Simulating Supernova 1987A Remnants

Ethan Briel, Tianlin Lao, Kaitlyn Le, Allison Lew, Amasha Perera, Jorge Sanchez Mentor: Sangeeta Kumar

University of California, Berkeley Undergraduate Lab at Berkeley, Physics & Astronomy Division

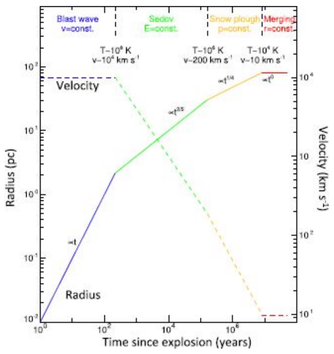
ABSTRACT

We simulated the remnants of Supernova 1987A in 2D over time using N-body simulation. We used parameter values found in the literature for a supernova remnant calculator to generate data for the radius and velocity over time after the initial explosion. Using the values for the radius and velocity, we then simulated the positions of 1000 ejecta particles, which we have decided to model as hydrogen, and implemented the interactions between the particles using an N-body simulation found in Github.

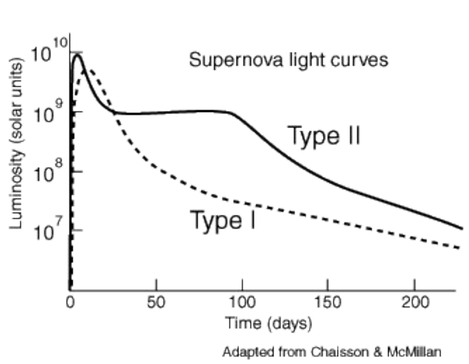
BACKGROUND & MOTIVATIONS

Supernova 1987A is a Type II supernova located in the Large Magellanic Cloud and is the brightest supernova since 1604 and the first directly visible supernova after the rise of modern astronomy. Since it was first observed on February 23, 1987, many observations and mathematical models have been made to better understand and study SN1987A and other celestial events and objects. How does a hypothetical supernova behavior differ from an actual supernova? Does the behavior match prediction models, or are there noticeable differences between the two? Using a python calculator for supernova remnant evolution we want to create a 2D model simulation of the ejecta material from SN1987A as point particles and find differences between theorized and observable data and find explanations for why there are deviations between the two.

Supernova 1987A (SN1987A) is a Type II supernova located in the Large Magellanic Cloud. SN1987A, by definition of a Type II supernova, contains hydrogen in its spectrum. It is the first directly visible supernova after the rise of modern astronomy and it is the first to be observed at every band of the electromagnetic spectrum. It is a core-collapse supernova; at the end of the star's life cycle, nuclear fusion in the core ceases and there are no forces balance the gravitational force. The outer layers of the star all collapse inward, releasing energy and rebounding outward. Energy is released through thermal and kinetic energy. As a supernova evolves, four phases can be identified on its light curve: photospheric, peak, cooling, and the radioactive tail as shown on the graph on the right.



(Micelotta, 2018)



(Hyper Physics)

We utilize the Supernova Remnant Calculator (Leahy and Williams, 2017), a graphical interface that models supernova remnants (SNR) and assumes a spherically symmetric SNR. It is intended to model observations of supernova remnants for obtaining estimates of supernova remnant properties. It does this by taking in various parameters such as age, energy, and initial mass and produces outputs (i.e shock radius and velocity).

METHODS

We had to collect the initial conditions of the progenitor star. The initial conditions we will be using include the initial mass, radius, temperature of the star, etc. These initial parameters will be gathered from papers that have already studied the supernova such as using *Progenitor of SN 1987A* (Podsiadlowski, 1992) and *The Physics of Supernova 1987A* (Murdin et al,2020). These parameters include the following:

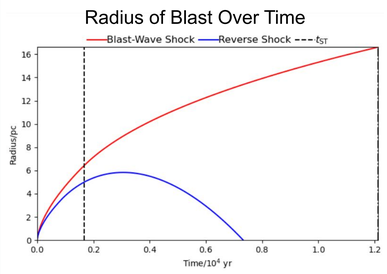
Property	Value
Age	30 years
Energy (10^15 ergs)	3.0
ISM Temperature (K)	166
Ejected Mass (M_sun)	18
Ejecta Power Law Index	8
CSM Power Law Index	0
Electron to Ion Temp Ratio	0.155
Cooling Adjustment Factor	5/2
ISM Number Density (cm^3)	1 H atom
ISM Turbulence/Random Speed (Km/s)	20-50
Initial Mass	15-20 solar masses

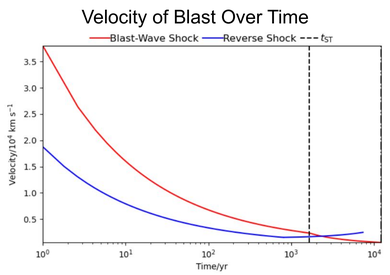
We entered these initial values into the Supernova Remnant Calculator (Leahy and Williams, 2017) for supernova remnants and extracted radius and velocity values over time (both the blast-wave shock and reverse shock) for our simulation. The graphs produced by the calculator demonstrate behavior similar to that depicted in the graph included in our background section, which indicates that our simulated data is aligned with observed behaviors.

Our code utilizes object oriented programming in order to optimize efficiency and versatility. We defined a class for our particles and within that class, we defined our functions for reading our data from the csv file, calculating particle positions, and calculating particle velocities. These objects are then called later when we generate our particles.

The main coding method utilizes N-body simulation which we referenced from an already existing code (Mocz, 2014) and we changed it for our use. In the N-body code, we used a "kick-drift" method where we calculate the acceleration from the N-body, apply this acceleration to the current velocity to give us a new velocity, and then use the new velocity to give us a new position. We used this code to simulate a number of particles moving under gravitational force. To achieve this, the initial conditions of the points, such as mass, position, and velocity were saved in NumPy arrays, and the acceleration of each particle at each time point would be calculated by the total gravitational force on it by other particles. For the next time point, the position and velocity of each particle will be updated from the velocity and acceleration of previous time point. With sufficient amount of particles and time between each time points sufficiently small, this would be a good approximation of real-time physics.

GRAPHS PRODUCED BY PYTHON CALCULATOR

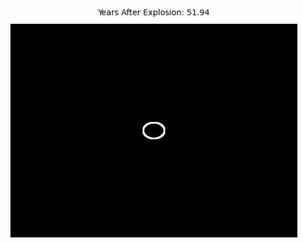


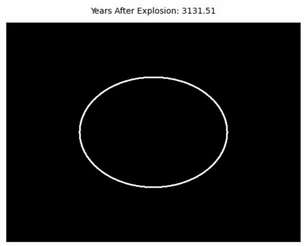


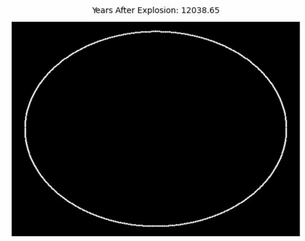
RESULTS & DISCUSSION

The Python Calculator we utilized was able to produce graphs and csv files containing simulated data from the progenitor conditions. These graphs are depicted in the methods section. The data generated by the calculator for the particle velocities over time ended up having a discrepancy with the times corresponding to the growing radius of the explosion. In order to obtain velocities that match the times for the radius data, we calculated the velocities directly by taking the difference between the positions and dividing it by time.

We simulated 1000 particles using our code and were able to produce an animation showing the behavior of the particles over time. The particles simulated were considered to all be identical hydrogen particles for simplicity. The final animation and code are viewable via the following QR code. Our simulation allowed us to visualize how supernova remnants might evolve over time.







FUTURE WORK

Future work can be done to broaden our code to include the various elements that are typically present in a supernova explosion, such as iron and silicon, rather than just hydrogen. There is also possible work to be done in order to convert the simulation into one that depicts the explosion in 3D as opposed to 2D.

ACKNOWLEDGEMENTS

We would like to thank the senior advisor Anmol Desai, program directors Rav Kaur and Saahit Mogan, lab manager Jordan Duan, faculty sponsor Dan Kasen, and mentor Sangeeta Kumar for their support throughout this project.

CITATIONS

- Alsabti_Murdin et al. (2020) Cham. *The Physics of Supernova 1987A. Handbook of Supernovae*. Springer International Publishing. pp. 2181–2210.
- "Analyzing the Universe - Course Wiki: The Sky Is Falling!" *Core-Collapse Supernovae*, www.physics.rutgers.edu/analyze/wiki/cc_supernovae.html.
- Blinnikov, S. I. (1999) "Modeling of the light curve and parameters of the supernova 1987A". *Astronomy Letters*. 25(6). pp. 359-368.
- Micelotta, Elisabetta & Matsuura, Mikako & Sarangi, Arkaprabha. (2018). Dust in Supernovae and Supernova Remnants II: Processing and Survival. Space Science Reviews. 214. 10.1007/s11214-018-0484-7.
- Mocz, Philip. "Create Your Own N-Body Simulation (with Python)." *Medium*, The Startup, 16 Sept. 2020, medium.com/swlh/create-your-own-n-body-simulation-with-python-f417234885e9.
- Leahy, D.A. and Williams, J.~E.(2017) "A Python Calculator for Supernova Remnant Evolution". DOI: 10.3847/1538-3881/aa6af6. 152(5) pp. 239
- Philipp Podsiadlowski. (1992) *The Progenitor of SN 1987A*. In: *Publications of the Astronomy Society*. pp. 717-729.