`nudustc++` : C++ Code for Modeling Dust Nucleation and Destruction in Gaseous Systems

Sarah Stangl, Christopher Mauney

July 2023

1 Summary

We introduce `nudustc++`, a **nu**cleating **dust** code in **C++** modeling dust grain formation, growth, and erosion in gaseous systems. `nudustc++` is a highly parallizable set of code and tools for solving a system of nonlinear ordinary differential equations describing dust nucleation, growth, and erosion for user-specified grain species. It leverages OpenMP and MPI to optimize threading and distribution on available CPUS.

2 Statement of Need

Understanding interstellar dust is crucial for astronomical observations (Draine, 2003), offering key insights into stellar processes. These grains absorb electromagnetic radiation, re-emitting it at longer wavelengths, leading to extinction and a spectral shift towards redder wavelengths. The size and composition of dust introduces variability in opacities and distortion of incident light, resulting in molecular lines and altering the resulting data. Dust formed in asymptotic giant branch (AGB) stars, on pre-existing grains in the interstellar medium (ISM), and within the expanding, cooling ejecta of core collapse supernovae explosions (CCSNe); these grains preserve important information about the nucleosynthetic processes within their host environment, locking up their unique isotopic signatures. However, despite their importance, the quantity, composition, and size distribution of dust formed in supernovae and deposited into the ISM remain poorly constrained.

Models of formation, growth, and weathering of dust are necessary to understand the origin and characteristics of dust in the interstellar medium, shedding light on where and what dust are possible in these environments. Specifically, modelling the formation and survivalability of dust in CCSNe produce a population of dust grains that can be compared with observations, allowing verification of our current understanding of physical models: ISM dust origin, chemical networks, nucleation models, hydrodynamics, supernovae engines, progenitor structure, erosion physics, and stellar compositions.

This project originated from the need to track dust nucleation and destruction in Core-Collapse Supernovae Explosions. It addresses the lack of subgrid physics associated with phase transitions, where hydrodynamical code's timesteps are an order of magnitude larger than those needed to capture gas vapor physics. A smaller more refined grid with chemical networks and smaller timesteps is needed. The code is structured to intake any hydro-dynamical temperature-density trajectory with vapor compositions. This allows `nudustc++` to track dust in a large range of environments: planetary atmospheres, nebulae, hydro-aerosoal formation, explosions, etc. Additionally, if a time series hydro-dynamical profile is unavailable but a dust size distribution and a profile snapshot is, `nudustc++` can calculate the evolution and survivability of the dust. The applications of `nudustc++` extend beyond Supernovae and Astronomy to include any model with thermodynamic and statistical physics.

3 State of the Field

In order to gain a deeper understanding of the origin and characteristics of dust in the interstellar medium, it is imperative to develop models that include the nucleation, chemistry, growth, and erosion of dust. Current methods of calculating dust formation and survival include Classical Nucleation Theory (CNT) and Kinetic Nucleation Theory (KNT).

CNT treats grain formation as a barrier crossing problem. As atoms stick to a cluster, the free energy increases. After reaching a critical size, the free energy decreases as atoms are added. It tracks the nucleation rate by assuming a steady state between monomer attachment and detachment. However, it neglects chemical reactions of formation, destruction, growth by coagulation, and treats the grains as bulk materials. Due to these simplified assumptions, CNT is widely used, but is increasingly less so due to these limitations. Kozasa & Hasegawa (1987) and Bianchi & Schneider (2007) used CNT to model dust grain formation in SN 1987A and SN 1987A-like Supernovae. Todini & Ferrara (2001) used CNT to calculate dust formation in Core Collapse Supernovae Explosions. More recently Paquette et al. (2023) used CNT to model dust formation in the outflows of AGB Stars.

KNT tracks the number densities of clusters with more than two atoms, treating them as particles that grow as spheres. It uses size dependent grain properties in place of bulk material properties. Grains grow by accreting atoms (condensation) through kinetic theory. Grains lose atoms through erosion (destruction) using the principle of detailed balance. While KNT does not assume a steady state between condensation and destruction, it still doesn't take into account the chemical reactions undergone or growth through coagulation. Nozawa et al. (2003) used KNT to model the nucleation and growth of dust in early Population III star supernovae. Nozawa et al. (2006) included destruction to model the effects of high velocity shock waves on dust. Fallest (2012) predicts dust mass yields in CCSNe using KNT.

Other publicly available dust codes include `sndust`, `starchem`, `DustPy` and `astrochem`. `sndust` is described in Brooker et al. (2022) and is a post processing dust nucleation code using KNT. `starchem` tracks the chemical network in stellar environments (Christopher Mauney, 2017). `DustPy` is a python package used to model dust evolution in protoplanetary disks (Stammler & Birnstiel, 2022). `astrochem` computes the chemical abundances and includes gas-dust interactions in astronomical environments such as the interstellar medium, diffuse clouds, and protostars (Maret & Bergin, 2015).

4 Design Principles and Salient Features

`nudustc++` is designed as a flexible, multi-use post-processing code. It is designed to take user supplied data in ascii, binary, or text format, including initial conditions, environment variables, and dust formation networks, and calculate a solution vector. We use an initial composition and interpolated user input data to construct a rate-of-change vector supplied to an implicit integrator in each cell. The user can easily change and modify the interpolator and integrator. Three main configuration paths are currently implemented: destruction only, destruction with nucleation and growth, and nucleation and growth. This allows the reduction of total computations by removing from the solution vector parameters modeling quantities not needed for the calculation path. Output data can be written to ascci, binary, or text files.

Because of the post processing nature of `nudustc++`, modeling grain nucleation and destruction is possible in a large range of physical environments. The user provides the hydrodynamical trajectory file and vapor compositions which can describe Supernovae explosions to planetary atmospheres to interstellar gas clouds and any temperature/density profile extended beyond astrophysics.

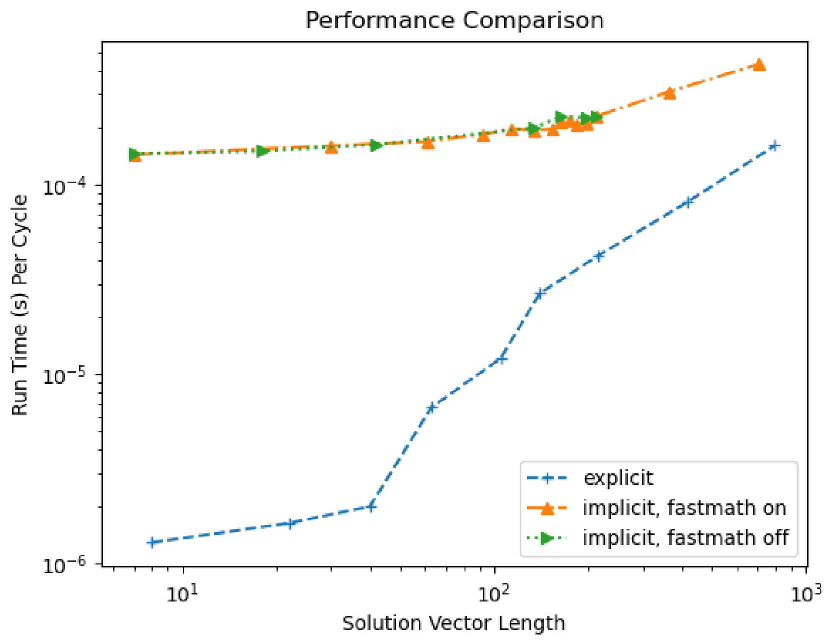
Because `nudustc++` is a post processing code where data is read in separately for each cell with not data flow between cells, there is no shared memory and each cell can be computed independently. This results in an embarrassingly parallel process, allowing for simultaneous computation of each cell and thereby reducing runtime.

5 Performance and Accuracy

Figure 5 shows a comparison between the python `snDust` using an implicit integrator versus `nudustc++` using an explicit integrator. At shorter solution lengths, the implicit integrator has a lot of overhead leading to an increased run time despite the shorter length. The explicit integrator in `C++` out performs the implicit integrator even at longer solution length. Overall, `nudustc++`

outperforms the python version. It is also embarrassingly parallel. Utilizing OpenMP and MPI, the user can run many cells in parallel.

link to github



6 Acknowledgements

This work was supported by the U.S. Department of Energy through the Los Alamos National Laboratory (LANL). LANL is operated by Triad National Seurity, LLC, for the National Nuclear Security Administration of U.S. Department of Energy (Contract No. 89233218CNA000001). This research used resources provided by the Darwin testbed at LANL which is funded by the Computational Systems and Software Environments subprogram of LANL's Advanced Simulation and Computing program (NNSA/DOE). This work is approved for unlimited release with report number LA-UR-XX-XXXXX.

References

Bianchi, S., & Schneider, R. 2007, , 378, 973, doi: http://doi.org/10.1111/j.1365-2966.2007.11829.x10.1111/j.1365-2966.2007.11829.x

Brooker, E. S., Stangl, S. M., Mauney, C. M., & Fryer, C. L. 2022, , 931, 85, doi: http://doi.org/10.3847/1538-4357/ac57c310.3847/1538-4357/ac57c3

Christopher Mauney, D. L. 2017, StarChem. https://github.com/lazzati-astro/starchemhttps://github.com/lazzati-astro/starchem

Draine, B. T. 2003, , 41, 241, doi: http://doi.org/10.1146/annurev.astro.41.011802.09484010.1146/annurev.astro

Fallest, D. W. 2012, PhD thesis, North Carolina State University

Kozasa, T., & Hasegawa, H. 1987, Progress of Theoretical Physics, 77, 1402

Maret, S., & Bergin, E. A. 2015, Astrochem: Abundances of chemical species in the interstellar medium, Astrophysics Source Code Library, record ascl:1507.010. http://ascl.net/1507.010http://ascl.net/1507.010

Nozawa, T., Kozasa, T., & Habe, A. 2006, , 648, 435, doi: http://doi.org/10.1086/50563910.1086/505639

Nozawa, T., Kozasa, T., Umeda, H., Maeda, K., & Nomoto, K. 2003, The Astrophysical Journal, 598, 785, doi: http://doi.org/10.1086/37901110.1086/379011

Paquette, J., Nuth, J., & Ferguson, F. 2023, in American Astronomical Society Meeting Abstracts, Vol. 55, American Astronomical Society Meeting Abstracts, 234.04

Stammler, S. M., & Birnstiel, T. 2022, , 935, 35, doi: http://doi.org/10.3847/1538-4357/ac7d5810.3847/1538-4357/ac7d58

Todini, P., & Ferrara, A. 2001, , 325, 726, doi: http://doi.org/10.1046/j.1365-8711.2001.04486.x10.1046/j.1365-8711.2001.04486.x