

THE SPEED-UP OF A BOLTZMANN-VLASOV CODE

R. TABACU¹, M. C. RAPORTARU^{2,3,*}, E. SLUSANSCHI¹, V. BARAN³, A. I. NICOLIN^{2,3}

¹Computer Science and Engineering Department, University POLITEHNICA of Bucharest, Splaiul
Independentei 313, Bucharest, Romania

²Horia Hulubei National Institute for Physics and Nuclear Engineering, Reactorului 30, Magurele,
Romania

³Faculty of Physics, University of Bucharest, Atomistilor 405, Magurele, Romania

*Corresponding author *E-mail*: mraportaru@nipne.ro

Received September 7, 2015

We report on the OpenMP parallelization of a Fortran-based Boltzmann-Vlasov code that is of interest to the experiments at ELI-NP, specifically for the interpretation of results concerning the excitation of pygmy and giant dipole resonances. We show that the optimized codes can be successfully employed to obtain in reasonable running times physically relevant results for the dynamics of such collective modes. The current results can serve as a starting point for future computational investigations into fusion-fission dynamics, also of interest at ELI-NP.

1. INTRODUCTION

Numerical investigations into nuclear and atomic dynamics typically rely on sequential computer recipes that use only one CPU core. This is physically associated with the time integration of the equations that govern the evolution of a given system, a task that usually precludes a direct parallelization. However, at every given time step the dynamics requires information about many quantities (observables) that determine the subsequent evolution of the system. Since the computing time of these observables varies nonlinearly with some parameters that define the numerical implementation scheme, the approach is not particularly efficient.

The Boltzmann-Vlasov equation discussed in this paper represents one of the most advanced semi-classical numerical tools used for the description of nuclear [1–4] and other mesoscopic quantum systems [5] which accounts both for mean-field effects as well as two body collisions in the presence of Pauli blocking. One numerical solution of the equation requires quantities that depend quadratically on the number of so-called test particles (*i.e.*, pseudo-particles) associated with every physical particle (*i.e.*, nucleon) in the system. As a good spanning of the phase space of the system (*i.e.*, nucleus in the case of collective modes or nuclei in the case of fusion-fission) requires a large number of test particles, the accuracy of the numerical results is strongly limited by the available hardware infrastructure.

In this brief contribution we show an OpenMP [6] parallelization of widely-used Boltzmann-Vlasov code, that provides considerable speed-up. The rest of the

article is structured as follows: in Section 2 we show the Boltzmann-Vlasov equation and its computational reduction to a (very large) set of ordinary differential equations of Hamilton type, in Section 3 we detail the parallelization recipe, while in Section 4 we gather the main numerical results and our concluding remarks.

2. THE TEST PARTICLES METHOD FOR NUCLEAR BOLTZMANN-VLASOV EQUATION

The well-known semi-classical Boltzmann-Vlasov equation for the one-body distribution function for protons and neutrons is:

$$\frac{\partial f_q(\mathbf{r}, \mathbf{p}, t)}{\partial t} + \frac{\mathbf{p}}{m} \cdot \nabla_{\mathbf{r}} f_q(\mathbf{r}, \mathbf{p}, t) - \nabla_{\mathbf{r}} U_q(\mathbf{r}) \cdot \nabla_{\mathbf{p}} f_q(\mathbf{r}, \mathbf{p}, t) = I[f_n(\mathbf{r}, \mathbf{p}, t), f_p(\mathbf{r}, \mathbf{p}, t)], \quad (1)$$

where $q = \{p, n\}$, with p indicating protons and n neutrons, and I is the collision integral which incorporates the Pauli blocking. In this equation the self-consistent mean-field contains an isoscalar part of the form

$$U_{is,q}(\rho) = A \frac{\rho}{\rho_0} + B \frac{\rho^\gamma}{\rho_0} \quad (2)$$

and an isovector part of the form

$$U_{iv,q}(\rho) = C(\rho) \frac{\rho_n - \rho_p}{\rho_0} \tau_q + \frac{1}{2} \frac{\partial C}{\partial \rho} \frac{(\rho_n - \rho_p)^2}{\rho_0}, \quad (3)$$

where $\tau_q = \{-1, 1\}$ for protons and neutrons, respectively. Naturally, we have that $U_q(\rho) = U_{is,q}(\rho) + U_{iv,q}(\rho)$. In the previous equations ρ is the total density, ρ_n is the neutron density and ρ_p is the proton density.

The coefficients in the isoscalar and isovector parts of the mean-field are taken to reproduce the known features of symmetric nuclear matter: the saturation density $\rho_0 = 0.16 \text{ fm}^{-3}$, the binding energy $E/A = -16 \text{ MeV}$ and a compressibility modulus $K = 200 \text{ MeV}$. These yield in $A = -356 \text{ MeV}$, $B = 303 \text{ MeV}$ and $\gamma = 7/6$. In the present calculations we choose a coefficient $C(\rho) = 32 \text{ MeV}$ which leads to a symmetry energy around 28.3 MeV .

The Boltzmann-Vlasov equation gives the time evolution of the one-body distribution function $f(\mathbf{r}, \mathbf{p}, t)$ of a fermionic system in the presence of a mean-field $U(\mathbf{r})$ and two-body collisions. For the numerical solution of the equation a convenient procedure is to decompose the distribution function f into a sum of N exponentials with respect to \mathbf{r} and \mathbf{p} :

$$f(\mathbf{r}, \mathbf{p}, t) = \frac{1}{N} \frac{1}{(2\pi\hbar)^3} \frac{1}{(4\pi\chi\phi)^{3/2}} \times \sum_i^N \exp\left(-\frac{(\mathbf{r} - \mathbf{r}_i(t)^2)}{2\chi}\right) \exp\left(-\frac{(\mathbf{p} - \mathbf{p}_i(t)^2)}{2\phi}\right). \quad (4)$$

Each exponential describes a so-called test particle through its position r_i and momentum p_i . The derivatives of the distribution function with respect to \mathbf{r} , \mathbf{p} and t are given bellow

$$\begin{aligned} \nabla_{\mathbf{r}} f(\mathbf{r}, \mathbf{p}, t) &= \frac{1}{N} \frac{1}{(2\pi\hbar)^3} \frac{1}{(4\pi^2\chi\phi)^{3/2}} \\ &\times \sum_i^N \left[-\frac{(\mathbf{r} - \mathbf{r}_i)}{\chi} \right] \exp\left(-\frac{(\mathbf{r} - \mathbf{r}_i)^2}{2\chi}\right) \exp\left(-\frac{(\mathbf{p} - \mathbf{p}_i)^2}{2\phi}\right) \end{aligned} \quad (5)$$

$$\begin{aligned} \nabla_{\mathbf{p}} f(\mathbf{r}, \mathbf{p}, t) &= \frac{1}{N} \frac{1}{(2\pi\hbar)^3} \frac{1}{(4\pi^2\chi\phi)^{3/2}} \\ &\times \sum_i^N \left[-\frac{(\mathbf{p} - \mathbf{p}_i)}{\phi} \right] \exp\left(-\frac{(\mathbf{r} - \mathbf{r}_i)^2}{2\chi}\right) \exp\left(-\frac{(\mathbf{p} - \mathbf{p}_i)^2}{2\phi}\right) \end{aligned} \quad (6)$$

$$\begin{aligned} \frac{\partial}{\partial t} f(\mathbf{r}, \mathbf{p}, t) &= \frac{1}{N} \frac{1}{(2\pi\hbar)^3} \frac{1}{(4\pi^2\chi\phi)^{3/2}} \\ &\times \sum_i^N \left[\frac{(\mathbf{r} - \mathbf{r}_i)}{\chi} \frac{\partial \mathbf{r}_i}{\partial t} \exp\left(-\frac{(\mathbf{r} - \mathbf{r}_i)^2}{2\chi}\right) \exp\left(-\frac{(\mathbf{p} - \mathbf{p}_i)^2}{2\phi}\right) \right. \\ &\left. + \frac{(\mathbf{p} - \mathbf{p}_i)}{\phi} \frac{\partial \mathbf{p}_i}{\partial t} \exp\left(-\frac{(\mathbf{r} - \mathbf{r}_i)^2}{2\chi}\right) \exp\left(-\frac{(\mathbf{p} - \mathbf{p}_i)^2}{2\phi}\right) \right] \end{aligned} \quad (7)$$

The difficult part of the Boltzmann-Vlasov equation concerns the mean-field $U(\mathbf{r})$ for which one has to calculate the derivative with respect to \mathbf{r} . We note that the mean field depends implicitly on the distribution function f through the dependence on the local densities of the nucleons (protons and neutrons). The average density is given as

$$\begin{aligned} \langle \rho \rangle &= \int d^3r d^3p \rho(\mathbf{r}) f(\mathbf{r}, \mathbf{p}, t) = \frac{1}{N} \frac{1}{(2\pi\hbar)^3} \sum_i^N \langle \rho(\mathbf{r}) \rangle_i \\ &= \frac{1}{N^2} \frac{1}{(2\pi\hbar)^6} \sum_{i,j}^N g_{2\chi}(\mathbf{r}_i - \mathbf{r}_j), \end{aligned} \quad (8)$$

and one also uses the approximation

$$\langle \rho^\gamma(\mathbf{r}) \rangle_i \approx \langle \rho(\mathbf{r}) \rangle_i^\gamma. \quad (9)$$

The derivatives of the local averaged densities are given by

$$\begin{aligned}
 \frac{\partial}{\partial x_i} \langle \rho(\mathbf{r}) \rangle_i &= \frac{1}{N} \frac{1}{(2\pi\hbar)^3} \sum_j^N \frac{\partial}{\partial x_i} g_{2\chi}(\mathbf{r}_i - \mathbf{r}_j) \\
 &= \frac{1}{N} \frac{1}{(2\pi\hbar)^3} \frac{1}{(4\pi\chi)^{3/2}} \sum_j^N \frac{\partial}{\partial x_i} \exp\left(-\frac{(\mathbf{r}_i - \mathbf{r}_j)^2}{4\chi}\right) \\
 &= \frac{1}{N} \frac{1}{(2\pi\hbar)^3} \frac{1}{(4\pi\chi)^{3/2}} \sum_j^N \frac{x_j - x_i}{2\chi} \exp\left(-\frac{(\mathbf{r}_i - \mathbf{r}_j)^2}{4\chi}\right)
 \end{aligned} \tag{10}$$

$$\frac{\partial}{\partial t} \langle \rho^\gamma(\mathbf{r}) \rangle_i \approx \frac{\partial}{\partial x_i} \langle \rho(\mathbf{r}) \rangle_i^\gamma = \gamma \langle \rho^\gamma(\mathbf{r}) \rangle_i^{\gamma-1} \frac{\partial}{\partial x_i} \langle \rho(r) \rangle_i. \tag{11}$$

The derivative of the mean-field with respect to coordinate x_i is given by

$$\frac{\partial}{\partial x_i} \langle U(\rho) \rangle_i = t_0 \frac{\partial}{\partial x_i} \langle \rho(r) \rangle_i + t_3 \frac{\partial}{\partial x_i} \langle \rho(r) \rangle_i^\gamma \tag{12}$$

With the above decomposition of the distribution function and approximation for the local average density we can see that the Boltzmann-Vlasov equation reduces numerically to the following set of equations

$$\mathbf{p}_i(t + \delta t) = \mathbf{p}_i(t) - \delta t \nabla U(\mathbf{r}_i, t + \delta t/2), \tag{13}$$

$$\mathbf{r}_i(t + \delta t/2) = \mathbf{r}_i(t - \delta t/2) + \delta t \mathbf{p}_i(t)/m, \tag{14}$$

where i goes over each test particle in the system. The full details of this numerical recipe can be found in Refs. [7–11].

As a good coverage of the phase-space requires a large number of test particles, the main computational obstacles concern the memory needed for large sets of test particles and the computing time. To better quantify the impact of the number of particles on the physical results, we show in Figure 1 the imaginary part of the Fourier transform of the local neutrons and protons densities for ^{148}Sn using 1500 test particles per nucleon (blue and green curves) and 6000 test particles per nucleon (black and red curves). The main message of the figure is that an increased number of test particles takes away the numerical noise from the plots and better evidences the two collective modes which are excited, *i.e.*, the out of phase oscillations associated with the giant dipole resonance (located around 11-12 MeV) and the faint pygmy dipole resonance (located around 7.5 MeV). This increase in the quality of the numerical results comes, however, with a substantial computational cost, as the run for 6000 test particles took roughly one year, while that for 1500 took around one month using a standard workstation.

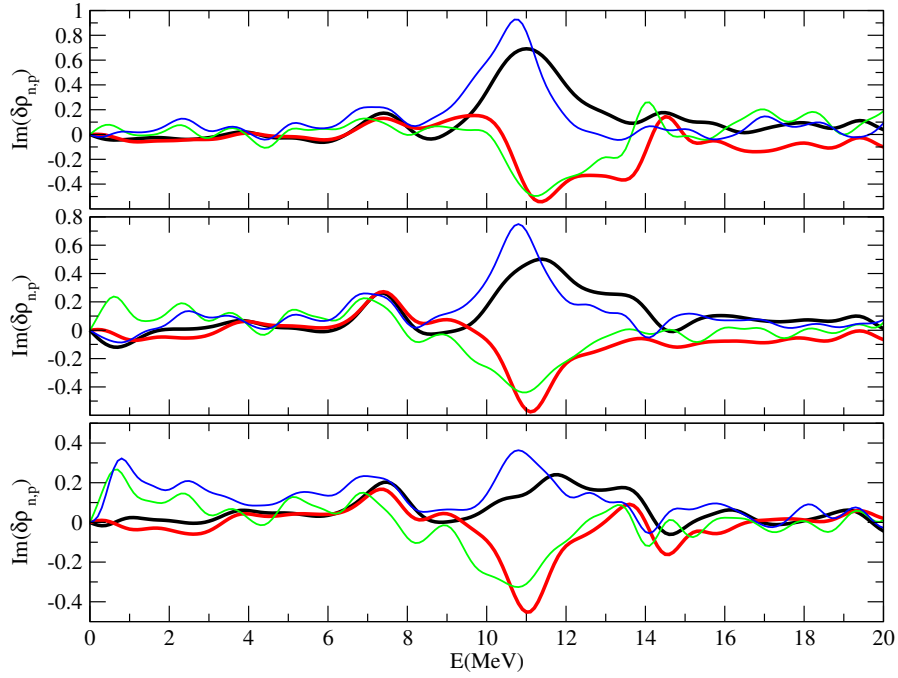


Fig. 1 – The imaginary part of the Fourier transform of the local densities of protons and neutrons for ^{148}Sn considering 1500 test particles (green and blue lines) and 6000 test particles (red and black lines). The upper panel corresponds to local densities calculated within a shell that extends from 4.5 to 5.5 fm. The middle panel corresponds to a shell that extends from 5.5 to 6.5 fm, while the lower panel corresponds to a shell between 6.5 to 7.5 fm.

3. OPENMP PARALLELIZATION

The overview of the program used to solve the two coupled Boltzmann-Vlasov equations is presented in Fig. 2. It comprises of three main parts: computing the ground-state of the system, carrying out the integration, and extracting information from the system state at specific time intervals.

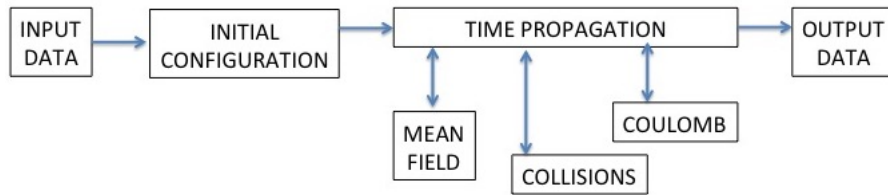


Fig. 2 – Program call graph.

For the initialization, the program reads from the input files the number of neu-

trons, protons, the width of Gaussian functions, χ and ϕ , the number of test particles per nucleon, the integration step N , the total evolution time, the time interval between subsequent CLUSTER calls, as well as the choice in isoscalar and isovector channels. Several test runs were performed to determine the optimal balance between computational speed and accuracy. The second part of the program simulates the time evolution of the system. It accomplishes this by integrating the differential equations using the Runge-Kutta method of integration at each time step. Once the new position for the Gaussians are known, the new distribution functions are reconstructed. The third part of the program computes the physical properties of interest, namely: the dipole moments, the quadrupole moments, the local densities and energies, and stores them on the corresponding files on disk.

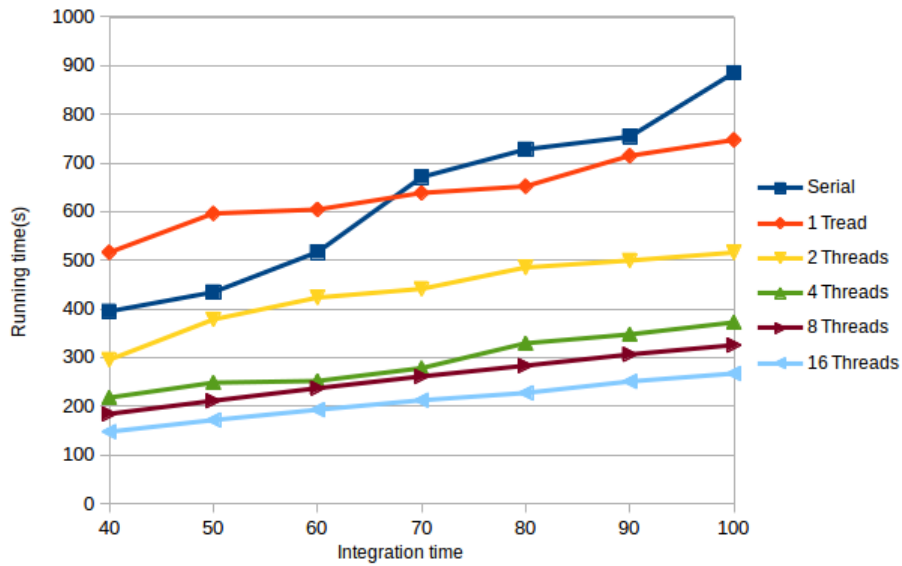


Fig. 3 – Program call graph.

An initial code analysis revealed that most of the processing (*i.e.* 65%) is happening inside the *TIME PROPAGATION* subroutine and therefore this will constitute the primary focus for code optimization. The detailed analysis of the *TIME PROPAGATION* subroutine outlined the fact that it is computationally intensive, and that I/O operations make up just an insignificant amount of time, making this subroutine an ideal candidate for parallelization. To this end, we have used an OpenMP parallel construct that involved adding the code to distribute the workload between concurrent threads. The challenging task was determining what were the shared and what were the private variables respectively. For individual values, simple PRIVATE/SHARED clauses were sufficient. For private arrays however, we allocated an array within

each thread and we deallocated it at the end of each parallel region. There were also special variables that accumulated values throughout the loop and for these cases we employed a REDUCTION clause. Data dependencies were solved by duplicating the arrays that were updated inside the inner loops. Thus, every thread reads from the original array and writes in the auxiliary array. By doing this, the original array does not change throughout the execution of the inner loop. Finally, an additional subroutine was developed, updating the values of the auxiliary array.

The next candidate for parallelization was the *COLLISIONS* subroutine, as its main loop is also computationally intensive. Given that there was no need to allocate arrays for each individual thread, another OpenMP parallel construct can be used. Test runs showed that the updated values are disjoint between threads so the data dependencies are limited to individual variables. The test runs also revealed that most of the iterations end quite fast and a very small number of iterations go through most of the instructions so another mechanism of assigning work to threads could be implemented. A good candidate would be dynamic scheduling which keeps track of the remaining workload so faster threads – having more iterations which end quickly – get more work. However, this method of scheduling adds a computational overhead which will need to be addressed in the future.

4. RESULTS AND CONCLUSIONS

As the current Boltzmann-Vlasov sequential numerical investigations are limited by long computing times, we have focused in this manuscript on an OpenMP parallelization which substantially reduces the overall running times of the codes. To this end, the computing load is distributed over multiple CPU cores, thereby reducing the computing time.

In our numerical investigations with the OpenMP parallelized code we have observed a consistent decrease of the computing time with an increasing number of threads, an aspect which is illustrated in Figs. 3 and 4. Figure 3, in particular, shows that the computing time has a linear dependence on the integration time, independent on the number of threads, while Fig. 4 shows the overall increase of the speed-up with an increasing number of threads.

The results reported in Figs. 3 and 4 can be substantially improved by a conceptual parallelization of the code which is done by spreading the computing load among several CPU cores through a direct allocation of a given region of phase-space to a given core. As the test particles that were previously assigned to the complete system now describe only a few physical particles, the number of test particles per physical particle can be substantially higher, and one can therefore simultaneously achieve a significant boost in accuracy and a reduced computational time. The main advantage of this method is that at a given iteration the computation in one cell is indepen-

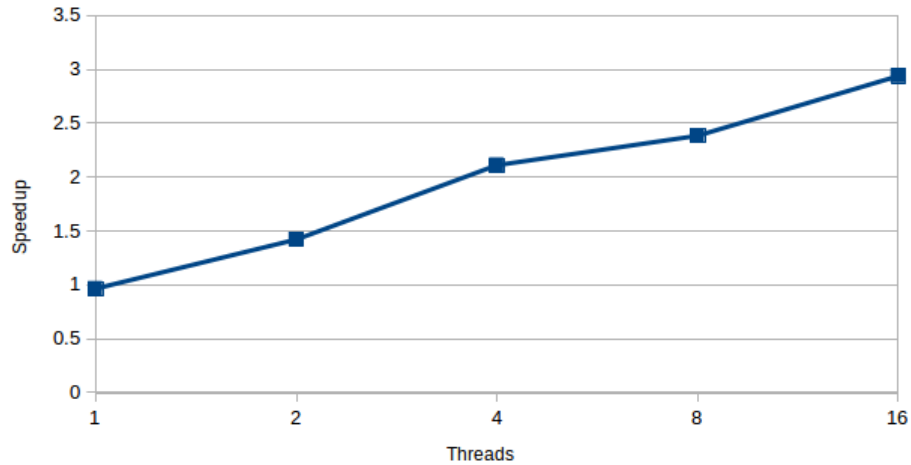


Fig. 4 – Program call graph.

dent from the computations in the other cells so, unlike parallel computing, the cores to which the cells are assigned do not need to have a joint shared memory. While the changes imposed by the parallelization of our existing source codes do not alter the overall structure of the applications, the distributed versions require conceptual modifications that make it necessary to rewrite the source codes.

The current results are of interest to the experiments at ELI-NP [12], specifically for the interpretation of results concerning the excitation of pygmy and giant dipole resonances.

Acknowledgements. M.C.R., V.B. and A.I.N. were supported by a grant from the Romanian National Authority for Scientific Research, CNCS - UEFISCDI, Project No. PN-II-ID-PCE-2011-3-0972. A.I.N. was also supported by the Romanian National Authority for Scientific Research, through the project PN 09370108/2015.

REFERENCES

1. G.F. Bertsch and S. Das Gupta, Phys. Rep. **160**, 189 (1988).
2. A. Bonasera, F. Gulminelli, and J. Molitoris, Phys. Rep. **243**, 1 (1994)
3. See sections 5 and 8 of the review V. Baran, M. Colonna, V. Greco and M. Di Toro, Phys. Rep. **410**, 335 (2005)
4. Baran, B. Frecus, M. Colonna and M. Di. Toro, Phys. Rev. C **85**, 051601 (2012).
5. Fennel Th., Meiwes-Broer K.-H., Tiggesbaumker J., Reinhard P.-G., Phuong Mai Dinh, and Eric Suraud, Rev. Mod. Phys. **82**, 1793 (2010).
6. B. Chapman, G. Jost, and R. V. D. Pas, Using OpenMP, MIT Press (2007).
7. C. Gregoire *et al.*, Nucl. Phys. A **465**, 77 (1987).
8. P. Schuck *et al.*, Prog. Part. Nucl. Phys. **22**, L81 (1989).

-
9. D. Idier, B. Benhassine, M. Farine, B. Remaud, F. Seville, Nucl. Phys. A **564**, 204 (1993).
 10. V. Baran, M. Colonna, M. Di Toro, A. Croitoru, and A. I. Nicolin, AIP Conference Proceedings **1645**, 267 (2015).
 11. V. Baran, M. Colonna, M. Di Toro *et al.*, Eur. Phys. J. D **68**, 356 (2014)
 12. The White Book of Extreme Light Infrastructure Nuclear Physics (ELI-NP), Magurele, Romania
– <http://www.eli-np.ro/documents/ELI-NP-WhiteBook.pdf>.