|  |  |
| --- | --- |
| **Review of pap479s2 by Reviewer 1** | [top](https://submissions.supercomputing.org/?args=zYprcnt3D40tprcnt3DArQbCzprcnt3DHzGMTz0Cx0zfsG_h_9aTrJUHtGAX0zU3z0IQ3TzU30CxGrQr_acz9THQP0Aprcnt3DxfGzU3ACIIfb0HQP0Aprcnt3DxfTEGcNcppTrAprcnt3DxGAX0zU3z0IQ3TtUbprcnt3DsfGQUIYprcnt3DbTzYprcnt3D40QHHGdbUfTtUbb0XfQbGztNRTzYprcnt3D40bprcnt3DQxGdbUfTzYprcnt3D40Iprcnt3Dxprcnt3DGdbUfTAprcnt3DxGPCf40zU3b0bfPz#top) |

**Summary and High Level Discussion**

The authors present hybrid MPI+OpenMP parallelization of their virtual dynamic-variable approach for simulating ions near nanoparticles by properly handling the heterogeneity of dielectric continuum. Computational parameters of the method are optimized by a machine learning (ML) approach based on neural networks to maintain the stability and accuracy of the simulation. A weak point of the paper is that no innovation in parallelization is described, and only limited scalability was achieved (Fig. 6). ML was used to define a dynamically stable computational model, prior to parallel computing itself to numerically solve the model. Thus, ML does not enable parallel computing nor it is enhanced by supercomputing.

**Comments for Revision/Rebuttal**

It would strengthen the paper if nontrivial parallel-computing innovations are introduced to boost the scalability of their simulation. Also, nontrivial integration of ML and parallel computing would make a solid supercomputing contribution, e.g., integrating the ML approach on-the-fly directly into parallel computing itself.

**Detailed Comments for Authors**

The paper is overall written well, but the related works on ML (section II.B) and parallel computing (section II.C) are too generic and lacks focus on specific ML and parallel-computing issues that are addressed in this paper. The paper presents an ingenious extended Lagrangian-based simulation method to study charged particles in inhomogeneous dielectric continuum, but its ML and parallel-computing approaches do not stand out as innovation. Key to successful ML often critically depends on the choice of an input feature vector (or descriptor) that satisfies certain conditions like the invariance with respect to rotation and particle permutation. The authors may describe their physical descriptors in more depth.  
  
While replacement of the Poisson equation by dynamic propagation of field equations is commonplace in plasma [e.g. Comput. Phys. Commun. 69, 306 (1992)] and quantum dynamics [ibid. 83, 181 (1994)] simulations, the novelty here is to replace the 3D polarization field by 2D surface charges, which drastically reduces the degrees of freedom for problems dominated by small dielectric interfaces. However, this introduces an O(n2) problem with complex interactions, where n = N (number of physical charges) + M (number of surface-charge artifacts). The current paper directly solves the O(n2) problem, which is not scalable. Implementing scalable parallel O(nlogn) or O(n) solvers for this problem would make this work an outstanding supercomputing paper.

|  |  |
| --- | --- |
| **Review of pap479s2 by Reviewer 2** | [top](https://submissions.supercomputing.org/?args=zYprcnt3D40tprcnt3DArQbCzprcnt3DHzGMTz0Cx0zfsG_h_9aTrJUHtGAX0zU3z0IQ3TzU30CxGrQr_acz9THQP0Aprcnt3DxfGzU3ACIIfb0HQP0Aprcnt3DxfTEGcNcppTrAprcnt3DxGAX0zU3z0IQ3TtUbprcnt3DsfGQUIYprcnt3DbTzYprcnt3D40QHHGdbUfTtUbb0XfQbGztNRTzYprcnt3D40bprcnt3DQxGdbUfTzYprcnt3D40Iprcnt3Dxprcnt3DGdbUfTAprcnt3DxGPCf40zU3b0bfPz#top) |

**Summary and High Level Discussion**

I found this article accurate and very well written. It presents an interesting hybrid approach built on a smart ML application and “code modernisation” via MPI/openMPI parallelism implementation. The case under study is the simulation of ions dynamics near polarizable nanoparticles. Beside the direct impact on the specific simulation problem, I think the article is a good example of how ML methods can be used to ease bottlenecks in scientific code and provide alternative solutions to otherwise “low efficiency” procedures.  
The text provides an excellent “related work” review and the authors included also an interesting outline on future development.

**Comments for Revision/Rebuttal**

The use of MLP to optimise the initialisation of the virtual system is a very interesting application and I think it could be useful to provide additional details about the hyper-parameters optimisation, the training process, etc. If the overall paper length is a concern, the authors could maybe consider shortening a bit the related work section.   
Here are some question concerning the MLP implementation:   
- First of all a generic question: how generic is the network? Once trained, can it be used for different problems or does it have to be retrained every time? i.e. the authors used it for simulating the NaCl + oil water nanoemulsion system. Assuming chemical species are changed, is the network retrained?   
- At the end of each simulation the R,Rv and fd parameter are calculated and and used “ for retraining”. Does it mean that the corresponding configuration is added to training set and used to update the MLP weights ? Or to adjust the hyper-parameteres? Could the authors provide more info?  
- - How many are the input parameters? Reading further the results section it seems to me they are 5, but I couldn’t find this info in section IV.  
- Is the MLP implemented using the mlpclsassifier in scikitlearn?   
- If I understand correctly the output is built as categorical “false/true” classification corresponding to the R,Rv,fd parameters satisfying the threshold values. What is the corresponding loss function that’s used for training ?   
- Why the choice of a tanh activation function whose output ranges in[-1,1] instead of a sigmoid for example with range [0,1]?  
- In section V.D the authors mention auto-tuning as a possible avenue of development and I think this is a very important point: Do they already have a more defined plan on tools and strategies to achieve this?   
- Food for thoughts: what other parts of the algorithm could be replaced by ML tools? Multi-objective regression approaches to replace other time consuming estimations or generative models to replace event sampling directly: there are already several applications in many different fields (High Energy physics or medical imaging)

**Detailed Comments for Authors**

Besides the comments concerning the ML part:   
The authors mention memory optimisation: would it be possible to get additional information (array layout, reducing cache calls, etc)  
What about vectorisation? Was it implemented (explicity or via compiler optimisation) in order to optimise inner loops?  
Authors mention non-uniform memory access: is their code NUMA –aware?   
Peak performance at number of physical cores: What is the hyper-threading capability of the hardware?   
Poor performance of pure MPI scaling seems due to small problem size as is mentioned in the paper: have there been tests on larger simulations (like the ones run for the hybrid model in table II)?  
Section V.C: what is the time to convergence of the hybrid system for the NaCl case? Would it be possible to compare to “standard” approaches?

|  |  |
| --- | --- |
| **Review of pap479s2 by Reviewer 3** | [top](https://submissions.supercomputing.org/?args=zYprcnt3D40tprcnt3DArQbCzprcnt3DHzGMTz0Cx0zfsG_h_9aTrJUHtGAX0zU3z0IQ3TzU30CxGrQr_acz9THQP0Aprcnt3DxfGzU3ACIIfb0HQP0Aprcnt3DxfTEGcNcppTrAprcnt3DxGAX0zU3z0IQ3TtUbprcnt3DsfGQUIYprcnt3DbTzYprcnt3D40QHHGdbUfTtUbb0XfQbGztNRTzYprcnt3D40bprcnt3DQxGdbUfTzYprcnt3D40Iprcnt3Dxprcnt3DGdbUfTAprcnt3DxGPCf40zU3b0bfPz#top) |

**Summary and High Level Discussion**

The authors discuss a framework for dynamically determining the optimal choices of parameters for Car-Parrinello molecular dynamics of a system with polarizable ions near spherical particles. The authors demonstrate the success of their code through the study of several different systems, including Na and Cl ions near a nanoparticle treated as a spherical mesh. The work is largely technically sound, aside from a few small concerns discussed below, and successfully integrates applications with an optimization framework. The primary concern from a programmatic perspective is of how much interest this work will be to the general SC audience.

**Comments for Revision/Rebuttal**

A few more details regarding the simulation protocols and choices made within them must be provided to the reader to understand what is going on, and why certain choices (in particular, the nature of the Lennard-Jones potential used for the van der Waals interactions.

**Detailed Comments for Authors**

\* The use of the “similar to” notation in Eq. 8 and the following parenthetical equation implies scale similarity in physics. The authors should use “approximately equal” instead.  
\* The use of engineering notation in Figs. 1 and 2 makes the plot harder to interpret. Use simulation time instead, since it’s already indicated in the caption.  
\* The parameter space of variables used as inputs for the 13600 runs should be identified. It would be sufficient to list the range of parameters, rather than every value used.  
\* Have the authors considered mesh-based methods such as those used in classical MD to reduce the complexity of the force and energy calculations from O(n^2) to O(n log n)? (The “Newton pairs” argument for calculating half the forces is already widely used in classical MD engines.)  
\* Most positive and negative ions have very different sizes, and therefore it is unlikely that the NaCl ions used in V.C would have the same sigma parameter for both cation and anion. This choice requires justification. Also, what was the epsilon value used for the Lennard-Jones part of the potential?  
\* There were a few small typos in the manuscript that would likely be caught by a careful spell check of the text.

|  |  |
| --- | --- |
| **Review of pap479s2 by Reviewer 4** | [top](https://submissions.supercomputing.org/?args=zYprcnt3D40tprcnt3DArQbCzprcnt3DHzGMTz0Cx0zfsG_h_9aTrJUHtGAX0zU3z0IQ3TzU30CxGrQr_acz9THQP0Aprcnt3DxfGzU3ACIIfb0HQP0Aprcnt3DxfTEGcNcppTrAprcnt3DxGAX0zU3z0IQ3TtUbprcnt3DsfGQUIYprcnt3DbTzYprcnt3D40QHHGdbUfTtUbb0XfQbGztNRTzYprcnt3D40bprcnt3DQxGdbUfTzYprcnt3D40Iprcnt3Dxprcnt3DGdbUfTAprcnt3DxGPCf40zU3b0bfPz#top) |

**Summary and High Level Discussion**

This paper describes performance enhancements to a framework for simulating ions near polarizable nanoparticles.   
  
While the paper is well-written, there are several vague points which are very important to the discussion (see revision/rebuttal comments). The details of the simulation framework are not clear and despite a lengthy related works section, it is not clear what the exact contributions of this paper are. Results section contains internal evaluations only, there are no comparisons to related software which would allow one to judge the merits of the described simulation framework.

**Comments for Revision/Rebuttal**

- Is the simulation framework using a molecular dynamics approach or is it using ab initio techniques? There are several references to CPMD, which is an ab initio code, but it is not clear what role CPMD plays in the simulation framework.   
  
- Tied to the above point, what are the interaction functions in the simulation framework? Why is an O(N^2) method being used for force calculations? If this is a molecular dynamics code, there are several well known algorithms (such as Ewald summation) to accelerate the electrostatics part.  
  
- Machine learning based optimization is claimed to be one of the main contributions of this work, but the discussion on ML is very brief and it does not contain sufficient details about how the training dataset is created, how certain parameters in the neural network were chosen, etc. Or in the first place, why does one need ML techniques here? Using the data from 13,600 simulation runs in the training dataset, can one not determine acceptable ranges for simulation parameters? What does an ML optimization framework bring?   
  
- The MPI/OpenMP parallelization techniques are relatively straight-forward, and I am not convinced that a single MPI\_Allgather after force calculations is sufficient to accurately propagate the simulation. In particular, there needs to be another MPI\_Allgather after positions are updated. Otherwise, each process will compute forces based on the original particle positions.   
  
- The related works section is unnecessarily long. It can be shortened to address some of the above points.

**Detailed Comments for Authors**

Please see comments above in the revision/rebuttal part.