Pasta nucleosynthesis: Molecular dynamics simulations of nuclear statistical equilibrium

M. E. Caplan, A. S. Schneider, C. J. Horowitz, and D. K. Berry

Phys. Rev. C **91**, 065802 — Published 22 June 2015

DOI: 10.1103/PhysRevC.91.065802

Pasta Nucleosynthesis: Molecular dynamics simulations of nuclear statistical equilibrium

M. E. Caplan,[1,*] A. S. Schneider,[1,†] C. J. Horowitz,[1,‡] and D. K. Berry[2,§]

[1] *Department of Physics and Nuclear Theory Center, Indiana University, Bloomington, IN 47405, USA*

[2] *University Information Technology Services, Indiana University, Bloomington, IN 47408, USA*

(Dated: June 3, 2015)

Background Exotic non-spherical nuclear pasta shapes are expected in nuclear matter at just below saturation density because of competition between short range nuclear attraction and long range Coulomb repulsion.

Purpose To explore the impact nuclear pasta may have on nucleosynthesis during neutron star mergers when cold dense nuclear matter is ejected and decompressed.

Methods We use a hybrid CPU/GPU molecular dynamics (MD) code to perform decompression simulations of cold dense matter with 51 200 and 409 600 nucleons from $0.080\,\mathrm{fm}^{-3}$ down to $0.00125\,\mathrm{fm}^{-3}$. Simulations are run for proton fractions $Y_P = 0.05$, 0.10, 0.20, 0.30, and 0.40 at temperatures $T = 0.5$, 0.75, and 1.0 MeV. The final composition of each simulation is obtained using a cluster algorithm and compared to a constant density run.

Results Size of nuclei in the final state of decompression runs are in good agreement with nuclear statistical equilibrium (NSE) models for temperatures of 1 MeV while constant density runs produce nuclei smaller than the ones obtained with NSE. Our MD simulations produces unphysical results with large rod-like nuclei in the final state of $T = 0.5\,\mathrm{MeV}$ runs.

Conclusions Our MD model is valid at higher densities than simple nuclear statistical equilibrium models and may help determine the initial temperatures and proton fractions of matter ejected in mergers.

PACS numbers: 26.60.-c,26.30.-k,26.30.Hj,02.70.Ns

I. INTRODUCTION

Determining the properties of neutron-rich matter is vital to our understanding of many astrophysical phenomena. For example, neutron-rich matter is formed during a core-collapse supernova as an increasing electron Fermi energy drives electron capture by nuclei. Neutron-rich matter is also produced during compact star mergers as the outer regions of the colliding stars come into contact. Furthermore, while the outer crust of a neutron star consists of an ion lattice, the core (or at least the outer core) is believed to be made of neutron-rich uniform nuclear matter.

Between the inner crust and core of a neutron star there likely exists a transition layer that involves neutron-rich non-spherical shapes [1, 2]. This transition layer happens near nuclear saturation density where the system is *frustrated* because of an inability to minimize the free-energy of all fundamental interactions. A competition between the attractive short-range nuclear force, with a range of order 1 fm, and the repulsive long-range Coulomb force produces complex nonuniform structures, called *nuclear pasta* [3]. Theoretical symmetry arguments and numerical simulations of the phases of nuclear pasta have identified a variety of structures. These phases are, in order of increasing density: spheroids ("gnocchi"), rods ("spaghetti"), slabs ("lasagna"), uniform matter with cylindrical voids ("anti-spaghetti"), and uniform matter with spherical voids ("anti-gnocchi") [4, 5]. These geometries have been produced by both large-scale classical molecular dynamics simulations and quantum Hartree-Fock calculations [6, 7].

More exotic structures have also been recently identified, such as plates with a lattice of holes or "nuclear waffles" [8], a networked "gyroid" phase [6, 9] and chiral deformations in intertwined lasagna configurations [10]. This indicates that *nuclear pasta* may have a rich variety of possible structures that only now can be studied due to recent increases in computational power.

As nuclear pasta is expected to form during the core collapse phase of a supernova, it may play an important role in the supernovae evolution and resulting neutron stars [11]. For example, supernova neutrinos can scatter coherently from pasta, because neutrino wavelengths are comparable to the pasta sizes. Therefore, calculations of the static structure factor of nuclear pasta can help determine the neutrino opacity in core collapse supernovae [3, 10].

It is also expected that pasta shapes could influence thermal and electrical conductivities of inner crust of neutron stars. Horowitz *et al.* suggest that topological defects in nuclear pasta increase electron-pasta scattering and reduce thermal conductivity [12]. They argue that this could slow crust cooling after accretion in low mass X-ray binaries. Furthermore, topological defects in the pasta would also reduce the inner crust electrical conductivity. This could lead to fast decay of neutron stars magnetic fields, if the fields are supported by currents in the crust and as a result neutron stars would

*Electronic address: mecaplan@indiana.edu
†Electronic address: andschn@indiana.edu
‡Electronic address: horowit@indiana.edu
§Electronic address: dkberry@iu.edu

stop spinning down [13].

Also relevant to neutron star structure are the elastic properties of nuclear pasta, such as its shear modulus and breaking strain. These may determine the maximum size of "mountains" that may be present on neutron stars. On rapidly rotating stars these mountains are energetic sources of gravitational wave radiation [14]. Finally the pasta breaking strain is relevant for crust breaking models of star quakes and magnetar giant flares [15].

However, despite the large literature on nuclear pasta and its relevant astrophysical properties, very little work has been done identifying the role, if any, pasta might play in heavy element nucleosynthesis. Some astrophysical sites involve the ejection of dense neutron rich matter that may originally be in a nuclear pasta phase. As the matter decompresses, the pasta shapes may react to form seed nuclei and free neutrons, and these may later undergo more conventional nucleosynthesis reactions.

It is thought the rapid neutron capture process, or r-process, produces about half of the elements heavier than iron, but the site of the r-process remains uncertain [16]. Previously, supernovae were prime candidates but the most recent simulations of the neutrino driven wind in core collapse supernova are not neutron rich enough to produce heavy r-process elements [16–18].

Neutron star mergers have recently been identified as strong candidate sites for the r-process due to their ejection of neutron rich matter and their relatively high galactic merger rate, which is now expected to be as high as $\sim 10^{-4}\,\mathrm{yr}^{-1}$ [16, 19]. Recently, double neutron star mergers (NS-NS) and neutron star-black hole mergers (NS-BS) have been studied using relativistic hydrodynamic simulations. These simulations find that the nuclear abundances in ejecta match well to solar ratios and are robust for a variety of mass ratios for the merging system [20, 21]. Calculations show that the inner crust provides the largest portion of ejecta mass and that the amount of ejecta varies with the mass ratio of binary. Estimates range between $10^{-3}\,\mathrm{M}_\odot$ for symmetric NS mergers and $10^{-2}\,\mathrm{M}_\odot$ for asymmetric NS mergers, with systems where $M_1/M_2 = 0.55$ ejecting greater than $0.2\,\mathrm{M}_\odot$ [22].

In this paper, we explore the possibility that nuclear pasta, ejected from the inner crust during these mergers, could provide the initial material for the r-process. Pasta properties could be important for the evolution of the material's temperature and entropy. Furthermore, weak interactions in the pasta, such as neutrino or charged lepton capture, will determine the evolution of the proton fraction.

In previous work we identified the average size of gnocchi (mass number of nuclei) using molecular dynamics (MD) simulations [5]. In that work we evolved dense matter with a proton fraction of $Y_P = 0.40$ at a temperature of 1 MeV from high to low densities, $n = 0.10\,\mathrm{fm}^{-3}$ to $n \sim 0.01\,\mathrm{fm}^{-3}$, by expanding the simulation volume at different rates. We observed the nucleation mechanism for a number of different pasta phase transitions and quantified those transitions by calculating the average mean and Gaussian curvatures, allowing us to characterize the phases by Minkowski functionals.

The simulated NS-NS mergers, cited above, observed ejecta evolve from densities near nuclear saturation density and low proton fractions, $Y_P \lesssim 0.10$, to densities many order of magnitude smaller and much higher final proton fractions, $Y_P \sim 0.40$. The timescales for the density to decrease a few orders of magnitude is on the order of milliseconds [22] which is much slower than the timescales of the simulations performed in this work and possible using MD. As we expect nuclear matter to remain in nuclear statistical equilibrium (NSE) while evolving over such timescales, we expand our simulation volume as slow as computationally allowable, and compare to faster expansion rates to confirm that we are expanding slow enough to remain in quasi-static equilibrium, which should be expected of ejecta with millisecond expansion timescales. State of the art computations suggest the ejected matter in NS-NS mergers has very low proton fraction, $Y_p \lesssim 0.10$, that later evolves to $Y_p \sim 0.40$ due to neutrino-driven wind. However, in our work we are limited to simulations with constant proton fractions. Thus, we perform runs for a variety of proton fractions from $Y_P = 0.05$ to 0.40 as these are close to the proton fractions expected to be found during r-process nucleosynthesis.

We note that there have been many MD simulations of heavy ion collisions to study the formation of clusters via multi-fragmentation, see for example [23, 24]. Typically these involve relatively small systems and sometimes neglect Coulomb interactions. In this paper we use the recently developed Indiana University Molecular Dynamics GPU code (IUMD) to simulate larger volumes of nuclear matter as its density decreases. We include full Coulomb interactions and consider a range of proton fractions and temperatures.

In Sec. II we describe our MD formalism and a clustering algorithm to identify which nuclei are formed by stretching of a piece of pasta. In Sec. III we discuss our results and compare to NSE results. We conclude in Sec. IV and for completeness add an Appendix to discuss our GPU code and its performance.

II. FORMALISM

To describe the decompression of ejected matter during a neutron star merger we perform molecular dynamics (MD) simulations with a fixed number of nucleons in a volume that slowly expands. We start our runs at a density of 0.08 fm^{-3}, where complex nuclear pasta phases may be present, and decompress it to 0.01 fm^{-3} or below, where nucleons bind into isolated nuclei. We first discuss our MD formalism in Sec. II A and then review a cluster algorithm in Sec. II B that determines which nucleons belong to which nuclei. This algorithm allows us to deduce the final nuclear abundances produced by

our simulations.

A. Semiclassical nuclear pasta model

Our MD formalism is the same as that used by Horowitz *et al.* and others in previous works [3, 5, 25–29] and is briefly reviewed here. In particular, much of this work is a continuation of Schneider *et al.*[5].

Our simulation volume is a cubic box with periodic boundary conditions which contains point-like protons and neutrons with mass $M = 939\,\mathrm{MeV}$. Electrons are assumed to form a degenerate relativistic Fermi gas and are not explicitly included in the simulations. Protons and neutrons interact via the two-body potentials:

$$V_{np}(r) = ae^{-r^2/\Lambda} + [b-c]e^{-r^2/2\Lambda} \tag{1a}$$

$$V_{nn}(r) = ae^{-r^2/\Lambda} + [b+c]e^{-r^2/2\Lambda} \tag{1b}$$

$$V_{pp}(r) = ae^{-r^2/\Lambda} + [b+c]e^{-r^2/2\Lambda} + \frac{\alpha}{r}e^{-r/\lambda}. \tag{1c}$$

The n and p indices indicate whether the potential describes a neutron-proton, a neutron-neutron, or a proton-proton interaction. Meanwhile, r is the separation between each pair of interacting nucleons and quantities a, b, c, and Λ are parameters of the model. Their values are found in Table I and are chosen to approximately reproduce some bulk properties of pure neutron matter and symmetric nuclear matter, as well as the binding energies of selected nuclei [3].

TABLE I: Nuclear interaction parameters. The parameter a defines the strength of the short-range repulsion between nucleons, b and c the strength of their intermediate-range attraction and Λ the length scale of the nuclear potential.

a (MeV)	b (MeV)	c (MeV)	Λ (fm^2)
110	−26	24	1.25

The proton-proton interaction includes the Coulomb repulsion that is screened by the electron gas. This screening has a characteristic length λ that depends on the fine structure constant α and the electron Fermi momentum $k_F = (3\pi^2 n_e)^{1/3}$, where n_e is the electron density (assumed equal to the proton density) and the electron mass is m_e. Its value is

$$\lambda = \frac{\pi^{1/2}}{2\alpha^{1/2}}\left(k_F\sqrt{k_F^2+m_e^2}\right)^{-1/2}, \tag{2}$$

though in this work we fix λ to the slightly smaller value 10 fm, for all proton fractions, to agree with the value used in earlier work. We do not expect our results to be very sensitive to the precise value of λ. A discussion of the effect of λ on final results can be found in Refs. [8, 30].

We use a cut-off radius for the nuclear potential of 11.5 fm, and no cutoff radius for the Coulomb potential. The nuclear potential is assumed to be zero and is not computed for separations greater than the cut-off distance. The boundary conditions allow particles to only interact with the nearest periodic image of other nucleons. Due to the short range of the nuclear potential, the computation of the nuclear force can be greatly accelerated with the periodic construction of neighbor-lists, which are discussed in detail in the Appendix.

After computing all the inter-particle forces, the nucleon positions and velocities are updated using a velocity-Verlet algorithm [31]. After the timestep Δt is completed, the box size is increased by a small amount. The length of each side of the box l_i $(i = x, y, z)$ and the volume V is

$$l_i(t) = l_i(0)\left(1+\dot{\xi}_i t\right) \tag{3a}$$

$$V(t) = V(0)\left(1+\dot{\xi} t\right)^3 \tag{3b}$$

where $l_i(0)$ and $V(0)$ are the initial side length and volume of the box, and $\dot{\xi}_i$ is the expansion rate. To preserve the cubic geometry, $\dot{\xi}_i$ is the same for all i. Particle positions and velocities are not incremented with the change in box volume, and are allowed to respond dynamically to the changing simulation volume. Furthermore, particles that cross one side of the box and reenter on the other do not have their velocities rescaled. Also, in order to approximately maintain an isothermal expansion, the velocities of all nucleons are rescaled every 100 time steps so that the average kinetic energy per particle is $(3/2)kT$. The errors these approximations introduce are discussed in detail in Schneider *et al.*[5] and are found to be negligible for the temperatures and expansion rates used in our simulations.

B. Cluster Algorithm

To find clusters of protons and neutrons in our simulations we use the Minimum Spanning Tree (MST) algorithm used in Schneider *et al.*[5] and common in other molecular dynamics studies of nuclear pasta [28, 33].

The MST algorithm identifies the nearest neighbors of each nucleon and builds a list of which cluster each nucleon belongs to. First, the algorithm examines every pair of protons i and j. A proton i is determined to belong to a cluster C if i is within a cut-off distance r_{pp} of at least one proton j that is also a part of C. As in the nucleon potential calculations the MST algorithm also accounts for periodicity in the system. Schneider *et al.* found that $r_{pp} = 4.5\,\mathrm{fm}$ was an acceptable value for all densities by examining the proton-proton correlation function $g_{pp}(r)$ [5]. If no protons j are within range of a proton i then i is considered its own cluster. After a list of protons in each cluster have been assembled, the neutrons in each cluster are counted. A neutron is a member of a cluster C if it is within a distance r_{np} of at least one proton j that belongs to C. Again following

TABLE II: Mass fraction of free neutrons, mean mass number of clusters ($A \geq 12$), and mean charge number of clusters from simulations of 51 200 nucleons, expanded at four proton fractions (Y_P) and four stretch rates ($\dot{\xi}$), are shown. All data shown is for the final configuration of the simulation, when $n = 0.01\,\mathrm{fm}^{-3}$. The data in the rightmost columns are offered for comparison to NSE tables. This data is generated from a Virial expansion of 8980 species of nuclei with $A \geq 12$, and shows fair agreement with the IUMD results [32].

		IUMD			NSE		
Y_P	$\log_{10}\dot{\xi}$	$M_{\text{free neutrons}}$	$\overline{A} \pm \sigma_A$	$\overline{Z} \pm \sigma_Z$	$M_{\text{free neutrons}}$	$\overline{A}$	$\overline{Z}$
0.1	-5	0.6538	90.12 ± 24.20	26.25 ± 7.69	Unavailable		
	-6	0.6561	94.08 ± 18.01	27.57 ± 5.69			
	-7	0.6578	99.18 ± 17.42	29.20 ± 5.57			
0.2	-5	0.3748	147.75 ± 33.88	47.31 ± 10.95	0.3335	179.3	53.80
	-6	0.3731	146.22 ± 24.96	46.69 ± 8.30			
	-7	0.3704	136.12 ± 20.32	43.29 ± 6.72			
0.3	-5	0.1359	186.47 ± 73.19	64.75 ± 24.59	0.0475	184.4	58.07
	-6	0.1377	175.13 ± 34.48	60.94 ± 12.16			
	-7	0.1367	166.74 ± 34.71	57.95 ± 12.44			
0.4	-5	0.0109	369.37 ± 426.03	149.32 ± 167.45	0.0001	194.4	77.75
	-6	0.0121	179.40 ± 29.83	72.64 ± 11.93			
	-7	0.0115	190.25 ± 29.34	76.94 ± 11.83			
	-8	0.0111	191.74 ± 24.55	77.55 ± 9.79			

Schneider *et al.* we set $r_{np} = 3\,\mathrm{fm}$. Neutrons that are not apart of any cluster are counted as free neutrons.

As neutrons are constantly being exchanged between the clusters and the free neutron gas, there exists a small probability that neutrons get miscounted if they are in the process of escaping or making close fly-bys. This probability is small, and has little effect on our results. Other cluster algorithms exist, such as those that check the energy between particle pairs to check if they are bound, such as the Minimum Spanning Tree in Two-particle Energy Space (MSTE). While MSTE has the advantage of discriminating against questionable surface neutrons, in our low energy simulations both MSE and MSTE have similar results and, therefore, we find the MSE is sufficient in this work [5, 28, 33].

III. RESULTS

We start in Sec. III A with a discussion of the effect of expansion rate on the final configuration of the simulation. In Sec. III B we discuss the final populations of nuclides present in our simulations at 0.75 and 1.0 MeV. In Sec. III C we discuss the results of simulations at 0.5 MeV, specifically the non-equilibium effects observed at higher proton fractions.

A. Expansion Rate

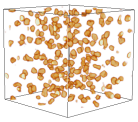
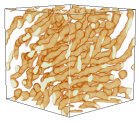
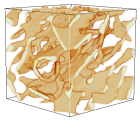
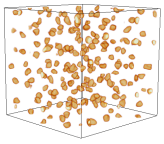
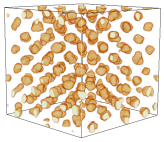
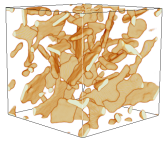
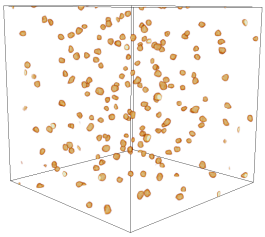
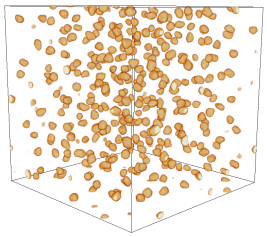
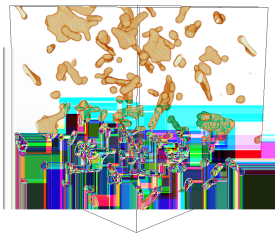
In milliseconds matter ejected in a neutron star merger can decompress from near nuclear saturation density to much lower densities; a rate of change in density of order $\mathcal{O} \sim -3n_0/(10^{19}\,\mathrm{fm}/c)$, where $n_0 \simeq 0.16\,\mathrm{fm}^{-3}$ is the nuclear saturation density. Current computational limits restrict MD simulation expansion rates in Eq. (3b) to 10^{12} or more times faster. This corresponds to a maximum of about 10^{10} MD time steps.

Our goal in this Section is to identify the fastest expansion rate $\dot{\xi}_{\max}$ the pasta in our simulations could experience before moving significantly away from equilibrium. For this end we loosely defined a simulation to be away from equilibrium whenever slab or rod-like nuclei appeared in the final low density state. This happens whenever the expansion rate $\dot{\xi}$ is so fast that the pasta does to have time to fission within the alloted simulation time. Determining $\dot{\xi}_{\max}$ allows us to minimize the computation times for the simulations in Sec. III B.

We use our highest temperature, $T = 1\,\mathrm{MeV}$, to determine $\dot{\xi}_{\max}$ for three reasons. Firstly, it allows us compare our nuclide populations to available nuclear statistic equilibrium data obtained with a Virial expansion, which is more consistent with our classical approach. This is opposed to the Hartree-Fock calculations which have been done for lower temperatures. Secondly, it allows us to compare to our previous work [5]. Lastly, the lower temperature data produced non realistic structures, which are discussed in more detail in Sec. III C.

We simulate 51 200 particles at a temperature of 1.0 MeV for expansion rates of $\dot{\xi} = 10^{-5}$, 10^{-6}, 10^{-7} and $10^{-8}c/\,\mathrm{fm}$ for $t = \dot{\xi}^{-1}$. We start our simulations at an initial density of $n(t=0) = n_0/2 = 0.08\,\mathrm{fm}^3$ and expand them to a final density $n(t = \dot{\xi}^{-1}) = 0.01\,\mathrm{fm}^3$. The initial configurations are first equilibrated from random for 5×10^5 timesteps at a constant density of $n = 0.08\,\mathrm{fm}^{-3}$ using a time step $\Delta t = 2\,\mathrm{fm}/c$. In Table II we compare the mass fraction of free neutrons, the mean mass number of clusters, and the mean charge number of clusters at $t = \dot{\xi}^{-1}$ in our simulations to NSE data. The NSE data is obtained from the tables of G. Shen *et al.*[32], which include 8980 species of nuclei with $A \geq 12$ as well as protons, neutrons and alpha particles and show a fair

TABLE III: (Color online) Comparisons of configurations at several densities obtained from three different simulations, shown to scale. The figures are generated in Paraview by finding isosurfaces of charge density. The dark surfaces are generated where $n_Z = 0.03\,\mathrm{fm}^{-3}$, and the lighter surfaces at the boundary show where $n_Z > 0.03\,\mathrm{fm}^{-3}$. The first column shows the density of the configurations in each row.

$n\,(\mathrm{fm}^3)$	$T = \mathbf{1.0\ MeV}$ $Y_P = \mathbf{0.1}$	$T = \mathbf{1.0\ MeV}$ $Y_P = \mathbf{0.4}$	$T = \mathbf{0.5\ MeV}$ $Y_P = \mathbf{0.4}$
0.0601			
0.0195			
0.0116			
0.0027			

agreement with the IUMD results.

We note that MD results showed an excess of free neutrons when compared to NSE results. Also, MD simulations produced some light isotopes ($A < 12$) which accounted for less than 1% of the mass fraction of all simulations and, thus, were excluded from the analysis of mean cluster mass and charge number for comparison to NSE. The vast majority of these light clusters were protons bound to several neutrons. Though this neutron rich hydrogen nuclei show up often in our semi-classical simulations they are not expected to occur in nature.

We observe that the only simulations clearly out of equilibrium (using our loose definition) were those with proton fraction $Y_P = 0.3$, 0.4 stretched at a rate $\dot{\xi} = 10^{-5}c/\,\mathrm{fm}$. These simulations were not able to equilibrate due to the short time of the simulation (50 000 timesteps). In the $Y_P = 0.3$ simulation there are many large oblong clusters that are approaching fission and the mass fraction of rod-like nuclei is 10%, where these clusters have mass number $A > 300$. In contrast, more than 66% of the mass of the system in $Y_P = 0.4$ case are in rod-like nuclei clusters, $A > 300$, with the largest of these clusters having mass number $A = 3063$.

Though we find that expansion rates of $\dot{\xi} = 10^{-5}c/\,\mathrm{fm}$ are too fast and produce non-equilibrium effects, final configurations obtained from expansion rates of $\dot{\xi} = 10^{-6}$ c/fm and $\dot{\xi} = 10^{-7}$ c/fm do not differ much from each other and are in good agreement with NSE data. For the one case we tested $\dot{\xi} = 10^{-8}$ c/fm, we also find good agreement between our results and NSE results. For this reason, we choose $\dot{\xi}_{\mathrm{max}} = 10^{-7}$ c/fm as the expansion rate for the main simulations discussed in Sec. III B as they can be computed in a reasonable amount of time while still producing results that are close to equilibrium.

Having determined which expansion rates can produce nuclei with sizes that agree with NSE at low densities ($n = 0.01\,\mathrm{fm}/c$) we run simulations to check whether the same results could be obtained from constant density simulations. We equilibrated systems from random with different proton fractions at $T = 1.0\,\mathrm{MeV}$ and $n = 0.01\,\mathrm{fm}/c$ for $2 \times 10^6\,\mathrm{fm}/c$. Nuclei sizes from these constant density simulations, as well as nuclei sizes obtained from expansion simulations and NSE, are shown in Table IV. The number of free neutrons and the size of nuclei formed in the constant density simulations quickly reached a steady state and barely changed during the second half of the run. Though the number of free neutrons obtained in the constant density simulations and the expansion simulations (at $0.01\,\mathrm{fm}^{-3}$) are similar, we observe that the nuclei formed at constant density are significantly smaller. This shows that, at least for our MD simulations, decompression from a higher density plays a

role in the size of clusters formed.

B. Nuclide Abudances at 0.75 and 1.0 MeV

As discussed in the previous section we evolve a uniformly random distribution of nucleons with a given proton fraction and temperature at a constant density of $n = 0.08\,\mathrm{fm}^{-3}$ for 5×10^5 timesteps of length $\Delta t = 2\,\mathrm{fm}/c$. We then use the final configuration of this $n = 0.08\,\mathrm{fm}^{-3}$ run as a starting point to our expansion runs with expansion rate $\dot{\xi} = 10^{-7}c/\,\mathrm{fm}$.

Our expansion runs survey five proton fractions, $Y_P =$ 0.05, 0.10, 0.20, 0.30, and 0.40, and three temperatures, $T =$ 0.5, 0.75, and 1.0 MeV. All of these 15 cases were simulated using 51 200 particles. Additionally, we ran simulations with 409 600 particles for the $Y_P = 0.05$ case to get better statistics for the mean number of protons in each cluster. All simulations were evolved for $3\dot{\xi}^{-1} = 3\times10^7\,\mathrm{fm}/c$ with a 2 fm/c timestep on 32 GPU nodes on Big Red II (see Appendix). The final density of the runs was 0.00125 fm^{-3}.

We note that the higher proton fractions runs, $Y_P =$ 0.3 and 0.4, form a lasagna phase at $n = 0.08\,\mathrm{fm}^{-3}$ for the temperatures in our simulations. Meanwhile, runs with lower proton fractions, $Y_P \leq 0.2$, have less regular structure that are dependent on the temperature. During the expansion, we observe that the pasta phase transitions for the $Y_P = 0.40$ run at $T = 1\,\mathrm{MeV}$ occurred at densities consistent with the results of our previous work [5]. Specifically, between densities of $0.02\,\mathrm{fm}^{-3}$ and $0.01\,\mathrm{fm}^{-3}$ the spaghetti fissions to produce a lattice of spheroidal gnocchi, whose statistics are presented in the last column of Figs. 1 and 2, and is visualized in the center column of Tab. III.

Using the MSE clustering algorithm we observed that below densities of $0.01\,\mathrm{fm}^{-3}$ the proton populations of individual gnocchi stayed relatively constant for all our runs. However, new clusters formed occasionally whenever protons and neutrons that escaped from one of the large nuclei bound to make a light isotope. This effect was more pronounced at $T = 1.0\,\mathrm{MeV}$, which produced a larger populations of light isotopes ($A < 12$) than the T =0.75 MeV runs. This can be seen by comparing the distribution of clusters near the origin in the top and bottom rows of Figure 1.

From the number for the free neutron mass m_n in Figure 1 we note that the free neutron population increases with temperature and decreases with proton fraction. It is also worth mentioning, though it is not shown, that in all cases the free neutron count increased as the simulation volume increased, until it eventually approached a constant value near the end of each run. This can be explained by the fact that in these semi-classical simulations nucleons bound to a nuclei often acquire enough kinetic energy to overcome the potential barrier that keeps them in a cluster. Once these neutrons are free they can become bound to another cluster by colliding with it. It is expected that the collision cross section for such event decreases significantly as the simulation volume increases. On the other hand, when large nuclei shed neutrons their binding energy per nucleon increases and the probability of losing yet another neutron decreases. Thus, by observing this asymptotic behavior of the free neutron count near the end of our simulations we conclude that these two competing effects happen at about the same rate and so we do not expect a significant change in the mass of free neutron by running our simulations longer.

Other trends we observe from our simulations, see Figures 1 and 2, are (1) distribution of cluster masses and charges is approximately Gaussian, (2) for a given proton fraction the mean mass and charge are larger for lower temperature, (3) the number of free neutrons is larger at higher temperatures and, finally, (4) the average cluster charge and mass increases with proton fraction in the range of proton fractions of our simulations, $Y_P \leq 0.40$. Figures 5 and 6 show that the ratio Z/N of protons to neutrons in the clusters formed within each simulation is approximately constant and that the deviations around the average Z/N ratio are larger for $T = 1.0\,\mathrm{MeV}$ than for $T = 0.75\,\mathrm{MeV}$.

Consistent with our observations in Sec. III A, we also observe the presence of light isotopes at $T = 1.0$ MeV which account for less than 1% of the total mass, but we find no such clusters at $T = 0.75$ MeV. This can be seen by comparing the top and bottom rown of Figures 1 and 2 or comparing the region near the origin of Figures 5 and 6.

C. Nuclide Abudances at 0.5 MeV

Much of the discussion offered for the $T = 0.75$ and 1.0 MeV simulations apply to the $T = 0.5$ MeV runs at proton fractions of $Y_P = 0.05$ and 0.10. We note a slight increase in the mean mass and mean charge of clusters, with a decrease in the mass fraction of free neutrons. This can be seen in Fig. 3 and Fig. 4

However, the simulations at $T = 0.50$ MeV with $Y_P \geq 0.20$ show significant non-equilibrium effects, with a static population of super heavy clusters in their final state. Some of the clusters formed large rods, slabs, and spheroids as seen in the last column of Tab. III. We find that these unrealistic shapes are the result of a phase transition in the nucleons. While the *macrophase* of the system can be described by the pasta shapes, the *microphase* of the nucleons within the pasta structures can be either liquid or solid. At large proton fractions and low enough temperatures the nucleons in the simulation undergo a phase transition to a solid microphase. These dynamics were recently observed in MD simulations of pasta by Alcain *et al.*[10], who found that the nucleons undergo a solid-liquid phase transition at low densities near $T = 0.5$ MeV. Since at the temperatures and densities simulated in this work nucleons are not expected to form a lattice, even at zero temperature, this

TABLE IV: Comparison of nuclear sizes obtained from constant density runs at $n = 0.01\,\mathrm{fm}^{-3}$, expansion runs with $\dot{\xi} = 10^{-7}c/\,\mathrm{fm}$, and the NSE model of Ref. [32].

	$n = 0.01\,\mathrm{fm}^{-3}$			$\dot{\xi} = 10^{-7}c/\mathrm{fm}$			NSE		
Y_P	$M_{\text{free neutrons}}$	$\overline{A} \pm \sigma_A$	$\overline{Z} \pm \sigma_Z$	$M_{\text{free neutrons}}$	$\overline{A} \pm \sigma_A$	$\overline{Z} \pm \sigma_Z$	$M_{\text{free neutrons}}$	$\overline{A}$	$\overline{Z}$
0.1	0.6523	82.32 ± 19.02	23.83 ± 5.81	0.6578	99.18 ± 17.42	29.20 ± 5.57	Unavailable		
0.2	0.3587	100.48 ± 21.50	31.36 ± 7.02	0.3704	136.12 ± 20.32	43.29 ± 6.72	0.3335	179.3	53.80
0.3	0.1271	111.70 ± 26.69	38.39 ± 9.58	0.1367	166.74 ± 34.71	57.95 ±12.44	0.0475	184.4	58.07
0.4	0.0112	106.57 ± 28.18	43.11 ± 11.45	0.0115	190.25 ± 29.34	76.94 ±11.83	0.0001	194.4	77.75

TABLE V: Average final ratio N/Z for clusters in simulations with $T = 0.75\,\mathrm{MeV}$ and $T = 1.0\,\mathrm{MeV}$.

Y_p	Z/N 0.75 MeV	Z/N 1.0 MeV
0.05	0.442	0.500
0.10	0.478	0.535
0.20	0.517	0.578
0.30	0.561	0.628
0.40	0.672	0.729

result simply shows that the MD model is not reliable at such low temperatures. Although our observation is consistent with Alcain *et al.*, it is worth mentioning the nuclear potential in their work differs slightly from ours.

D. Simulations with 409 600 Nucleons

We perform simulations containing 409 600 particles for three reasons: (1) to test the scaling of our code performance, which is discussed in the appendix; (2) to check for finite size effects, see discussion in Ref. [34]; and (3) to specifically obtain better cluster statistics for the $Y_P = 0.05$ cases, see Fig. 7. With 51 200 particles at $Y_P = 0.05$ the simulation only contains 2 560 protons, which yields the poor Gaussians in the first column in Figs. 1, 2, 3, & 4.

Aside from the poor cluster statistics at low proton fractions, we observe negligible finite size effects. Simulations with 409 600 particles are in good agreement with simulations using 51 200 particles. In particular, we observe that the mass fraction of free neutrons, mean cluster charge, and mean cluster mass are in good agreement with results found in Secs. III B & III C.

IV. CONCLUSIONS

We have performed molecular dynamics (MD) simulations of the decompressing nuclear matter that may be ejected during neutron star mergers. We slowly expand a simulation volume and then analyze the final configurations for the different kinds of nuclei (clusters) present. We find that as long as the expansion rate is not too high and the temperature is not too low the system appears to attain nuclear statistical equilibrium when decompressed. In general, at temperatures of 0.75 and 1.0 MeV, expanding MD simulations at rates of $\dot{\xi} = 10^{-7}\,\mathrm{c/fm}$ or slower produces distributions of nuclei similar to many nuclear statistical equilibrium (NSE) models. Expansion at faster rates do not allow the clusters to attain NSE and produces a greater spread in the masses and charges of the final clusters.

This simple MD model can describe matter over a large range of densities where the system may be uniform nuclear matter, a variety of complex nuclear pasta phases, or a collection of more or less isolated nuclei. At lower densities we reproduce many features of NSE models, such as the mass fraction of free neutrons and the mean mass and charge of heavy nuclei. Therefore, our model can describe matter at high densities, during the initial stages as it is ejected during neutron star mergers, through latter stages where nuclei and free neutrons provide the initial conditions for more conventional nuclear reaction network calculations of nucleosynthesis.

Traditional nucleosynthesis calculations involve a high density description using a simplified astrophysical equation of state that is somewhat arbitrarily matched to a nuclear reaction network at low density. This high density equation of state often does not include complex pasta shapes. Merger simulations with the EOS are used to predict initial conditions of proton fraction and entropy for a different low density description involving an explicit reaction network. The simplified high density description could lead to errors in the initial conditions for the reaction network because it may not be able to accurately treat weak interaction rates or thermodynamic properties of nuclear pasta.

Instead, our MD simulations can be used to describe the system in a unified framework valid over a range of both high and low densities. Therefore these simulations will allow more accurate determinations of the initial conditions for more traditional nucleosynthesis reaction networks. In future work we will use our simulations to predict a variety of weak interaction rates for charged lepton and neutrino capture on the complex nuclear pasta phases. This will provide better predictions for the initial proton fraction and temperature of the neutron rich matter ejected in NS mergers.

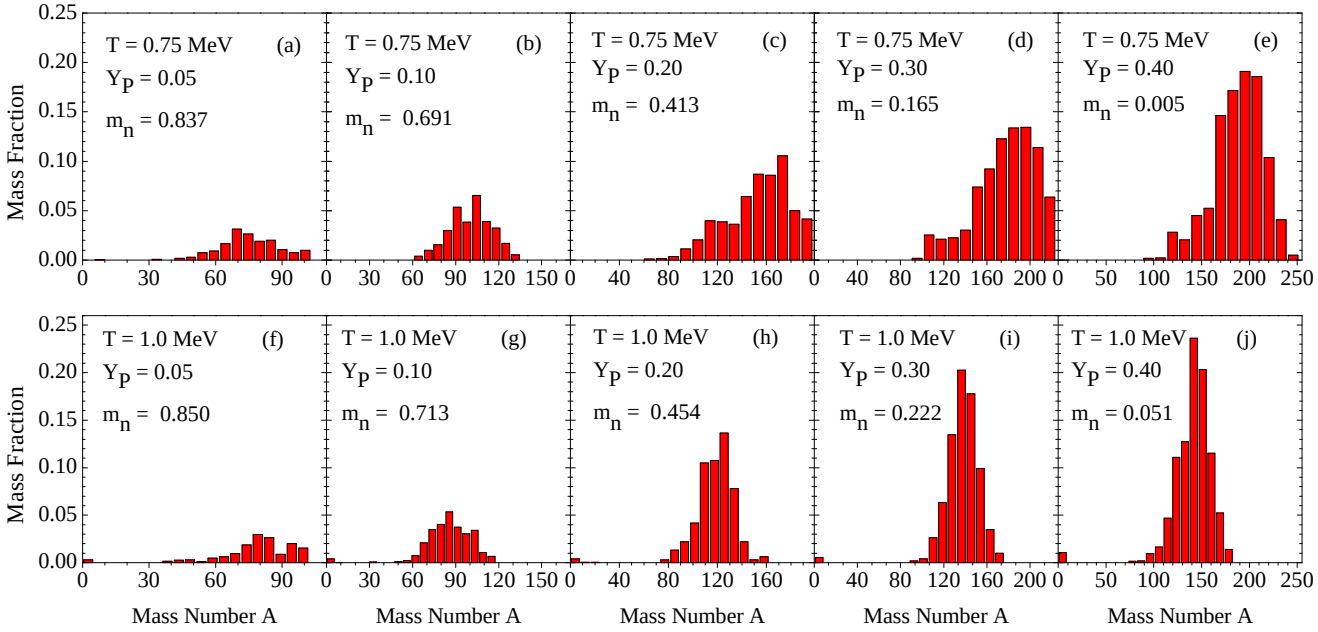


FIG. 1: (Color online) Distribution of cluster masses for $T = 0.75\,\mathrm{MeV}$ (top row) and $T = 1.0\,\mathrm{MeV}$ (bottom row) for proton fractions $Y_P = 0.05$, 0.10, 0.20, 0.30, and 0.40 (columns) from simulations of 51 200 nucleons at a density of $n = 0.00125\,\mathrm{fm}^{-3}$. The mass fraction of free neutrons m_n is indicated.

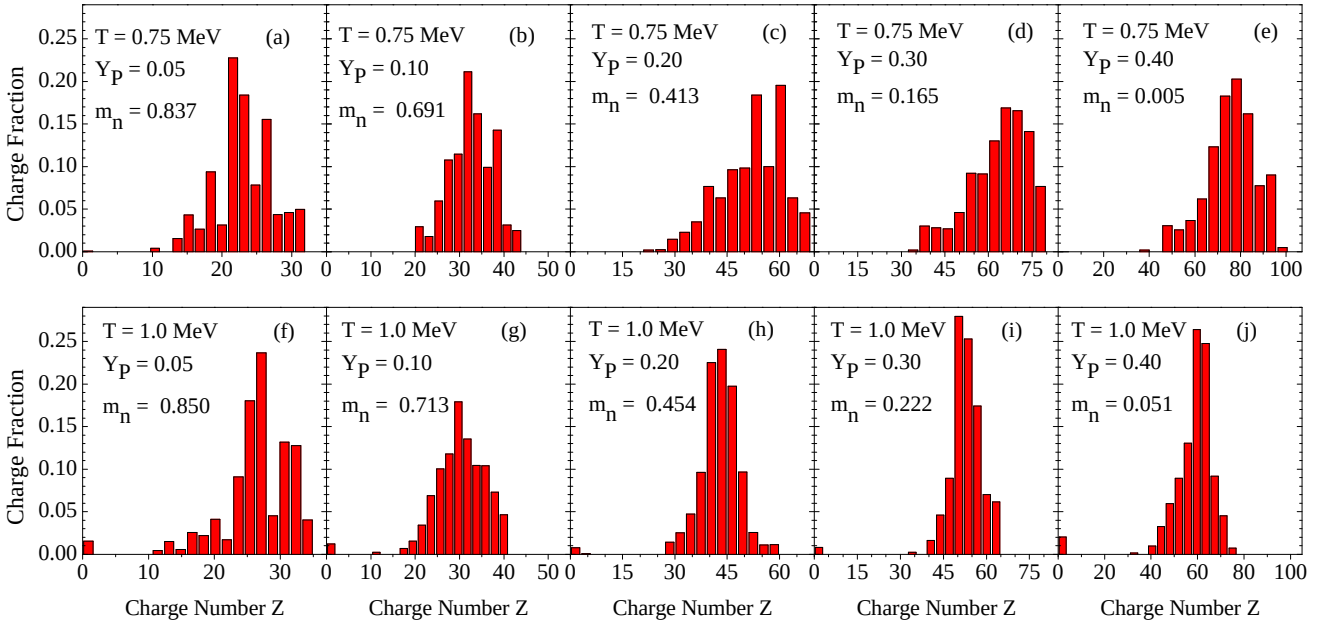


FIG. 2: (Color online) Distribution of charge for $T = 0.75\,\mathrm{MeV}$ (top row) and $T = 1.0\,\mathrm{MeV}$ (bottom row) for proton fractions $Y_P = 0.05$, 0.10, 0.20, 0.30, and 0.40 (columns) from simulations of 51 200 nucleons at a density of $n = 0.00125\,\mathrm{fm}^{-3}$. The mass fraction of free neutrons m_n is indicated.

Acknowledgments

We are grateful to David Reagan at the Advanced Visualization Laboratory - Indiana University for his help with ParaView. We would also like to thank Indiana University for access to the BigRed II supercomputer. This research was supported in part by DOE grants DE-FG02-87ER40365 (Indiana University) and DE-SC0008808 (NUCLEI SciDAC Collaboration).

Appendix: GPU code

The simulations in this work were computed using a new version of the Indiana University Molecular Dynam-

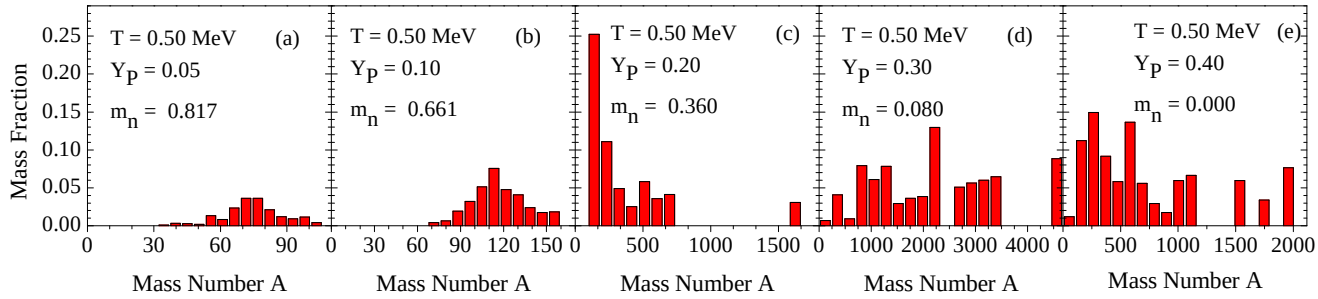


FIG. 3: (Color online) Distribution of cluster masses for $T = 0.5\,\mathrm{MeV}$ for proton fractions $Y_P = 0.05$, 0.10, 0.20, 0.30, and 0.40 from simulations of 51 200 nucleons at a density of $n = 0.00125\,\mathrm{fm}^{-3}$. The mass fraction of free neutrons m_n is indicated.

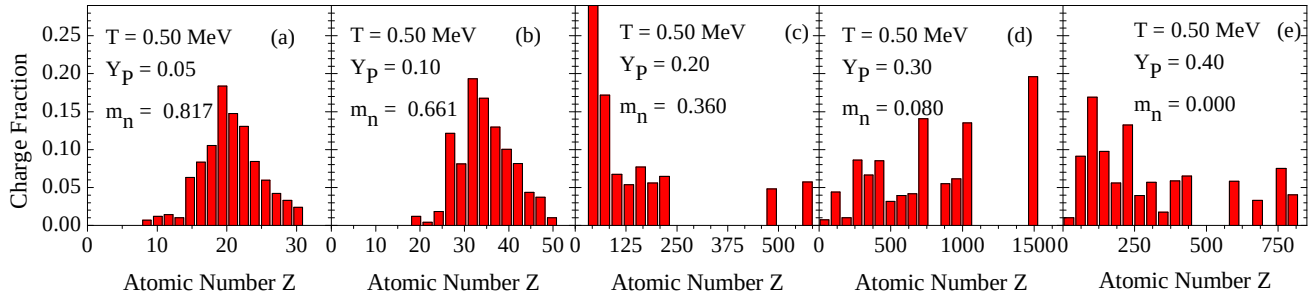


FIG. 4: (Color online) Distribution of charge for $T = 0.5\,\mathrm{MeV}$ for proton fractions $Y_P = 0.05$, 0.10, 0.20, 0.30, and 0.40 from simulations of 51 200 nucleons at a density of $n = 0.00125\,\mathrm{fm}^{-3}$.

ics (IUMD) Fortran code, which has been modified from our previous work Ref. [5] to run on the GPU nodes of the Big Red II supercomputer at Indiana University.

1. The IUMD code

The IUMD code has been used for a decade and run on the original Big Red supercomputer at Indiana University (an IBM JS21), and the Kraken supercomputer at Oak Ridge National Lab (a Cray XT5). Both these machines consisted of general purpose multi-core CPU nodes.

In mid 2013, Indiana University acquired a Cray XE6/XK6 supercomputer. The XE6 part of the machine consists of 344 general purpose dual 16-core CPU nodes. The XK6 part consists of 676 accelerated nodes, containing one 16-core CPU and one Nvidia Kepler K20 GPU [35]. A new version of IUMD (version 6.3.1) was created to take advantage of the powerful accelerated nodes. IUMD 6.3.1 is explained in greater detail in [8]. Here we explain it only enough to understand the performance we observed in our expansion runs.

IUMD is a parallel Fortran code which uses MPI to pass data between nodes, OpenMP on each CPU to take advantage of its 16 cores, and CUDA Fortran to take advantage of the GPUs. In discussing the code, it is helpful to think of the two-particle interactions as making up a *force matrix*, whose ij element is the force $\boldsymbol{f}_{ij}$ that *source* j exerts on *target* i. Overall of course, targets and sources are the same particles. In IUMD the force matrix is partitioned into a $P \times Q$ block matrix, where PQ is the total number of MPI processes (nodes). Each MPI process is responsible for calculating forces represented by one block. Note that this decomposition is an abstract one, rather than one based on geometry. Targets and sources are distributed randomly among MPI processes so that each process is responsible for the entire simulation volume, but only a fraction of the targets and sources in the volume. This is different from parallel algorithms where each process is responsible for all particles in a subvolume. The advantage of IUMD is particles do not have to be transferred from process to process as they move from one subvolume to another. Orchestrating such transfers involves a level of coding complexity that may be difficult to optimize. The disadvantage of IUMD is that an MPI_Allreduce must be performed to combine partial forces from different processes to get the total force on a target. However, this is a single call to an MPI subroutine that hopefully has been optimized in the MPI library. A similar MPI_Allreduce is used to update source particle positions each time step. Again, see [8] for details.

The force calculation is by far the most time consuming part of each time step. It consists of a short-range two-nucleon nuclear force, and a screened Coulomb force between protons. IUMD calculates the nuclear force on the CPUs using an efficient neighbor list algorithm. Each target's neighbor list consists of those sources within distance $r_{nuc} + \delta r_{nuc}$. Forces are calculated only for those

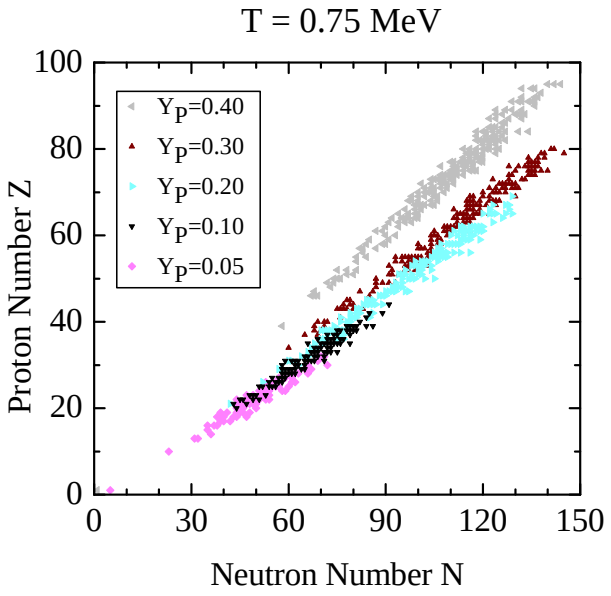


FIG. 5: (Color online) Atomic number Z and neutron number N of nuclides for simulations at $T = 0.75$ MeV with $Y_P = 0.05$, 0.10, 0.20, 0.30 and 0.40 described in Sec. III B.

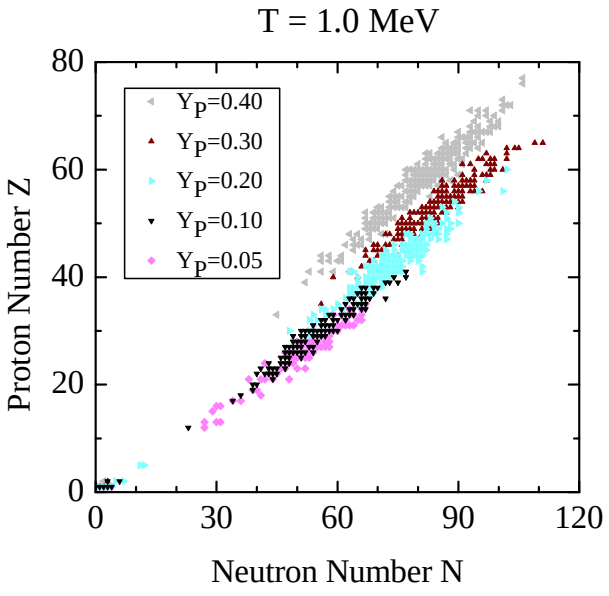


FIG. 6: (Color online) Atomic number Z and neutron number N of nuclides for simulations at $T = 1.0$ MeV with $Y_P = 0.05$, 0.10, 0.20, 0.30 and 0.40 described in Sec. III B.

sources within interaction range r_{nuc}; however, sources within a halo of thickness δr_{nuc} are included in the list so it does not have to be rebuilt every time step. Rather, a target's list needs to be rebuilt only when the distance it has moved, plus the maximum distance any source on its node has moved exceeds δr_{nuc}. For only then is it possible that a source not in the target's list could have come into interaction range. Since rebuilding neighbor lists is time-consuming and disruptive, all lists on all processes are rebuilt if any one list needs to be. We set $r_{nuc} = 11.5$ fm for the runs in this paper. Due to the rapid decrease of the nuclear force, it will rarely register for separations greater than this, even in 64-bit IEEE arithmetic.

While the CPUs calculate the nuclear force, the GPUs calculate the screened Coulomb force using a simple all-pairs, or *particle-particle* (PP) algorithm. The work for each process to calculate the screened Coulomb force scales like $(Y_P N)^2/PQ$, where N/P and N/Q are the number of targets and sources on each process. With work rising as the square of the number of particles, the PP algorithm would seem to be very inefficient. However, each Kepler K20 GPU on BigRed II contains 2496 single precision floating point cores, 832 double precision cores, and 416 special function units for computing square root, exponential and trigonometric functions. This is a powerful computational capacity for implementing simple, straightforward algorithms, whereas complex algorithms may not work so well. Thus we keep the algorithm as simple as possible, eschewing even the use of Newton's Third Law, as it entails branching that would slow the GPU. We depend instead on the Kepler K20's massive parallelism. In the next section we show that for 51 200 nucleons, $Y_p <= 0.400$ and densities of interest in nuclear pasta research, the PP algorithm running on GPUs still finishes before the neighbor list algorithm on the CPUs. Even for 409 600 nucleons it outperforms the CPU at low proton fractions.

2. Performance

The PP algorithm for the Coulomb interaction takes order $O((Y_p N)^2/PQ)$ amount of work per process. For a fixed number of nucleons, it therefore scales as the square of the proton fraction. On the other hand, the work involved in the neighbor list algorithm for the nuclear force is independent of Y_p, but depends linearly on density, because the the number of sources within interaction range of each target depends linearly on density. We therefore expect the work to calculate the nuclear force for all N/P targets on a process to be of order $O((N/P)(4\pi/3)(r_{nuc} + \delta r_{nuc})^3(n/Q))$. To this should be added the work to build neighbor lists. How frequently neighbor lists need to be rebuilt depends on the density and temperature of the system. We chose $\delta r_{nuc} = 4.0$ fm, to reduce the frequency, while still keeping lists relatively short. With this value of δr_{nuc} we found that neighbor lists were rebuilt approximately every ten timesteps for high temperatures and densities, and every hundred time steps for low density, low temperature, equilibrium systems. The algorithm IUMD uses to

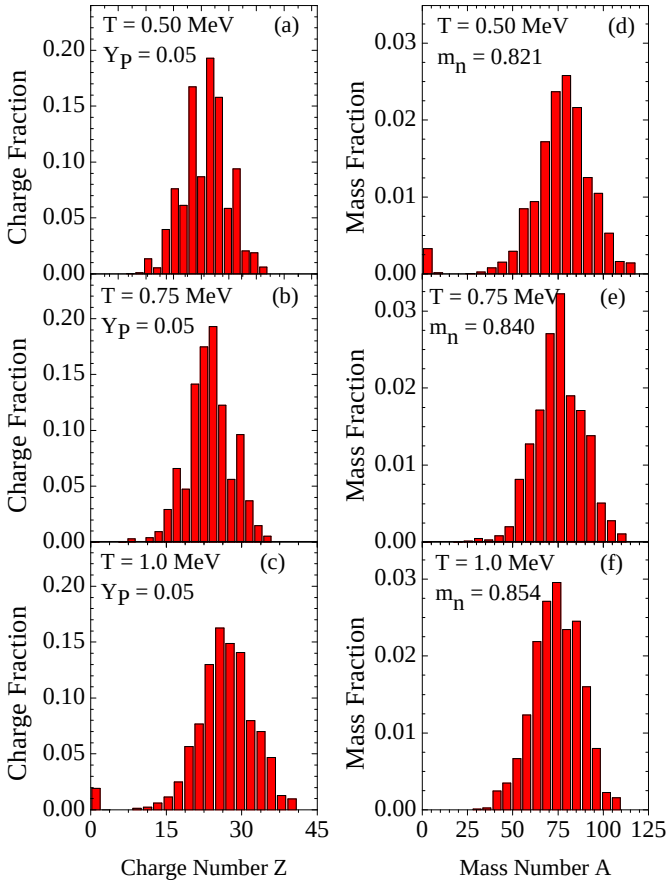


FIG. 7: (Color online) Charge fractions (left) and mass fractions (right hand side) for simulations of 409 600 particles at $T = 0.5$, 0.75, and 1.0 MeV with $Y_P = 0.05$ as described in Sec. III D. The mass fraction of free neutrons m_n is indicated on the left hand panels.

build neighbor lists involves dividing the simulation volume into cubical cells of width $r_{nuc} + \delta r_{nuc}$ and checking only sources in a target's cell and its 26 neighboring cells. The work in this algorithm is thus of order $O(27(N/P)(r_{nuc} + \delta r_{nuc})^3(n/Q))$, the same order as the nuclear force calculation itself. The exact coefficients in this estimate and the nuclear force calculation were not determined, but the work to decide if a source should go in a neighbor list is much less than calculating the force it exerts. Thus the neighbor list build may be noticeable, but still a small fraction of the force calculation.

We collected timing data from the simulations described in Sec. III B and Sec. III D to check the above workload estimates. These simulations involved 51 200 and 409 600 particles, run on 32 and 128 GPU nodes respectively. The simulations were evolved for 3×10^7 fm/c with 2 fm/c per timestep on Big Red II, and were expanded at a rate of $\dot{\xi} = 10^{-7}$ c/fm, starting at an initial nucleon density of $0.08\,\mathrm{fm}^{-3}$ and decreasing to $0.00125\,\mathrm{fm}^{-3}$. Since the density was monotonically decreasing the average time per MD timestep could be calculated from timestamps on checkpoint files output by the program every 5×10^5 timesteps during the 15×10^6 timesteps of the run.

Figure 8 shows the average time per MD timestep as a function of density for the five proton fractions we ran with 51 200 nucleons. For $Y_p = 0.05$ and 0.10 time decreases almost linearly as density decreases, in accord with our model for the work involved in the nuclear calculation. This indicates the nuclear calculation takes longer than the Coulomb at all densities. We call this the linear case. For $Y_p = 0.40$ time decreases linearly from $n = 0.06$ to 0.01 fm^{-3}, then flattens out. For this proton fraction the nuclear force calculation takes longer above 0.01 fm^{-3}, while the Coulomb calculation takes longer below 0.01 fm^{-3}. We call this the broken linear case.

For $Y_p = 0.20$ and 0.30 the decrease is linear for high density, but slower and nonlinear at lower density. According to our simple model, if the performance model is not linear or broken linear, then the Coulomb calculation should take longer at all densities, which we call the flat performance case. However, these two proton fractions fit none of these cases. What the model does not take into account, is that the size of neighbor lists actually depends on local density rather than mean density. As nucleons cluster into pasta shapes, or nuclei at very low mean density, the local density inside clusters goes to saturation density, while outside is a low density gas of neutrons, a few free protons and alpha particles. Thus the curves for $Y_p = 0.05$ and 0.10 are linear because there are few clusters. For $Y_p = 0.40$ clusters have not yet formed by the time mean and local density diverge. However, for $Y_p = 0.20$ and 0.30 clusters do form while the nuclear force calculation still dominates, so the curves gradually flatten. Note that at about $n = 0.005\,\mathrm{fm}^{-3}$ the curve appears to suddenly go horizontal. This may be the point at which the Coulomb calculation finally dominates the time.

The slopes of the curves do appear to depend on proton fraction and temperature. At $T = 1.00$ MeV, neighbor lists are rebuilt more often. Since the work to do that depends linearly on density, they increase the coefficient in the order estimate, thus increasing the slope.

One odd characteristic of Figure 8 is that the curves do not all start at the same point at $n = 0.06\,\mathrm{fm}^{-3}$. Other tests we did to investigate this indicate it is due to differences in the way MPI communication is set up between different runs. Either the node set allocated to the run, or the way the MPI runtime system places processes on nodes, may affect the efficiency of message passing.

In Figure 9 we show the time per MD timestep for the $N = 409600$ nucleon run for $Y_p = 0.05$ on 128 nodes. Because of the low proton fraction, the nuclear calculation dominates at all densities, resulting in the nearly linear decrease in time as density decreases. Again there is a slight temperature dependence, being greater for $T = 1.00$ MeV. Finally, we note the dependence on N.

This run involved eight times as many nucleions on four times as many nodes (128 vs. 32). According to our model for the nuclear force, the run should have therefore taken twice as long. Figure 9 shows it performed better than this, taking on average only about 1.6 times as long. This may be due to better performance of the MPI communication for larger messages, and better performance in the velocity Verlet update with a larger number of particles.

In general, we observe that lower proton fractions can be computed the fastest and speed up with decreasing density, while higher proton fractions take longer to compute and only slightly accelerate for decreasing densities. We observe negligible dependence on temperature. While the 409 600 particle simulations mark an eightfold increase in the number of particles from the 51 200 particle runs, they were only run on four times the GPU nodes. In good agreement with our prediction of performance scaling linearly with N, we observe the time-per-timestep to double at high density, but to only increase by $\sim 50\%$ for very low density.

[1] C. Pethick and A. Potekhin, Physics Letters B **427**, 7 (1998).
[2] D. G. Ravenhall, C. J. Pethick, and J. R. Wilson, Phys. Rev. Lett. **50**, 2066 (1983).
[3] C. J. Horowitz, M. A. Pérez-García, and J. Piekarewicz, Phys. Rev. C **69**, 045804 (2004).
[4] M. Okamoto, T. Maruyama, K. Yabana, and T. Tatsumi, Physics Letters B **713**, 284 (2012).
[5] A. S. Schneider, C. J. Horowitz, J. Hughto, and D. K. Berry, Phys. Rev. C **88**, 065807 (2013).
[6] B. Schuetrumpf, M. A. Klatt, K. Iida, G. E. Schröder-Turk, J. A. Maruhn, K. Mecke, and P.-G. Reinhard, Phys. Rev. C **91**, 025801 (2015).
[7] W. G. Newton and J. R. Stone, Phys. Rev. C **79**, 055801 (2009).
[8] A. S. Schneider, D. K. Berry, C. M. Briggs, M. E. Caplan, and C. J. Horowitz, Phys. Rev. C **90**, 055805 (2014).
[9] K. Nakazato, K. Oyamatsu, and S. Yamada, Phys. Rev. Lett. **103**, 132501 (2009).
[10] P. N. Alcain, P. A. Giménez Molinelli, and C. O. Dorso, Phys. Rev. C **90**, 065803 (2014).
[11] C. Horowitz, (2012), arXiv:1212.6405 [nucl-th] .
[12] C. J. Horowitz, D. K. Berry, C. M. Briggs, M. E. Caplan, A. Cumming, and A. S. Schneider, Phys. Rev. Lett. **114**, 031102 (2015).
[13] J. A. Pons, D. Viganò, and N. Rea, Nature Physics **9**, 431 (2013).
[14] C. J. Horowitz and K. Kadau, Phys. Rev. Lett. **102**, 191102 (2009).
[15] A. I. Chugunov and C. J. Horowitz, Contributions to Plasma Physics **52**, 122 (2012).
[16] M. Arnould, S. Goriely, and K. Takahashi, Phys. Rep. **450**, 97 (2007), arXiv:0705.4512 .
[17] C. J. Horowitz and G. Li, Phys. Rev. Lett. **82**, 5198 (1999).
[18] C. J. Horowitz, Phys. Rev. D **65**, 083005 (2002).
[19] D. R. Lorimer, Living Rev. Relativity **11** (2008), 10.12942/lrr-2008-8.
[20] S. Goriely, A. Bauswein, and H.-T. Janka, Astrophys. J. Lett. **738**, L32 (2011), arXiv:1107.0899 [astro-ph.SR] .
[21] O. Korobkin, S. Rosswog, A. Arcones, and C. Winteler, Mon. Not. R. Astron. Soc. **426**, 1940 (2012), arXiv:1206.2379 [astro-ph.SR] .
[22] R. Oechslin and H.-T. Janka, Monthly Notices of the Royal Astronomical Society **368**, 1489 (2006).
[23] S. Chikazumi, T. Maruyama, S. Chiba, K. Niita, and A. Iwamoto, Phys. Rev. C **63**, 024602 (2001).
[24] J. Singh, R. K. Puri, and J. Aichelin, Physics Letters B **519**, 46 (2001).
[25] C. J. Horowitz, M. A. Pérez-García, J. Carriere, D. K. Berry, and J. Piekarewicz, Phys. Rev. C **70**, 065806 (2004).
[26] C. J. Horowitz, M. A. Pérez-García, D. K. Berry, and J. Piekarewicz, Phys. Rev. C **72**, 035801 (2005).
[27] C. J. Horowitz and D. K. Berry, Phys. Rev. C **78**, 035806 (2008).
[28] C. O. Dorso, P. A. Giménez Molinelli, and J. A. López, Phys. Rev. C **86**, 055805 (2012).
[29] P. N. Alcain, P. A. Giménez Molinelli, J. I. Nichols, and C. O. Dorso, Phys. Rev. C **89**, 055801 (2014).
[30] P. A. Giménez Molinelli and C. O. Dorso, Nuclear Physics A **933**, 306 (2015).
[31] L. Verlet, Phys. Rev. **159**, 98 (1967).
[32] G. Shen, C. J. Horowitz, and S. Teige, Phys. Rev. C **82**, 045802 (2010).
[33] C. O. Dorso and J. Aichelin, Physics Letters B **345**, 197 (1995).
[34] P. A. Giménez Molinelli, J. I. Nichols, J. A. López, and C. O. Dorso, Nuclear Physics A **923**, 31 (2014).
[35] I. Base, "Big Red II at Indiana University (unpublished)," .

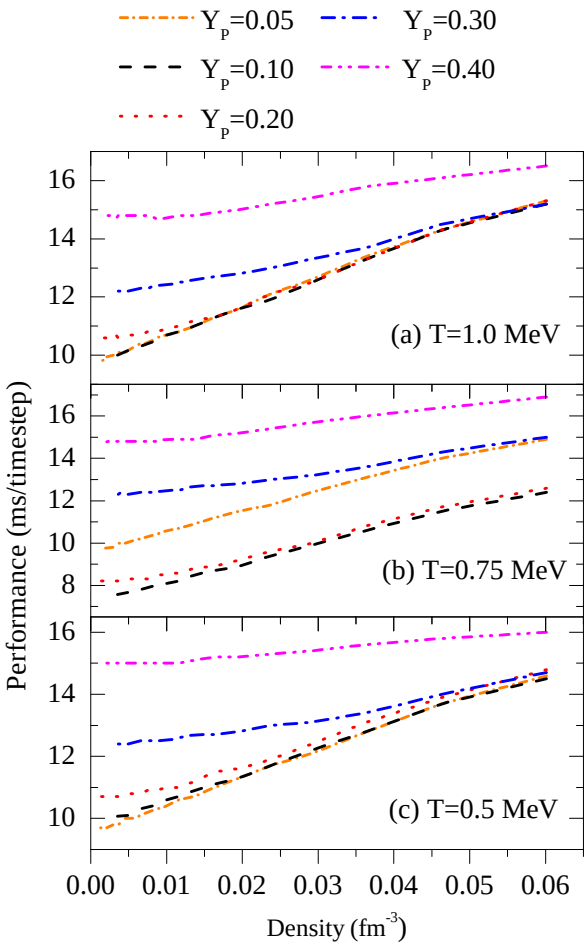


FIG. 8: (Color online) Performance of the IUMD GPU code for the 15 simulations described in Secs. III B & III C. All computations were performed using 51 200 particles with 32 GPU nodes on Big Red II.

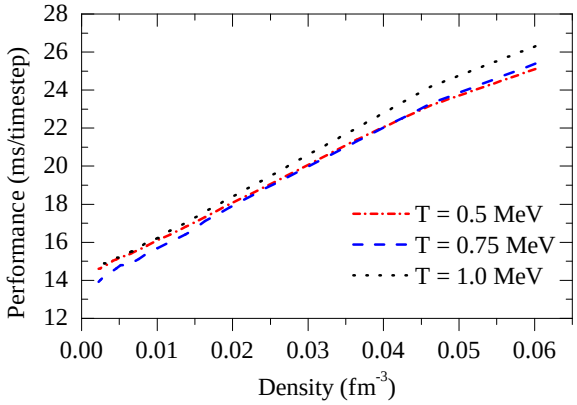


FIG. 9: (Color online) Performance of the IUMD GPU code for the 3 simulations described in Sec. III D. All simulations were performed using 409 600 particles, $Y_P = 0.05$, with 128 GPU nodes on Big Red II.