4.331.

useful to explain the FUN CAIs problem in the future.

Chapter 5

Conclusions

In this thesis I have given a detailed instruction of libstatmech, a library released online and free to use. I have described how people can calculate the reverse rate of weak nuclear interaction from detailed balance on top of libstatmech and libnuceq. And finally I have applied the network code (mainly using libnucnet) to a simple model of a thermonuclear supernova have followed the nucleosynthesis during the explosion.

Although libstatmech is created for calculating nuclear network equilibrium, it can certainly be used in other aspects that need the thermodynamic properties in the system of interest. It is open source, easy to incorporate with other code and pretty light in size. I hope more people would take benefit of it.

The main achievement that is new in this work is that we calculated the weak reverse rate from the detailed balance. With the help of libstatmech and libnuceq, we now are able to handle the electron chemical potential change in each timestep in simulations, for both single zone calculation and our simple Type Ia supernova model. This allows us to evolve nuclear reaction networks into weak nuclear stastistical equilibrium, which has not been done by others before.

We achieved a low Y_e value about 0.4 in our simple Ia calculation, which resulted in more neutron-rich iron-group isotopes and that is important to understand the meteorites and early solar system. Which would be the next step of my research.

Appendices

Appendix A Fermions and Bosons

Fermions and bosons are stored as Libstatmech__Fermion and Libstatmech__Boson structures, respectively in libstatmech. To create a fermion or boson, the user must supply the particle's name, its rest mass in MeV, its multiplicity (usually $2J + 1$, where J is the particle's spin), and its charge (in units of the proton's charge). When a fermion or boson is created, it automatically has attached to it four thermodynamic quantities, namely, the number density, the pressure, the energy density, and the entropy density. These quantities are computed in cgs units. Once a fermion or boson is created, a user may compute one of the four thermodynamic quantities by the API routine Libstatmech__Fermion__computeQuantity() or Libstatmech__Boson__computeQuantity(). These routines will compute the various quantities by numerical integration of the default integrands, which are those for fully relativistic, non-interacting particles (see §2.2 for the definition of these integrals). The user may also invert the number density integrand to compute the chemical potential. The four standard quantities are identified by the strings "number density", "pressure", "energy density", and "entropy", which may also be set by the defined parameters S_NUMBER_DENSITY, S_PRESSURE, S_ENERGY_DENSITY, and S_ENTROPY_DENSITY.

Appendix B Boson Integrals

Here I use similar method for fermion integrals to derive boson integrals.

Start from Grand potential $\Omega(T, V, \mu)$,

$$d\Omega = -SdT - PdV - Nd\mu \tag{1}$$

The coefficients are immediately given by

$$S = -\left(\frac{\partial\Omega}{\partial T}\right)_{V,\mu} \quad P = -\left(\frac{\partial\Omega}{\partial V}\right)_{T,\mu} \quad N = -\left(\frac{\partial\Omega}{\partial \mu}\right)_{T,V} \tag{2}$$

and

$$\Omega(T, V, \mu) = -k_B T \ln Z_G \tag{3}$$

where Z_G is grand partition function and

$$Z_G = \prod_{i=1}^{\infty} \sum_{n=0} (e^{-\beta(\mu-\epsilon_i)})^n \tag{4}$$

For fermion the sum over n is from 0 to 1. However for boson each state can hold as many particles as to infinity. So

$$Z_G = \prod_{i=1}^{\infty} \sum_{n=0}^{\infty} (e^{-\beta(\mu-\epsilon_i)})^n = \prod_{i=1}^{\infty} (1 - e^{\beta(\mu-\epsilon_i)})^{-1} \tag{5}$$

Then,

$$\Omega = -k_B T \ln \prod_{i=1}^{\infty} (1 - e^{\beta(\mu-\epsilon_i)})^{-1} = k_B T \sum_{i=1}^{\infty} \ln(1 - e^{\beta(\mu-\epsilon_i)}) \tag{6}$$

and $\sum_i \to g \int d^3n = gV(2\pi)^{-3} \int d^3k$. Then we substitute k using $\epsilon^2 = \hbar^2 k_B^2 c^2 + m^2 c^4$ and take the derivatives to get the boson integrals eq(2.11 - 2.14).

Appendix C User-Supplied Functions and Integrands for Thermodynamic Quantities

If a user wishes to use a function or an integrand other than the default version to compute a thermodynamic quantity, he or she must supply those. To supply a function for a thermodynamic quantity of a fermion, the user writes a routine with the prototype

```
double
fermion_function(
  Libstatmech__Fermion *p_fermion,
  double d_T,
  double d_mukT,
  void *p_user_data
);
```

Here, **p_fermion** is a pointer to a libstatmech fermion structure, **d_T** is the temperature in K, **d_mukT** is the μ'/kT, the chemical potential (less the rest mass, that is, $\mu' = \mu - mc^2$, where μ is the full chemical potential) divided by kT, where k is Boltzmann's constant, and **p_user_data** is a pointer to a user-defined structure carrying extra data into the routine. The user's routine need not be named **fermion_function**. Analogously, a user-supplied function for a boson thermodynamic quantity has the prototype

```
double
boson_function(
  Libstatmech__Boson *p_boson,
  double d_T,
  double d_mukT,
  void *p_user_data
);
```

For both the fermion and boson function, the user's routine must return the thermodynamic quantity for the input temperature, μ'/kT, and other user data.

A user may also supply an integrand for a thermodynamic quantity. Here the respective prototypes are

```
double
fermion_integrand(
  Libstatmech__Fermion *p_fermion,
  double d_x,
  double d_T,
  double d_mukT,
  void *p_user_data
);
```

and

```
double
boson_integrand(
  Libstatmech__Boson *p_boson,
  double d_x,
  double d_T,
  double d_mukT,
  void *p_user_data
);
```

The input parameters are the same as for the functions. The additional parameter **d_x** is the integration variable.

Once the user has written an appropriate function and integrand, he or she updates them for the fermion or boson. For example, to update the pressure for a fermion **p_fermion** with function **my_pressure_function** and **my_pressure_integrand**, the user calls

```
Libstatmech__Fermion__updateQuantity(
  p_fermion,
  S_PRESSURE,
  (Libstatmech__Fermion__Function) my_pressure_function,
  (Libstatmech__Fermion__Integrand) my_pressure_integrand
);
```

Now when the user calls Libstatmech__Fermion__computeQuantity for **p_fermion** and for the pressure, libstatmech will compute the pressure by first evaluating my_pressure_function for the input

temperature, chemical potential, and other data and will add to it the result of the numerical integration of the integrand my_pressure_integrand. If either the function or integrand is set to NULL, it will not be used in the calculation of the quantity. If the integrand is set to DEFAULT_INTEGRAND, the integrand will be reset to the default for that quantity.

The user may also define his or her own quantity (other than one of the four standard ones). To do so, the user writes the appropriate function and integrand and then sets the quantity for the fermion or boson with the updateQuantity() routine but with the string giving the quantity name set appropriately. The quantity is then computed by calling computeQuantity with that quantity name. For example, suppose we have a fermion pointed to by **p_fermion**. Now we write a Libstatmech__Fermion__Function my_enthalpy_function and a Libstatmech__Fermion__Integrand my_enthalpy_integrand that follow the appropriate prototypes. To compute the enthalpy at a temperature of 10^3 K and $\mu'/kT = -23$ with no extra data, we set the quantity enthalpy and then compute it:

```
Libstatmech__Fermion__updateQuantity(
  p_fermion,
  "enthalpy",
  (Libstatmech__Fermion__Function) my_enthalpy_function,
  (Libstatmech__Fermion__Integrand) my_enthalpy_integrand
);


fprintf(
  stdout,
  "The enthalpy density is %g (ergs/cc)\n",
  Libstatmech__Fermion__computeQuantity(
    p_fermion,
    "enthalpy",
    1000.,
    -23.,
    NULL
  );
```

Examples in the libstatmech distribution further demonstrate how to supply user-defined functions and integrands and how to apply them to thermodynamic quantities.

Appendix D Constraints of NSE and WSE

Helmoholtz free energy f per nucleon is

$$df = -sdT - Pdv + \sum_i \mu_i dY_i \tag{7}$$

where s is the entropy per nucleon, P is the pressure, μ_i is the chemical potential of species i, and Y_i is the abundance of species i per nucleon. To compute the equilibrium, we assume fixed temperature and volume; thus,

$$df = \sum_i \mu_i dY_i \tag{8}$$

The sum on i is over nuclear species and electrons. Charge neutrality requires that Y_e, the number of electrons per nucleon, equal the number of protons in the system:

$$\sum_i Z_i Y_i = Y_e. \tag{9}$$

We thus may write

$$df = \sum_i \left(\mu_i + Z_i \mu_e\right) dY_i, \tag{10}$$

where now the sum on i is only over nuclear species.

Equilibrium occurs for a free energy minimum ($df = 0$). This equilibrium is, however, subject to constraints. The basic constraint on nucleosynthetic systems is that the total number of nucleons is fixed:

$$\sum_i A_i Y_i = 1 \tag{11}$$

We add an undetermined Lagrange multiplier λ to handle the constraint and seek

$$d(f - \lambda g) = 0 \tag{12}$$

where

$$g = \sum_i A_i Y_i - 1 \tag{13}$$

This yields

$$\sum_i \left(\mu_i + Z_i \mu_e - \lambda A_i\right) dY_i = 0 \tag{14}$$

The addition of the Lagrange multiplier allows us to vary each term in the above equation separately. If we consider the neutron ($Z = 0$, $A = 1$), we find

$$\mu_n = \lambda. \tag{15}$$

From the proton we find

$$\mu_p + \mu_e = \mu_n. \tag{16}$$

For general species i, we then find

$$\mu_i = Z_i\mu_p + N_i\mu_n \tag{17}$$

Eqs. (15), (16), and (17) embody weak statistical equilibrium (WSE), as applied to a nucleosynthetic system. Eq. (16) says that there is no net thermodynamic cost in converting a neutron into a proton and electron (we assume neutrinos have zero chemical potential). Eq. (17) says that there is no net thermodynamic cost in converting nuclide i, with charge Z_i and mass A_i, into its constituent neutrons and protons.

For regular nuclear statistical equilibrium (NSE), weak reactions are slow and, hence, Y_e is a constant and is thus an additional constraint:

$$\sum_i Z_i Y_i = Y_e \tag{18}$$

where Y_e is a fixed value. We now let $g_1 = \sum_i Z_i Y_i$ and $g_2 = \sum_i A_i Y_i - 1$ and apply Lagrange multipliers:

$$df - \lambda_1 dg_1 - \lambda_2 dg_2 = 0 \tag{19}$$

$$\sum_i (\mu_i + Z_i\mu_e - \lambda_1 Z_i - \lambda_2 A_i) dY_i = 0 \tag{20}$$

For neutrons, we find $\lambda_2 = \mu_n$. For protons we get $\lambda_1 = \mu_p - \mu_n + \mu_e$. In general, we find

$$\mu_i = \lambda_1 Z_i + \lambda_2 A_i - Z_i\mu_e = Z_i\mu_p + N_i\mu_n. \tag{21}$$

For NSE, the nuclei and break apart into constituent neutrons and protons without any thermodynamic cost, but the total number of protons is fixed.

The final relevant equilibrium is QSE in which the total number of heavy nuclei (defined as

those nuclei with $Z \geq 6$) is fixed. An additional constraint is thus,

$$\sum_{i, Z_i \geq 6} Y_i = Y_h \tag{22}$$

where Y_h is a fixed number. Upon applying three Lagrange multipliers, we find for the heavy nuclei that

$$\mu_i = \mu_h + Z_i \mu_p + N_i \mu_n \tag{23}$$

where μ_h is a chemical potential associated with all the heavy nuclei. If μ_h is positive, then there are too many heavy nuclei compared to what NSE required. Spontaneous evolution of the system then would tend to disintegrate some heavy nuclei. By contrast, if μ_h is negative, the spontaneous evolution is to assemble new heavy nuclei from neutrons, protons, and alpha particles.

To study nuclear statistical equilibria relevant for nucleosynthesis, Prof. Bradley Meyer and I wrote *libnuceq*, a C toolkit for computing arbitrary nuclear statistical equilibria. The user defines equilibrium constraints by an XPath expression. It can easily compute WSE, NSE, and QSE, as well as more complicated equilibria, given the appropriate input nuclear data. It is open-source software and will be publicly released soon.

Appendix E Frequency of an Oscillating Polytrope

The polytrope approximation of an oscillating star is derived here.

In stellar model, a polytrope refers to a solution of the Lane-Emden equation (refer to text book) in which the pressure depends on the density in the form

$$P = K\rho^{\gamma} = K\rho^{\frac{n+1}{n}} \tag{24}$$

where P is pressure, K is a constant, ρ is the density, γ is adiabatic index and n is called polytropic index. A degenerate white dwarf model is well described by a polytrope with $n = 1.5$. Now

$$\frac{P}{P_0} = \frac{\rho}{\rho_0}^{5/3} = x^{-5} \tag{25}$$

Apply this to 4.8 we have

$$\frac{dy}{dt} = \omega^2 \left(\frac{1}{x^3} - \frac{1}{x^2} \right) \tag{26}$$

where $\omega = \sqrt{\frac{P_0}{\rho_0 R_0^2}}$ and will be the frequency of a simple harmonic approximation showed below.

For small displacement δx about $x = 1$, we can rewrite the hydrodynamic equations (4.7) and (4.7) as

$$\frac{d(1+\delta x)}{dt} = \frac{d(\delta x)}{dt} = y \tag{27}$$

and

$$\frac{dy}{dt} = \omega^2[(1+\delta x)^{-3} - (1+\delta x)^{-2}] = -\omega^2 \delta x \tag{28}$$

where small δx approximation is used: $(1+\delta x)^{-3} - (1+\delta x)^{-2} = 1 - 3\delta x - (1 - 2\delta x) = -\delta x$. The result is just a simple harmonic oscillation. If we set initial $\delta x = 0$

$$\delta x = \delta x_{max} \sin \omega t \tag{29}$$

$$y = \delta x_{max} \omega \cos \omega t = y_0 \cos \omega t \tag{30}$$

So the amplitude of the oscillation is like y_0/ω.

Also we notice that $\omega = \sqrt{\frac{P_0}{\rho_0 R_0^2}} \approx \frac{c_s}{R}$, where $c_s = \sqrt{P/\rho}$ is the sound speed. So the

frequency of an oscillating polytrope is just like that of a sound wave traveling from the center to the surface of the star.

Bibliography

[1] Y. Amelin, A. N. Krot, I. D. Hutcheon, and A. A. Ulyanov. Lead Isotopic Ages of Chondrules and Calcium-Aluminum-Rich Inclusions. *Science*, 297:1678–1683, September 2002.

[2] J. S. Brown. Electron-positron equation of state. *M. S. Thesis, Clemson University*, 1995.

[3] J. N. Connelly, Y. Amelin, A. N. Krot, and M. Bizzarro. Chronology of the Solar System's Oldest Solids. , 675:L121–L124, March 2008.

[4] G. M. Fuller, W. A. Fowler, and M. J. Newman. Stellar weak interaction rates for intermediate mass nuclei. III - Rate tables for the free nucleons and nuclei with A = 21 to A = 60. , 48:279–319, March 1982.

[5] G. M. Fuller, W. A. Fowler, and M. J. Newman. Stellar weak interaction rates for intermediate-mass nuclei. IV - Interpolation procedures for rapidly varying lepton capture rates using effective log (ft)-values. , 293:1–16, June 1985.

[6] D. Hartmann, S. E. Woosley, and M. F. El Eid. Nucleosynthesis in neutron-rich supernova ejecta. , 297:837–845, October 1985.

[7] G. C. Jordan, IV, R. T. Fisher, D. M. Townsley, A. C. Calder, C. Graziani, S. Asida, D. Q. Lamb, and J. W. Truran. Three-Dimensional Simulations of the Deflagration Phase of the Gravitationally Confined Detonation Model of Type Ia Supernovae. , 681:1448–1457, July 2008.

[8] K. Langanke and G. Martínez-Pinedo. Rate Tables for the Weak Processes of pf-SHELL Nuclei in Stellar Environments. *Atomic Data and Nuclear Data Tables*, 79:1–46, September 2001.

[9] T. Lee, D. A. Papanastassiou, and G. J. Wasserburg. Calcium isotopic anomalies in the Allende meteorite. , 220:L21–L25, February 1978.

[10] B.S̃. Meyer, T.D̃. Krishnan, and D.D̃. Clayton. 48ca production in matter expanding from high temperature and density. , 462:825–838, May 1996.

[11] F. R. Niederer, D. A. Papanastassiou, and G. J. Wasserburg. Endemic isotopic anomalies in titanium. , 240:L73–L77, August 1980.

[12] L. Pitaevskii and S. Stringari. *Bose-Einstein Condensation*. Oxford University Press, New York, NY, 2003.

[13] L.-S. The, M. F. El Eid, and B. S. Meyer. s-Process Nucleosynthesis in Advanced Burning Phases of Massive Stars. , 655:1058–1078, February 2007.

[14] S. E. Woosley. Neutron-rich Nucleosynthesis in Carbon Deflagration Supernovae. , 476:801–810, February 1997.