

ppiclF: A Parallel Particle-In-Cell Library in Fortran

David Zwick¹

¹ Department of Mechanical & Aerospace Engineering, University of Florida

DOI: [10.21105/joss.01400](https://doi.org/10.21105/joss.01400)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Submitted: 19 April 2019

Published: 04 May 2019

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Statement of Need

Particle-in-cell (PIC) simulations are ubiquitous in science and engineering. Relevant applications range from explosively driven compressible multiphase gas-solid flow to atomistic modeling. In typical PIC simulations, individual particles are tracked as they move through a surrounding mesh, or cells. Throughout a simulation, particle-mesh and particle-particle interactions occur frequently. At the core of the particle portion of these simulations is a system of differential equations describing the physical behavior of each particle. As a result, the system given by

$$\frac{d\mathbf{Y}}{dt} = \dot{\mathbf{Y}}$$

must be solved subject to appropriate initial conditions $\mathbf{Y}(t = 0)$ and right-hand-side forcing $\dot{\mathbf{Y}}$. For PIC applications, \mathbf{Y} often includes particle coordinates and other evolving particle states, while $\dot{\mathbf{Y}}$ may be composed of appropriate force models.

While initially the given system of equations appears straightforward, computational constraints make this a daunting task when there are billions of equations, whose forcing may even depend on the states of other equations. Adding to this difficulty is the fact that the particles are often added as an afterthought when building new PIC software. As a result, typical PIC implementations may initially perform below expectations.

In order to remedy these challenges, a parallel dynamic load-balancing approach can be taken where cells and particles are distributed to different processors in memory based on some measure of overall computational workload (Liao, Ou, & Ranka, 1996). This can be accomplished through the use of modern libraries, such as Zoltan (Boman et al., 1999), where the cell and particle data is redistributed in memory. In preexisting cell-only applications though, the cells may already be evenly balanced and need not be altered. In this case, it is of practical importance that particles can unobtrusively be added to traditional cell-only codes without altering preexisting data structures, while at the same time achieving high parallel performance.

Summary

ppiclF is a parallel particle-in-cell library written in Fortran. Its main purpose is to provide a unified and scalable interface for a user to solve the above general system of differential equations in PIC simulations. The library performs on-the-fly load-balancing of the given system of equations across MPI processing ranks based on the coordinates associated with each particle. The library allows simple user input of an external overlapping mesh for interactions between particles and their nearby cells. Additionally, ppiclF includes

an optional fast binned parallel nearest neighbor search between particles within a user specified distance so that more sophisticated user-implemented right-hand-side forcing models can easily be evaluated. The algorithms have demonstrated scalability to 100,000 processors, allowing billions of equations to be solved simultaneously. The library can easily be integrated with existing cell-only codes since it does not alter data structures associated with the cells. A more complete description of ppiclF and its use is found on the documentation site (Zwick, 2019).

ppiclF was originally designed to be used by researchers in the field of multiphase flow. In fact, it has been used as a discrete element method solver coupled to both the incompressible Navier-Stokes solver NEK5000 (2019) and the compressible Euler equation solver CMT-nek (2019) to solve large-scale fluid-particle systems. Examples of these applications include fluidized beds and multiphase shock-tubes, which are detailed in the dissertation of Zwick, D. (2019). However, since the above system of differential equations are also found in many other disciplines, ppiclF's applications are not limited to multiphase flow. As a result, this library is organized such that a user is only required to set the initial conditions and forcing for the system; other details, such as the parallelization strategy, are handled internally. Thus, ppiclF's framework allows a user to easily translate their mathematical system of equations into executable code.

Acknowledgements

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1315138. This work was also supported in part by the US Department of Energy, National Nuclear Security Administration, Advanced Simulation and Computing Program, as a Cooperative Agreement under the Predictive Science Academic Alliance Program, under Contract No. DE-NA0002378.

There have been numerous suggestions and reworking of ppiclF over the past few years, including valuable input from the NEK5000 team at Argonne National Laboratory and the Center for Compressible Multiphase Turbulence team at the University of Florida.

References

- Boman, E., Devine, K., Fisk, L. A., Heaphy, R., Hendrickson, B., Leung, V., Vaughan, C., et al. (1999). Zoltan home page. Retrieved from <https://cs.sandia.gov/Zoltan>
- CMT-nek. (2019). The Center for Compressible Multiphase Turbulence at the University of Florida. Retrieved April 21, 2019, from <https://github.com/UFCCMT/Nek5000>
- Liao, W. K., Ou, C. W., & Ranka, S. (1996). Dynamic alignment and distribution of irregularly coupled data arrays for scalable parallelization of particle-in-cell problems. In *Proceedings of International Conference on Parallel Processing* (pp. 57–61). doi:10.1109/IPPS.1996.508039
- NEK5000. (2019). Argonne National Laboratory, Illinois. Retrieved April 21, 2019, from <https://nek5000.mcs.anl.gov>
- Zwick, D. (2019). A Parallel Particle-In-Cell Library in Fortran (ppiclF). Retrieved April 21, 2019, from <https://dpzwick.github.io/ppiclF-doc>
- Zwick, D. (2019). *Scalable highly-resolved Euler-Lagrange multiphase flow simulation with applications to shock tubes*. (Doctoral dissertation). University of Florida.