SoftwareX 21 (2023) 101322

Contents lists available at ScienceDirect

SoftwareX

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

Original software publication

A user material interface for the Peridynamic Peridigm framework

Christian Willberg [a,*], Jan-Timo Hesse [a], Marc Garbade [a], Martin Rädel [a], Falk Heinecke [a], Andreas Schuster [a], Anna Pernatii [b]

[a] German Aerospace Center, Lilienthalplatz 7, Braunschweig, 38126, Germany
[b] Otto von Guericke University, Universitätsplatz 2, Magdeburg, 39104, Germany

ARTICLE INFO

Article history:
Received 25 November 2022
Received in revised form 17 January 2023
Accepted 19 January 2023

Keywords:
Peridynamics
User material
UMAT
Interface
Material modeling
Damage model

ABSTRACT

User materials (UMAT) in finite element codes allow the researchers or engineers to apply their own material routines. Simple software interfaces are specified to represent the material behavior in software. In order to use these already existing and often validated models to Peridynamics a UMAT interface is presented. It allows the simplified use of already existing material routines in the peridynamic framework Peridigm. The interface is based on the Abaqus UMAT definition and allows the integration of Fortran routines directly into Peridigm. The integration of already existing UMAT routines based in Peridigm eliminates the need for redevelopment and reprogramming material models from classical continuum mechanics theory. In addition, the same material model implementations are applicable in finite element as well as peridynamic simulations. This opens up new possibilities for analysis, verification and comparison. With this interface many material routines can be reused and applied to progressive failure analysis. The source code is stored in a GitHub repository.

Code metadata

Current software version	0.1
Permanent link to executables of this version	https://github.com/ElsevierSoftwareX/SOFTX-D-22-00389
Legal Software License	BSD
Computing platforms/Operating Systems	Linux
Installation requirements & dependencies	Trilinos 13.2.0, HDF5 1.12.0, NetCDF 4.8.0, CMake 3.20.5
Zenode archive	https://zenodo.org/record/6418265
Support email for questions	christian.willberg@dlr.de; jan-timo.hesse@dlr.de

1. Motivation and significance

In engineering applications, the material behavior is usually modeled using the classic continuum mechanics. The modeling takes place with the help of partial differential equations. Due to the requirement for the spatial derivability of the displacements, this theory has its limits when describing fracture mechanisms. A possible approach to cross these limits represents the Peridynamics. Here the requirement for the spatial derivability is overcomed by using an integral formulation instead of a differential one [1–5]. In the original theory as a result [6], existing material models had to be rewritten. To improve the usability of the Peridynamic theory in 2007 the so-called correspondence formulation was developed by Silling et al. [7]. This formulation introduces a non-local integral deformation gradient which allows the use of classical continuum mechanical models in Peridynamics. The non-local deformation gradient allows the calculation of classical strain and stress measures.

Since, Peridynamics is motivated by the analysis of crack propagation processes, a mesh-free method is usually used for the numerical solving process. One of the more advanced frameworks is provided by Sandia National Labs and is called Peridigm [8,9]. The framework allows the parallelization of large scale models and has a post processing interface to the open source software ParaView.[1] Some extensions were introduced in the recent years. Within this software publication, e.g. energy based ordinary state-based damage model, anisotropy, correspondence energy damage model [10–12] will be added to Peridigm. However, the current structure of Peridigm does not allow the direct use of already

---

[*] Corresponding author.
E-mail address: christian.willberg@dlr.de (Christian Willberg).

[1] https://www.paraview.org/ access date: 07/03/2022.

https://doi.org/10.1016/j.softx.2023.101322

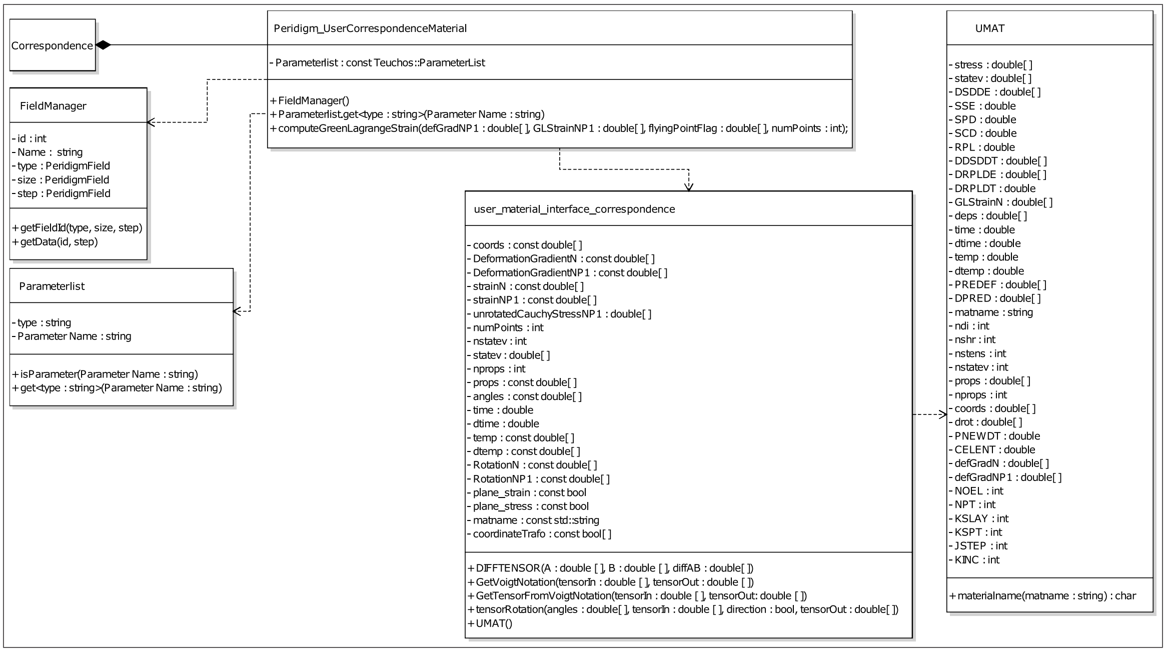


Fig. 1. UML schema.

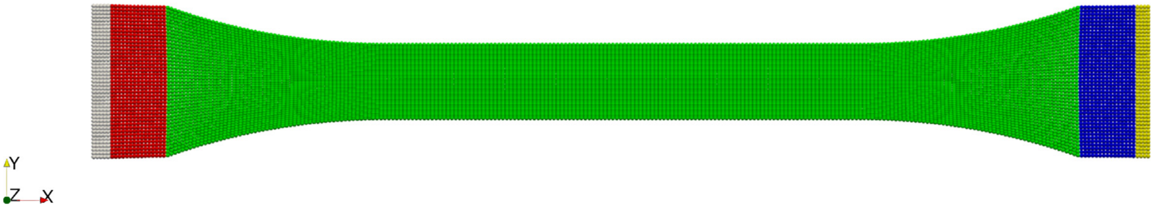


Fig. 2. Block definition for a dogbone model, which allows the assignment of properties.

existing material models. Material models have to be rewritten to use them in Peridigm. This holds true for classical peridynamic as well as correspondence material models. Abaqus is a general finite element solver that can be used to model different material behaviors. This software provides an interface to include user materials (UMAT). UMATs usually are written in Fortran and it is the quasi standard in this research domain, although alternative formats exist. Therefore, the goal of this publication is the provision of a direct Peridigm - Abaqus UMAT interface. This interface reduces the hurdle of material modeling in Peridynamics and increases the advantages of the Peridigm framework significantly. The approach presented here follows a setup that allows easy extension to other material models as well as languages.

2. Software description

In this section, we discuss the architecture and functionalities of the UMAT interface.

2.1. Software architecture

The user material interface is motivated by the Abaqus UMAT interface. Before the material routine can be used, the strain values have to be transformed into local coordinates. The stress values, which are calculated via the fortran routine, are transformed back into the original coordinates. The Table 1 gives an overview about the interface and supported parameters. Obviously not all parameters are valid, because they are specific to the finite element format.

The overall interface architecture is given in Fig. 1. The Peridigm user material is structured like a typical material used in Peridigm. In order to be able to use a material routine, the UMAT file must be precompiled and copied to a specific folder. An additional interface layer is introduced. It is used to transform or calculate specific parameters for the UMAT. Because, the material name is not transferable directly from Peridigm which is written in C++ to Fortran, a dedicated Fortran routine is provided. This routine transforms the string definition of C++ to the character field definition of Fortran.

2.2. Software functionalities

The interface to an arbitrary already existing material model allows the definition of any number of properties and state variables. The definition is shown in Listing 1 or in the example, which is given in the Zenodo archive. The state variables allow specific calculations, e.g. the history of a discrete material response or property. These state variables can be saved and requested for output in the output section of the Peridigm input file Listing 2.

Listing 1: Yaml interface to call the UMAT with name (User Material Name), three properties and three state variables.

```
Materials:
    User Material Name:
      Material Model: "User Correspondence"
      Plane Strain: false
      Plane Stress: True
```

Table 1
Interface parameter of the UMAT in Peridigm (PD). In example.yaml it is shown how to call a user material with a user defined number of properties and state variables.

Name	Type	Size	Description	Supported
sigmaNP1	double[ ]	ntens	Mechanical stresses	Yes
statev	double[ ]	nstatev	User defined state variables	Yes
DDSDDE	double[ ]	ntens × ntens	Jacobian matrix of the constitutive model $\partial\sigma/\partial\boldsymbol{\varepsilon}$	No
SSE	double	1	Specific elastic strain energy	No
SPD	double	1	Specific plastic dissipation	No
SCD	double	1	Specific creep dissipation energy	No
RPL	double	1	Volumetric heat generation per unit time	No
DDSDDT	double[ ]	ntens	Variation of the stress increments with respect to the temperature	No
DRPLDE	double[ ]	ntens	Variation of RPL with respect to the strain increment.	No
DRPLDT	double	1	Variation of RPL with respect to the temperature	No
stran	double[ ]	ntens × ntens	Strain	Yes
dstran	double[ ]	ntens × ntens	Strain increment	Yes
time(1)	double	1	Step time at the beginning of the current increment	No
time(2)	double	1	Total time at the beginning of the current increment	Yes
dtime	double	1	Time increment	Yes
temp	double	1	Temperature	Yes
dtemp	double	1	Temperature increment	Yes
PREDEF	double[ ]	–	Predefined fields	No
DPRED	double[ ]	–	Array of increments of predefined field variables	No
CMNAME	string	80	Material name	Yes
ndi	int	2 or 3	Number of direct stress components at this point	Yes
nshr	int	1 or 3	Number of engineering shear stress components	Yes
ntens	int	ndi+nshr	Size of the stress or strain component array	Yes
nstatev	int		Number of state variables	Yes
props	double[ ]	nprops	Property values	Yes
nprops	int	1	Number of properties	Yes
coords	double[ ]	2 or 3	Coordinates	Yes
drot	double[ ]	3 × 3	Rotation increment matrix	Yes
PNEWDT	double	1	Ratio of suggested new time increment	No
CELENT	double	1	Characteristic element length	No
DFGRD0	double[ ]	3 × 3	Deformation gradient N	Yes
DFGRD1	double[ ]	3 × 3	Deformation gradient N + 1	Yes
NOEL	int	1	Element number	
NPT	int	1	Integration point number	
KSLAY	int	1	Layer number	
KSPT	int	1	Section point number	No
JSTEP	int	1	Step number	No
KINC	int	1	Increment number	No

```
    Density: 2.7e+03
    Young's Modulus: 7.24e+10
    Poisson's Ratio: 3.3e-01
    Number of Properties: 3
    Prop_1: 1.0727111897390535e+11
    Prop_2: 5.2835028748341446e+10
    Prop_3: 2.721804511278195e+10
    Number of State Vars: 3
```

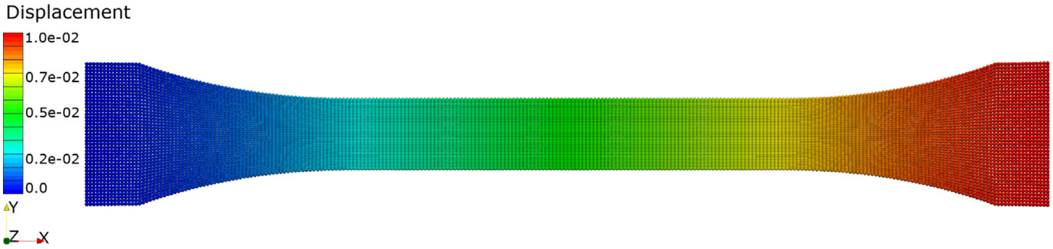
Listing 2: Yaml interface export state parameter.

```
Output:
    Output File Type: "ExodusII"
    Output Filename: "Example"
    Output Frequency: 1
    Output Variables:
      State_Parameter_Field_1: true
      State_Parameter_Field_2: true
      State_Parameter_Field_3: false
      Displacements: true
```

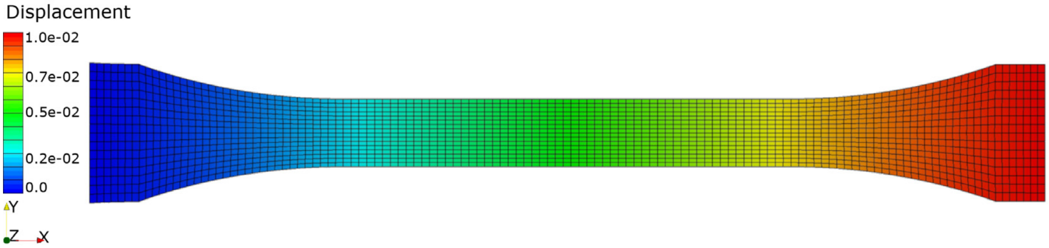
3. Illustrative examples

Fig. 2 shows the example provided in the repository in its block definition. Blocks define regions of different properties (material, horizon, damages, etc.). The user material is defined with the above properties Listing 1. Property one is the P-wave modulus, property 2 is Lames first parameter and the third property is the Shear Modulus. The dogbone is loaded under tension by applying a $u_1(x_1 = l) = 0.01\,\mathrm{m}$ displacement at the right-hand side. All translations on the boundary condition application region on the left of the specimen are fixed.

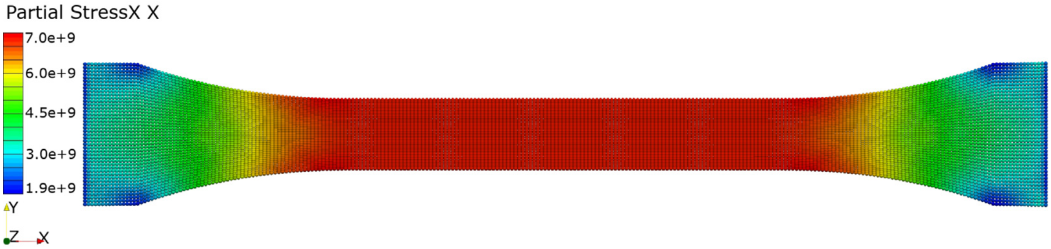
The material routine is utilized in Peridigm and Abaqus. Figs. 3(a) and 3(b) shows the resulting displacement. As expected the results are identical. This is because the u_1 displacement was applied as boundary condition. Figs. 3(c) and 3(d) illustrate the σ_{11} stress distribution. There are some differences between both results. The numerical representation between Peridigm and Abaqus is different. Boundary conditions cannot be applied in the same way, because each point in the peridynamic model represents a volume. Surface boundary conditions lead to an



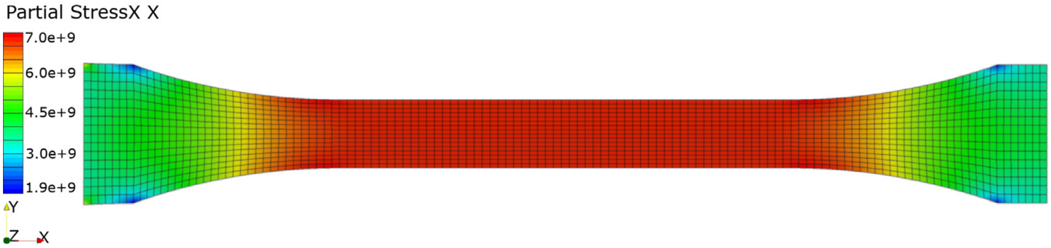
(a) Resulting displacements using Peridigm



(b) Resulting displacements using Abaqus



(c) Resulting S11 stresses using Peridigm



(d) Resulting S11 stresses using Abaqus

Fig. 3. Aluminum dogbone loaded under tension ($u_1(x_1 = l) = 0.01\,\mathrm{m}$).

error. Another difference is not of numerical nature. The classical continuum mechanics theory and Peridynamics are different formulations and also if fully converged will lead to minor differences.

4. Impact

The interface allows a simple integration of already existing material routines integrated in the finite element method. Researches have two main advantages. The first advantage is, that they can verify peridynamic modeling of complex materials very easy, because the material model is usable for two different approaches. This might help to increase the development speed. The second advantage is, that they can focus on there field of expertise. Researches do not have to understand the Peridigm code as a whole to analyze complex material model with the peridynamic approach.

The current Peridigm software without the extension is used at several universities and research institutes. To simplify the access might help to increase the user base.

5. Quality control

Multiple test routines are provided for the interface. They are based on the existing Peridigm CMake test environment. In combination with the use of CTest the user can make sure that the Fortran interface is working as expected.

Two unit tests and one functional test are implemented. The first unit test is required to ensure that all tensors are translated into a Voigt notation and returned as a full tensor. The second unit test is able to control the correct passing of variables in and from the Fortran interface. Therefore, a test user material library, which modifies every parameter by a defined value, was compiled. If the returned values are as expected the Fortran interface and the property definition is working.

In addition, full testing ensures that the user material implementation works across the Peridigm framework. Hence, the test will compare an exodus result file which is based on a predefined Peridigm material model and a file which is the result of the user material model. Both material models are similar. If the exodus results are within a defined tolerance the test will pass. As reference a dogbone model as shown in Fig. 2, the compiled fortran routine and the exodus result file can be found in the referenced Zenodo archive.

6. Conclusions

In this work, we presented a Peridigm - UMAT interface, allowing material researchers an easy access to the Peridigm software. The integrate was used integrate models which were not available in the original code. The reuse potential of already existing material routines is great. The researcher avoid the challenging implementation of their routines in the complex Peridigm software. In next steps this interface is used to analyze the difference of complex material models represented in Peridynamics and in the finite element method.

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.

Data availability

We have included the link to the software.

Acknowledgments

The authors like to acknowledge the development team of the original Peridigm framework (David J. Littlewood, djlittl@sandia.gov, John A. Mitchell, amitch@sandia.gov, Michael L. Parks, mlparks@sandia.gov, Stewart A. Silling, sasilli@sandia.gov) [9].

The work was funded by the German Research Foundation funded project: “Gekoppelte Peridynamik-Finite-Elemente-Simulationen zur Schädigungsanalyse von Faserverbundstrukturen” Grant number: WI 4835/5-1 and the M-ERA.NET, Germany funded project Exploring Multi-Method Analysis of composite structures and joints under consideration of uncertainties engineering and processing (EMMA).

This measure is co-financed with tax funds, Germany on the basis of the budget passed by the Saxon state parlament. Grant number: 3028223. The authors like to thank for the funding.

References

[1] Silling SA, Askari E. A meshfree method based on the peridynamic model of solid mechanics. Comput Struct 2005;83(17–18):1526–35. http://dx.doi.org/10.1016/j.compstruc.2004.11.026.

[2] Bobaru F, Foster JT, Geubelle PH, Silling SA. Handbook of peridynamic modeling. Advances in applied mathematics, CRC Press; 2016.

[3] Dias JP, Bazani MA, Paschoalini AT, Barbanti L. A review of crack propagation modeling using peridynamics. In: Ekwaro-Osire S, Gonçalves AC, Alemayehu FM, editors. Probabilistic prognostics and health management of energy systems. Cham: Springer International Publishing; 2017, p. 111–26. http://dx.doi.org/10.1007/978-3-319-55852-3_7.

[4] Javili A, Morasata R, Oterkus E, Oterkus S. Peridynamics review. Math Mech Solids 2018. http://dx.doi.org/10.1177/1081286518803411.

[5] Shojaei A, Hermann A, Cyron CJ, Seleson P, Silling SA. A hybrid meshfree discretization to improve the numerical performance of peridynamic models. Comput Methods Appl Mech Engrg 2022;391:114544. http://dx.doi.org/10.1016/j.cma.2021.114544, URL https://www.sciencedirect.com/science/article/pii/S0045782521007283.

[6] Silling SA. Reformulation of elasticity theory for discontinuities and long-range forces. J Mech Phys Solids 2000;48(1):175–209. http://dx.doi.org/10.1016/S0022-5096(99)00029-0.

[7] Silling SA, Epton M, Weckner O, Xu J, Askari E. Peridynamic states and constitutive modeling. J Elasticity 2007;88:151–84. http://dx.doi.org/10.1007/s10659-007-9125-1.

[8] Rädel M, Willberg C. PeriDoX. 2018, http://dx.doi.org/10.5281/zenodo.1403015, GitHub repository, URL https://github.com/PeriDoX/PeriDoX.

[9] Parks M, Littlewood D, Mitchell J, Silling S. Peridigm users’ guide. Tech. rep., Report SAND2012-7800, Sandia National Laboratories; 2012.

[10] Willberg C, Wiedemann L, Rädel M. A mode-dependent energy-based damage model for peridynamics and its implementation. J Mech Mater Struct 2019;14(2):193–217. http://dx.doi.org/10.2140/jomms.2019.14.193.

[11] Willberg C, Heinecke F. Evaluation of manufacturing deviations of composite materials. PAMM 2021;20(1):e202000345. http://dx.doi.org/10.1002/pamm.202000345.

[12] Willberg C, Hesse J-T, Heinecke F. Peridynamic simulation of a mixed-mode fracture experiment in PMMA utilizing an adaptive-time stepping for an explicit solver. J Peridyn Nonlocal Model 2022. http://dx.doi.org/10.1007/s42102-021-00079-6.