

PhaseFieldPet: An Open-Source Phase-Field Modeling Software for Heterogeneous Architectures

Shore Salle Chota^{1,4}, Martin Reder^{1,2}, Daniel Schneider^{1,2}, Harald Koestler⁴, and Britta Nestler^{1,2,3}

¹ Institute for Applied Materials (IAM), Karlsruhe Institute of Technology, Karlsruhe, 76131, Germany ² Institute of Digital Materials Science (IDM), Karlsruhe University of Applied Sciences, Karlsruhe, 76133, Germany ³ Institute of Nanotechnology (INT), Karlsruhe Institute of Technology, Karlsruhe, 76131, Germany ⁴ Chair for System Simulation, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany ¶ Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- Review
- Repository
- Archive

Editor: [Open Journals](#)

Reviewers:

- @openjournals

Submitted: 01 January 1970

Published: unpublished

License

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

Summary

Phase field method has emerged as a powerful computational tool for simulating complex, evolving interfaces at mesoscale in materials science and engineering, fluid dynamics, cell migration and other fields. However, achieving scalability and efficiency for parallel simulations of multicomponent multiphase systems across diverse hardware architectures remains a challenge. This paper presents PhaseFieldPet, an open-source, message passing interface (mpi) based software package for large-scale phase field simulations, specifically designed to leverage heterogeneous architectures such as CPUs and GPUs. It is built upon the Portable, Extensible Toolkit for Scientific Computation (PETSc). The software's modular design facilitates and easily integrates various phase field models such as Multiphase-Field and Multi-Order Parameter models with various choice of gradient and potential energy contributions.

Phase Field Equation

The phase-field approach represents surfaces and interfaces implicitly using continuous scalar fields (order parameter) $\phi_\alpha(\mathbf{r}, t)$, $\alpha \in \{1, 2, \dots, N\}$, which are constant in bulk phases and with smooth transition across the interface. The order parameters $\phi_\alpha(\mathbf{r}, t)$ represent different grains or phases such as solid, liquid, gas and their state like crystallographic orientation, polarization, volume fraction. For example, in a solid-solid phase transition, a solid phase can be represented by N order parameters $\phi_\alpha(\mathbf{r}, t)$ based on N crystallographic orientations or grains.

Microstructure evolution, and hence evolution of order parameter $\phi_\alpha(\mathbf{r}, t)$ can be obtained from functionals of entropy, or free energy, or grand potential (Hötzer et al., 2018). Following an energy approach, one can write the total free energy functional of the system as

$$\mathcal{F}(\phi, \nabla\phi, \dots) = \int_v f dv = \int_v f_{\text{grad}}(\phi, \nabla\phi) + f_{\text{pot}}(\phi) + f_{\text{bulk}}(\phi, \dots) dv.$$

There exists various formulations to $f_{\text{grad}}(\phi, \nabla\phi)$ and $f_{\text{pot}}(\phi)$ by many scholars in the phase field community (Daubner et al., 2023). As an example, Nestler et al. (2005) formulate these terms as

$$f_{\text{grad}}(\phi, \nabla\phi) = \varepsilon \sum_{\alpha} \sum_{\beta > \alpha} \gamma_{\alpha\beta} |\phi_{\alpha} \nabla\phi_{\beta} - \phi_{\beta} \nabla\phi_{\alpha}|^2,$$

35

$$f_{\text{pot}}(\phi) = \frac{16}{\varepsilon\pi^2} \sum_{\alpha} \sum_{\beta>\alpha} \gamma_{\alpha\beta} \phi_{\alpha} \phi_{\beta} + \frac{1}{\varepsilon} \sum_{\alpha} \sum_{\beta>\alpha} \sum_{\delta>\beta} \gamma_{\alpha\beta\delta} \phi_{\alpha} \phi_{\beta} \phi_{\delta}.$$

36 See Daubner et al. (2023) table B.4 for other formulations of these terms and their explanations
 37 currently considered in PhaseFieldPet. The bulk contribution $f_{\text{bulk}}(\phi, \dots)$ can be of various
 38 type depending on the problem at hand such as chemical, thermal, mechanical, electrical,
 39 magnetic and etc.

40 Following total energy minimization principle of the system, the phase field evolution equations
 41 for each $\phi_{\alpha}(\mathbf{r}, t)$ are in general derived by variational derivatives of the functional $\mathcal{F}(\phi, \nabla\phi, \dots)$
 42 and are given by Allen-Cahn or time-dependent Ginzburg–Landau equations for each order
 43 parameter by

$$\frac{\partial\phi_{\alpha}}{\partial t} = -L \frac{\delta\mathcal{F}}{\delta\phi_{\alpha}} = -L \left(\frac{\partial f}{\partial\phi_{\alpha}} - \nabla \cdot \frac{\partial f}{\partial\nabla\phi_{\alpha}} \right),$$

44 where L is kinetic coefficient. The resulting set of nonlinear parabolic differential equations
 45 define the multi-order parameter (MOP) phase-field model and are selected in PhaseFieldPet
 46 via option `pfe_mop`.

47
 48 Multiphase-field models restrict the phase fields such that $\phi_{\alpha}(\mathbf{r}, t) \in [0, 1]$, $\sum_{\alpha} \phi_{\alpha} = 1$.
 49 Nestler et al. (2005) introduced a Lagrange multiplier λ to restrict the summation, yielding
 50 Allen-Cahn type Phase field equation

$$\frac{\partial\phi_{\alpha}}{\partial t} = -L \left(\frac{\partial f}{\partial\phi_{\alpha}} - \nabla \cdot \frac{\partial f}{\partial\nabla\phi_{\alpha}} \right) + \lambda.$$

51 This is Lagrangian based Multiphase-field model (`mpfl`), and is chosen in PhaseFieldPet by
 52 `pfe_mpfl`. Steinbach & Pezzolla (1999) rewrote the phasefield evolution equations by the sum
 53 of binary interactions

$$\frac{\partial\phi_{\alpha}}{\partial t} = -\frac{1}{\tilde{N}\epsilon} \sum_{\beta \neq \alpha}^{\tilde{N}} M_{\alpha\beta} \left(\frac{\delta\mathcal{F}}{\delta\phi_{\alpha}} - \frac{\delta\mathcal{F}}{\delta\phi_{\beta}} \right),$$

54 where $M_{\alpha\beta}$ is a mobility matrix. This Multiphase-field model (`mpf`), is chosen in PhaseFieldPet
 55 via `pfe_mpf`.

56 We refer interested readers to Daubner et al. (2023), Moelans et al. (2008) and chapter
 57 seven of the book by Provatas & Elder (2010) for a detailed overview of various phase-field
 58 formulations and associated evolution equations.

59 Statement of need

60 For the past couple of decades, phase-field software has been being developed and used with
 61 in house codes, and open source phase field software started to be available from 2007 (Hong
 62 & Viswanathan, 2020). Many existing open source software are limited to one or two spatial
 63 dimensions, focus on binary systems, use only one type time step solver (usually explicit time
 64 stepping), work only on one CPU core (serial code) or are not capable of using heterogeneous
 65 compute resources available such as GPUs for compute and energy efficiency. Notable large
 66 scale, distributed computing capable open source phase-field software that mainly target CPUs
 67 include: MOOSE (Schwen et al., 2017), PRISMS-PF (DeWitt et al., 2020), OpenPhase
 68 (Tegeler et al., 2017), MicroSim (Dutta et al., 2025). Among proprietary, distributed machines
 69 capable software is PACE3D (Hötzer et al., 2018). Table 1 below gives a comparison of selected

70 state of the art software for Allen-Cahn (and variations thereof) type phase-field model solvers
71 with online tutorial available, able to run on distributed - large scale hardware architectures.

Software	Various Time Step Solver	3D capability	GPU capability	Phase Field Model	Spatial Discretization	Remark
PRISMS-PF	No	Yes	No	mop	FEM	Built on Deal.II
MicroSim	No	Yes*	Yes*	mop, mpfl*	FDM	* mpfl is not 3D and GPU capable yet
OpenPhase	No	Yes	No	mpf	FDM	GPU capability ongoing
PACE3D	No	Yes	No	mpfl, mpf	FDM	Proprietary
MOOSE	Yes	Yes	Yes**	mop, mpfl	FEM	**Possible for Nonlinear solver (via PETSc)
PhaseFieldPet	Yes	Yes	Yes	mop, mpfl, mpf	FDM	FEM possible

Table 1: Actively developed MPI-capable phase-field software.

72 PhaseFieldPet is a finite difference method (FDM) based software built on top of Time Step
73 (TS) solver from PETSc (Abhyankar et al., 2018). It is based on the previous work (Daubner
74 et al., 2023) including all the different model formulations compared therein. PhaseFieldPet
75 extends the numerical solutions to 3D, an arbitrary amount of N phases and includes various
76 bulk driving forces according to Hoffrogge et al. (2025). It fills the aforementioned gaps in
77 existing software by combining the following features:

- 78 1. Enabling of multiphase simulations in 1D / 2D / 3D.
- 79 2. Decoupling the numerical solution methods from the physical modeling such that one
80 can choose various solution methods without restricting to one time step solver (i.e. one
81 can use methods like semi implicit, implicit time stepping algorithms, various underlying
82 nonlinear solver, linear solvers and preconditioners, etc) based on composability features
83 of PETSc (Balay et al., 2024).
- 84 3. Executable on single core, multicore to multi node High Performance Computing clus-
85 ters/supercomputers coupled with accelerators such as GPUs (Mills et al., 2021).
- 86 4. Flexibility to easily switch between various phase-field models and energy contributions
87 at run time.

88 Usage

89 To use PhaseFieldPet, PETSc (Balay et al., 2024) needs to be installed, compiled using
90 `make PhaseFieldPet`,
91 run the executable generated with `mpiexec/mpiexec/srun` or similar other tools. For example
92 using `mpiexec` with 4 mpi processes:
93 `mpiexec -n 4 PhaseFieldPet [options]`,
94 where options can be various gradient, potential, bulk energy contributions or numerical methods
95 used to solve. For details of usage and examples, including your own energy contributions, see
96 the associated [GitHub](#) to this paper.

97 Example Result

98 [Figure 1](#) shows a PhaseFieldPet result of static triple junctions on $256 \times 256 \times 3$ grids using
99 adaptive Backward Euler (fully implicit) time step solver and `-grad_dot -pot_toth -pfe_mpfl`
100 options. See [GitHub](#) for scalability results.

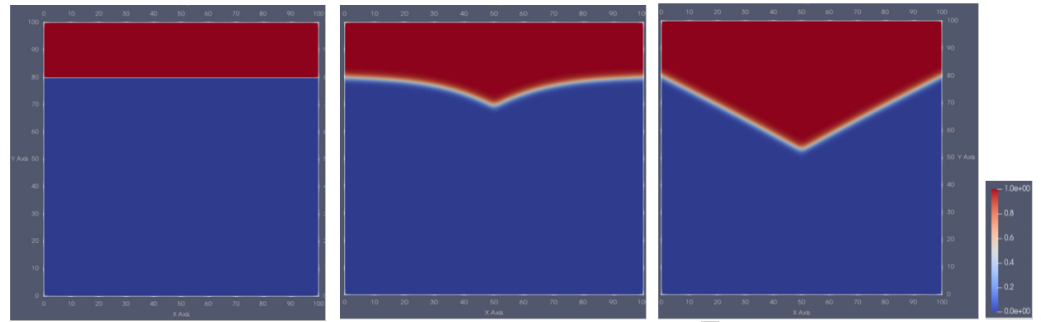


Figure 1: Time evolution of $\phi_1(\mathbf{r}, t)$: Left: Initial, Middle: Intermediate, Right: Final state.

Conclusions

PhaseFieldPet provides users with flexible methods to solve multiphase-field equations, along with various energy contributions on heterogeneous hardware architectures including CPUs and GPUs. More specifically, a user can include specific gradient, potential, bulk driving energy terms and choose the type of the phase field equation, and type of numerical algorithm to use in order to solve the differential equation. Inline with PACE3D software (Hötzer et al., 2018) and its extension, the future version of PhaseFieldPet will include various other modules corresponding to different applications of general multiphysics multiphase-field methods.

Acknowledgement

We acknowledge discussions with Dr. Simon Daubner during the genesis of this project. The authors are grateful for the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). The hardware is funded by the German Research Foundation (DFG).

Funding

The author SSC thanks the NHR-Verein e.V for supporting this work/project within the NHR Graduate School of National High Performance Computing (NHR). Furthermore, DS and BN acknowledge support from the Helmholtz association within programme MTET, no.38.04.04.

References

- Abhyankar, S., Brown, J., Constantinescu, E. M., Debojyoti Ghosh, Smith, B. F., & Zhang, H. (2018). PETSc/TS: A modern scalable ODE/DAE solver library. In *arXiv e-prints*. <https://arxiv.org/abs/1806.01437>
- Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E. M., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., ... Zhang, J. (2024). *PETSc Web page*. <https://petsc.org/>
- Daubner, S., Hoffrogge, P. W., Minar, M., & Nestler, B. (2023). Triple junction benchmark for multiphase-field and multi-order parameter models. *Computational Materials Science*, 219, 111995. <https://doi.org/10.1016/j.commatsci.2022.111995>
- DeWitt, S., Rudraraju, S., Montiel, D., Andrews, W. B., & Thornton, K. (2020). PRISMS-PF: A general framework for phase-field modeling with a matrix-free finite element method. *Npj Computational Materials*, 9, 29. <https://doi.org/10.1038/s41524-020-0298-5>

- 132 Dutta, T., Mohan, D., Shenoy, S., Attar, N., Kalokhe, A., Sagar, A., Bhure, S., Swaroop
133 S. Pradhan, SS., Praharaj, J., Mridha, S., Kushwaha, A., Shah, V., Gururajan, M. P.,
134 Venkatesh Shenoi, V., Phanikumar, G., Bhattacharyya, S., & Choudhury, A. (2025).
135 MicroSim: A high-performance phase-field solver based on CPU and GPU implementations.
136 *Computational Materials Science*, 246, 113438. [https://doi.org/10.1016/j.commatsci.2024.](https://doi.org/10.1016/j.commatsci.2024.113438)
137 [113438](https://doi.org/10.1016/j.commatsci.2024.113438)
- 138 Hoffrogge, PW., Daubner, S., Schneider, D., Nestler, B., Zhou, B., & Eiken, J. (2025).
139 Triple junction benchmark for multiphase-field models combining capillary and bulk driving
140 forces. *Modelling and Simulation in Materials Science and Engineering*, 33, 015001.
141 <https://doi.org/10.1088/1361-651X/ad8d6f>
- 142 Hong, Z., & Viswanathan, V. (2020). Open-Sourcing Phase-Field Simulations for Accelerating
143 Energy Materials Design and Optimization. *ACS Energy Letters*, 10, 3254–3259. <https://doi.org/10.1021/acsenergylett.0c01904>
144 <https://doi.org/10.1021/acsenergylett.0c01904>
- 145 Hötzer, J., Reiter, A., Hierl, H., Steinmetz, P., Selzer, M., & Nestler, B. (2018). The parallel
146 multi-physics phase-field framework Pace3D. *Journal of Computational Science*, 26, 1–12.
147 <https://doi.org/10.1016/j.jocs.2018.02.011>
- 148 Mills, R. T., Adams, M. F., Balay, S., Brown, J., & Dener, A. (2021). Toward performance-
149 portable PETSc for GPU-based exascale systems. *Parallel Computing*, 108, 102831.
150 <https://doi.org/10.1016/j.parco.2021.102831>
- 151 Moelans, N., Blanpain, B., & Wollants, P. (2008). An introduction to phase-field modeling of
152 microstructure evolution. *Calphad*, 134, 268–294. [https://doi.org/10.1016/j.calphad.2007.](https://doi.org/10.1016/j.calphad.2007.11.003)
153 [11.003](https://doi.org/10.1016/j.calphad.2007.11.003)
- 154 Nestler, B., Garcke, H., & Stinner, B. (2005). Multicomponent alloy solidification: Phase-field
155 modeling and simulations. *Physical Review E*, 71, 041609. [https://doi.org/10.1103/](https://doi.org/10.1103/PhysRevE.71.041609)
156 [PhysRevE.71.041609](https://doi.org/10.1103/PhysRevE.71.041609)
- 157 Provatas, N., & Elder, K. (2010). *Phase-field methods in materials science and engineering*.
158 Wiley-VCH Verlag GmbH & Co. KGa. <https://doi.org/10.1002/9783527631520>
- 159 Schwen, D., Aagesen, L. K., Peterson, J. W., & Tonks, M. R. (2017). Rapid multiphase-
160 field model development using a modular free energy based approach with automatic
161 differentiation in MOOSE/MARMOT. *Computational Materials Science*, 132, 36–45.
162 <https://doi.org/10.1016/j.commatsci.2017.02.017>
- 163 Steinbach, I., & Pezzolla, F. (1999). Generalized field method for multiphase transformations
164 using interface fields. *Physica D*, 134, 385–393. [https://doi.org/10.1016/S0167-2789\(99\)](https://doi.org/10.1016/S0167-2789(99)00129-3)
165 [00129-3](https://doi.org/10.1016/S0167-2789(99)00129-3)
- 166 Tegeler, M., Shchyglo, O., Kamachali, R. D., Monas, A., Steinbach, I., & Sutmann, G. (2017).
167 Parallel multiphase field simulations with OpenPhase. *Computer Physics Communications*,
168 215, 173–187. <https://doi.org/10.1016/j.cpc.2017.01.023>