SoftwareX 34 (2026) 102644

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Update (v2.0) to PeriLab - peridynamic laboratory☆

Jan-Timo Hesse[a,*], Christian Willberg[b]

[a] German Aerospace Center, Lilienthalplatz 7, 38126 Braunschweig, Germany
[b] Magdeburg-Stendal University of Applied Sciences, Breitscheidstr. 2, 39114 Magdeburg, Germany

ARTICLE INFO

Keywords:
Peridynamics
Static condensation
HPC
Computational science
MPI
Julia

ABSTRACT

This paper introduces new features for `PeriLab`, a modern Peridynamics solver developed in the Julia programming language. Emphasizing easy installation, usability, and implementation of new features, the code's structure is detailed, accompanied by illustrative examples highlighting some of the code's core functionality. Key features of the version v2.0 are the introduction of a correspondence matrix based linear static solver, the implementation of the Guyan reduction, a contact formulation and a massive code restructuring reducing the compilation time. Using the Revise.jl package is made possible by this reorganization. This makes adding new code much easier and faster.

Metadata

Code metadata

Nr.	Code metadata description	Please fill in this column
C1	Current code version	v2.0
C2	Permanent link to code/repository used for this code version	https://github.com/PeriHub/PeriLab.jl
C3	Permanent link to Reproducible Capsule	
C4	Legal Code License	BSD 3-Clause License
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	Julia
C7	Compilation requirements, operating environments & dependencies	AbaqusReader 0.2.6, ArgParse 1, CDDLib 0.10.1, CSV 0.10, Combinatorics 1, DataFrames 1, DataStructures 0.18, Dates 1, Dierckx 0.5, Exodus 0.13, FastGaussQuadrature 1, JSON3 1, LazyGrids 1, LibGit2 1, LoggingExtras 1, LoopVectorization 0.12, MPI 0.20, MPIPreferences 0.1, Meshes 0.51, NLsolve 4.5, NearestNeighbors 0.4, OrderedCollections 1, PointNeighbors 0.4, Polyhedra 0.8.1, PrettyTables 2, ProgressBars 1, Rotations 1, SparseArrays 1, StaticArrays 1, StyledStrings 1, TimerOutputs 0.5, Unitful 1.22, YAML 0.4, julia 1.10
C8	If available Link to developer documentation/manual	https://perihub.github.io/PeriLab.jl/stable
C9	Support email for questions	christian.willberg@h2.de; christian.willberg@dlr.de; jan-timo.hesse@dlr.de

1. Description of the software-update

`PeriLab` (Peridynamic Laboratory) is a comprehensive Julia-based software platform designed for tackling diverse computational problems in Peridynamics (PD). The package provides modular support for various peridynamic formulations, material models, damage simulations, contact algorithms, and FEM/PD coupling for researchers at all expertise levels. This update addresses critical needs in computational mechanics by improving efficiency through matrix-based formulations, adding multi-body contact capabilities for practical engineering applications, and optimizing Julia compilation times to enhance development workflows.

DOI of original article: https://doi.org/10.1016/j.softx.2024.101700
☆ Refers to: C. Willberg, J.-T. Hesse, A. Pernatii, PeriLab - Peridynamic Laboratory, SoftwareX 26 (2024). doi: 10.1016/j.softx.2024.101700 [1].
* Corresponding author.
Email addresses: jan-timo.hesse@dlr.de (J.-T. Hesse), christian.willberg@h2.de (C. Willberg).

https://doi.org/10.1016/j.softx.2026.102644
Received 24 February 2026; Received in revised form 27 March 2026; Accepted 4 April 2026

Glossary

BB	Bond-based
FEM	Finite Element Method
NOSB	Non-ordinary state-based
PD	Peridynamics

Since v1.3.6, the focus has been to improve efficiency and include contact algorithms. In order to reduce computational effort, it is desirable to structure the problem in terms of global stiffness matrices, similar to the Finite Element Method (FEM). For the simple Bond-based (BB) [2] and the Non-ordinary state-based (NOSB) [3,4] approaches, such formulations have already been introduced. However, most implementations still use standard summation rules and do not use a matrix approach. The advantage of the matrix approach is that for a known stiffness matrix, model reduction methods can be used. In version v2.0 the Guyan static condensation is introduced, optimizing the solving process.

In `PeriLab` the implementation of matrix-based correspondence formulations includes zero-energy stabilization and a Guyan static condensation technique to enable efficient model reduction.

The code restructuring effort tackles a common challenge in Julia-based scientific computing: lengthy compilation times that hinder rapid development and testing. By reorganizing module dependencies and optimizing type structures, compilation times have been reduced from several minutes to mere seconds, dramatically improving the development workflow and user experience.

The contact algorithm implementation extends `PeriLab`'s capabilities to multi-body contact problems, a crucial requirement for many practical engineering applications. The MPI-based parallel search algorithm ensures scalability to large problems while maintaining accuracy in detecting and resolving contact constraints between PD bodies.

This paper presents the recent developments. All models and the source code are available in the linked project repository.

1.1. Matrix-based correspondence formulation and model reduction

1.1.1. Correspondence material models

The correspondence formulation in Peridynamics provides a bridge between nonlocal peridynamic theory and classical continuum mechanics by introducing approximate deformation gradients and stress measures at the material point level. The implementation in PeriLab follows the state-based peridynamic framework with stabilization techniques to mitigate zero-energy modes.

For a material point $\mathbf{x}$, the nonlocal deformation gradient $\mathbf{F}$ is computed as:

$$\mathbf{F}(\mathbf{x}) = \left[\int_{\mathcal{H}_\mathbf{x}} \omega(|\xi|)\underline{\mathbf{Y}} \otimes \xi \, dV_{\mathbf{x}'}\right] \mathbf{K}^{-1} \tag{1}$$

where $\mathcal{H}_\mathbf{x}$ is the horizon, ω is the influence function, $\underline{\mathbf{Y}}$ is the deformed bond vector state, ξ is the reference bond vector, V is the node volume, and $\mathbf{K}$ is the shape tensor:

$$\mathbf{K}(\mathbf{x}) = \int_{\mathcal{H}_\mathbf{x}} \omega(|\xi|)\xi \otimes \xi \, dV_{\mathbf{x}'}. \tag{2}$$

The implementation assembles the system in matrix form, enabling efficient solution strategies and the application of standard linear algebra techniques. The global stiffness matrix $\mathbf{K}_{\text{global}}$ is assembled from bond-level contributions:

$$\mathbf{K}_{\text{global}} = \sum_{\text{bonds}} \mathbf{K}_{\text{bond}} \tag{3}$$

where each bond stiffness contribution accounts for the constitutive response derived from the stress-strain relationship in the correspondence framework.

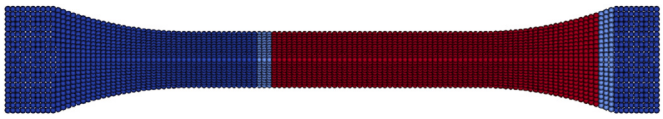


Fig. 1. Visualization of different node types in a dogbone model – PD nodes (blue), master nodes (light blue) and condensed nodes (red). (For interpretation of the references to colour in this Figure legend, the reader is referred to the web version of this article.)

1.1.2. Guyan static condensation for model reduction

For problems involving localized damage or failure, maintaining full PD resolution across the entire domain is computationally expensive. Guyan reduction [5] provides an effective strategy for reducing the system size while preserving accuracy in regions of interest.

The displacement field is partitioned into master degrees of freedom $\mathbf{u}_m$ and condensed degrees of freedom $\mathbf{u}_s$:

$$\begin{bmatrix} \mathbf{K}_{mm} & \mathbf{K}_{ms} \\ \mathbf{K}_{sm} & \mathbf{K}_{ss} \end{bmatrix} \begin{bmatrix} \mathbf{u}_m \\ \mathbf{u}_s \end{bmatrix} = \begin{bmatrix} \mathbf{f}_m \\ \mathbf{f}_s \end{bmatrix} \tag{4}$$

Assuming quasi-static conditions in the condensed region ($\mathbf{f}_s = \mathbf{0}$), the condensed system becomes:

$$\mathbf{K}_{\text{reduced}}\mathbf{u}_m = \mathbf{f}_m \tag{5}$$

where the reduced stiffness matrix is:

$$\mathbf{K}_{\text{reduced}} = \mathbf{K}_{mm} - \mathbf{K}_{ms}\mathbf{T} \tag{6}$$

with

$$\mathbf{T} = \mathbf{K}_{ss}^{-1}\mathbf{K}_{sm} \tag{7}$$

It is also necessary to reduce the mass matrix for dynamic problems. Due to the fact that the mass matrix is diagonal in the used PD formulation, the partition simplifies to:

$$\mathbf{M}_{\text{reduced}} = \mathbf{M}_{mm} + \mathbf{T}^T\mathbf{M}_{ss}\mathbf{T} \tag{8}$$

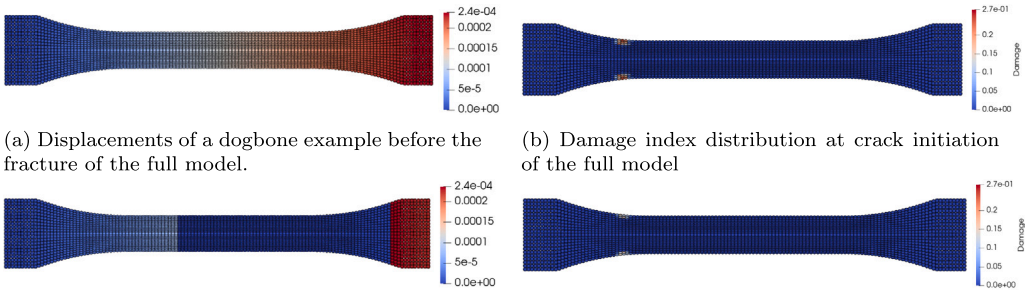
The reduced dynamic system then takes the form:

$$\mathbf{M}_{\text{reduced}}\ddot{\mathbf{u}}_m + \mathbf{K}_{\text{reduced}}\mathbf{u}_m = \mathbf{f}_m \tag{9}$$

The implementation in PeriLab allows the classical point based approach used in many publications and software implementations [6]. Parts of the reduced matrix are substituted by this approach. Fig. 1 shows this approach. The master nodes are displayed in blue, while the condensed nodes are highlighted in red. The blue region is subdivided into a matrix and a non-matrix component. The light blue nodes are the remaining master matrix nodes. They must contain at least all neighbors of the material points of the PD point based matrix free approach.

Fig. 2 shows an example of a dogbone under tension. As expected, the fracture occurs near the left radius. Displacement wise both the full and reduced models are in good agreement. Due to the choice of reduction, the number of degrees of freedom is halved, approximately doubling the computation speed.

However, the fracture of the reduced model is initiated at a later stage in the simulation. The point of origin and the pattern are similar. However, the chosen critical stretch value is very small, and the condensation region is next to the zone where fracture could occur. A small critical stretch for the damage model leads to a fast crack initiation due to the dynamics of the dogbone. It starts to swing because no energy is dissipated. The assumption of the static condensation is that quasi-static behavior is dominant. Future research will be conducted to analyze how to improve the design of the condensation region.



(a) Displacements of a dogbone example before the fracture of the full model.

(b) Damage index distribution at crack initiation of the full model

(c) Displacements of a dogbone example before the fracture of the condensed model.

(d) Damage index distribution at crack initiation of the condensed model

Fig. 2. Dogbone example.

1.2. Code restructuring

The growing number of modules and features in PeriLab has led to a notably longer compilation time, which in turn slows down the development iterations. To address this, the code base was restructured for the new major release.

The first optimization eliminated redundant imports of identical modules. In Julia, a hierarchical namespace can be defined so that a module is loaded only once and then passed to the components that need it. For example, the material helper module is now imported at the factory level and propagated to the individual material modules, rather than being imported in each module separately. This change not only cut compilation time but also enabled the use of the Revise package, which allows source code modifications to take effect without restarting the Julia session.

A similar approach was applied to the data manager. Previously the data-manager module, responsible for storing all field data and providing access to it, was imported and instantiated at the top level and then passed to sub-modules as an argument. This introduced unnecessary overhead. In the new design, modules that require the data manager directly share a single instance, removing the repeated initialization and further shortening compilation.

In addition to this, another focal point of the v2.0 release is to increase the overall type stability. While the Julia language does not require a strict type definition, it is a very important topic for high performance codes. Especially the definition of function arguments needed refinement. While strictly typed interfaces won't improve performance directly, they reduce the overall compilation effort, improve readability and also allow for multiple dispatches.

1.2.1. Results

The restructuring effort achieved a dramatic reduction in compilation time from approximately 180–300 s to 5–15 s for typical simulation setups, representing a speedup of 15–30×. In addition to this and in combination with the use of Revise, many modifications in the code base can now be tested in the same Julia session. This improvement significantly enhances the development experience and makes `PeriLab` more accessible for interactive and exploratory usage.

1.3. Contact algorithm

A bond-based contact formulation can now be utilized in the new major release of `PeriLab`. The contact stiffness $c_{contact}$ and radius $r_{contact}$ properties are defined in the YAML input file and can be adopted if needed. Based on the horizon δ, the penalty stiffness is computed as

$$c_{penalty} = \begin{cases} \frac{9}{\pi\delta^3} & \text{plane stress} \\ \frac{48}{5\pi\delta^3} & \text{plane strain} \\ \frac{12}{\pi\delta^4} & \text{three – dimension} \end{cases} \tag{10}$$

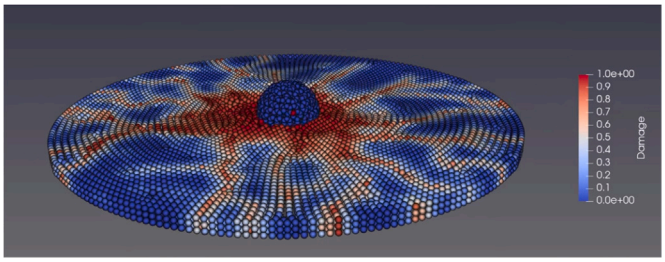


Fig. 3. Disk impact model taken from `Peridigm` and simulated with `PeriLab` v2.0.

These parameters are comparable to the bond-based formulation. The equations were taken from [6,7].

The distance d is computed using the deformed positions $\mathbf{y}$ as

$$d = |\mathbf{y}_{master} - \mathbf{y}_{cond}| \tag{11}$$

The normal is

$$\mathbf{n} = \frac{\mathbf{y}_{master} - \mathbf{Y}_{cond}}{d} \tag{12}$$

Utilizing the contact radius $r_{contact}$ the contact force densities are

$$\mathbf{f}_{master} = c_{contact} c_{penalty} (r_{contact} - d) \mathbf{n} V_{cond} \tag{13}$$

$$\mathbf{f}_{cond} = -c_{contact} c_{penalty} (r_{contact} - d) \mathbf{n} V_{master} \tag{14}$$

Fig. 3 shows an example taken from Peridigm. The model shown here is a metallic ball hitting a brittle circular plate.[1] The contact model implemented in `PeriLab` is able to reproduce the `Peridigm` results.

CRediT authorship contribution statement

Jan-Timo Hesse: Writing – review & editing, Visualization, Validation, Software, Data curation, Conceptualization. **Christian Willberg:** Writing – original draft, Visualization, Software, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge the helpful support from the `Exodus.jl` package owner https://github.com/cmhamel.

[1] Full video can be found here https://www.youtube.com/watch?v=qj7xGgmjEdE

References

[1] Willberg C, Hesse J-T, Pernatii A. PeriLab - peridynamic laboratory. Softwarex 2024;26. https://doi.org/10.1016/j.softx.2024.101700

[2] Prakash N, Stewart RJ. A multi-threaded method to assemble a sparse stiffness matrix for quasi-static solutions of linearized Bond-Based peridynamics. J Peridynamics Nonlocal Model 2021;3(2):113–47. https://doi.org/10.1007/s42102-020-00041-y

[3] Bode T, Weißenfels C, Wriggers P. Peridynamic Petrov–Galerkin method: a generalization of the peridynamic theory of correspondence materials. Comput Methods Appl Mech Eng 2020;358:112636. https://doi.org/10.1016/j.cma.2019.112636. http://www.sciencedirect.com/science/article/pii/S0045782519305201.

[4] Chan W, Chen H. Modeling material length-scale effect using the second-order peridynamic material correspondence model. Int J Eng Sci 2023;189:103877. https://doi.org/10.1016/j.ijengsci.2023.103877. https://www.sciencedirect.com/science/article/pii/S002072252300068X.

[5] Guyan RJ. Reduction of stiffness and mass matrices. AIAA J 1965;3(2):380. https://doi.org/10.2514/3.2874

[6] Littlewood DJ, Parks ML, Foster JT, Mitchell JA, Diehl P. The peridigm meshfree Peridynamics code. J Peridynamics Nonlocal Model 2023. https://doi.org/10.1007/s42102-023-00100-0

[7] Guan J, Guo L. A unified bond–based peridynamic model without limitation of Poisson's ratio. Appl Math Model 2024;128:609–29. https://doi.org/10.1016/j.apm.2024.01.015. https://www.sciencedirect.com/science/article/pii/S0307904X24000143.