WHY DEVELOP TWICE? INTEGRATION OF CONTINUUM MECHANICAL MATERIAL MODELS IN PERIDYNAMICS

Dr.-Ing. Christian Willberg

Jan-Timo Hesse, Anna Pernatii, Prof. Dr.-Ing. Ulrich Gabbert

DLR

Introduction

- **A better understanding of damage initiation and progression leads to improved structures and avoid expensive experiments.**

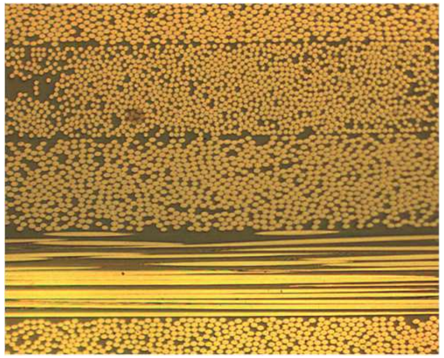
Introduction

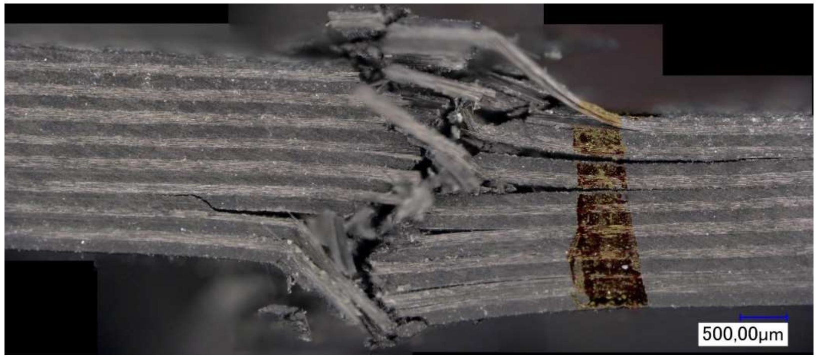
- **A better understanding of damage initiation and progression leads to improved structures and avoid expensive experiments.**
- Knowledge in
  - Material physics
  - Material and fracture modeling
  - Numerics
  - Software engineering

Classical Continuum Mechanics Theory

Fundamental assumptions

1. The medium is continuous (a continuous mass density field exists).
2. Internal forces are contact forces (material points interact only if they are separated by zero distance).
3. The deformation is twice continuously differentiable (this assumption is relaxed in the weak form of the equations).
4. The conservation laws of mechanics apply (conservation of mass, linear momentum, and angular momentum).



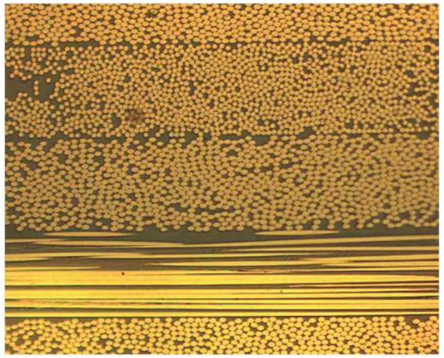


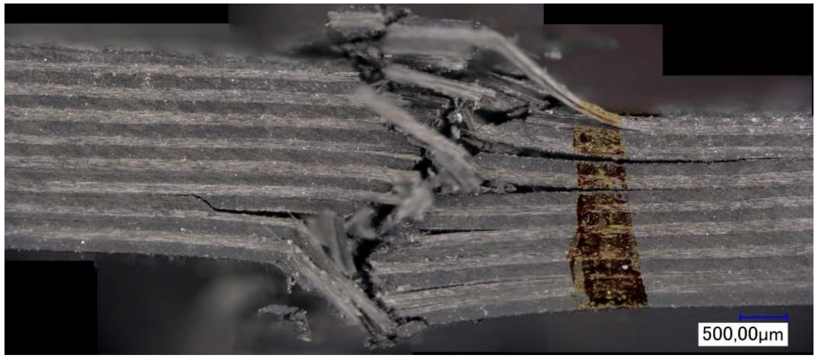
[1]Bobaru, F.; Foster, J. T.; Geubelle, P. H. & Silling, S. A. „Handbook of Peridynamic Modeling“ *CRC Press,* **2016**

Classical Continuum Mechanics Theory

Fundamental assumptions

1. ~~The medium is continuous (a continuous mass density field exists).~~
2. ~~Internal forces are contact forces (material points interact only if they are separated by zero distance).~~
3. ~~The deformation is twice continuously differentiable (this assumption is relaxed in the weak form of the equations).~~
4. The conservation laws of mechanics apply (conservation of mass, linear momentum, and angular momentum).





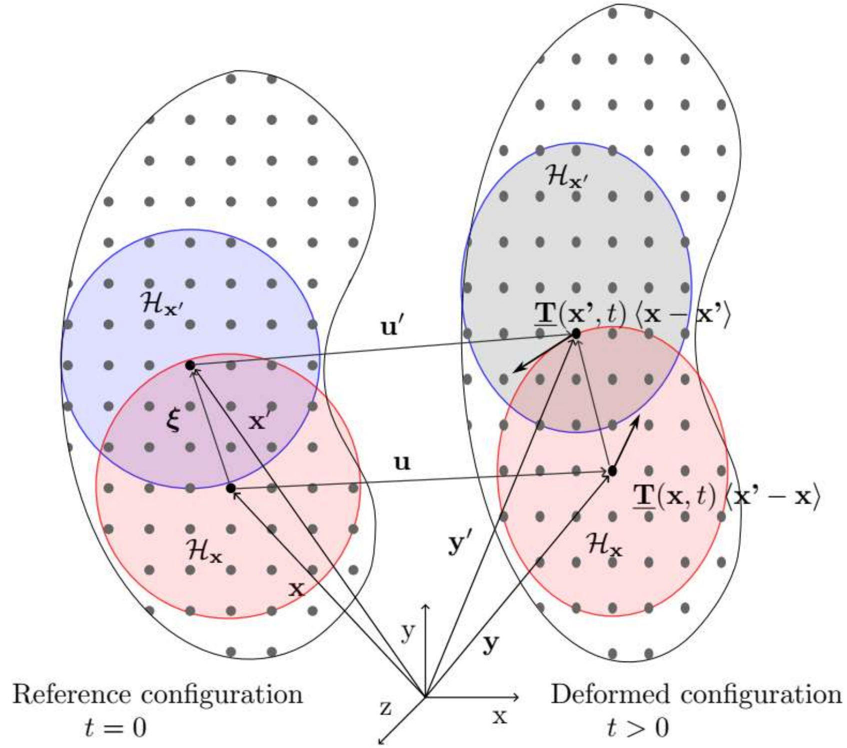
[1]Bobaru, F.; Foster, J. T.; Geubelle, P. H. & Silling, S. A. „Handbook of Peridynamic Modeling“ *CRC Press,* **2016**

Peridynamics

Formulations

$$\int_H \left[\underline{\mathbf{T}}(\mathbf{x},t)\langle\mathbf{x}'-x\rangle-\underline{\mathbf{T}}(\mathbf{x}',t)\langle\mathbf{x}-\mathbf{x}'\rangle\right]dV+\mathbf{b}=\rho\ddot{\mathbf{u}}$$

Type	conservation of momentum	conservation of angular momentum



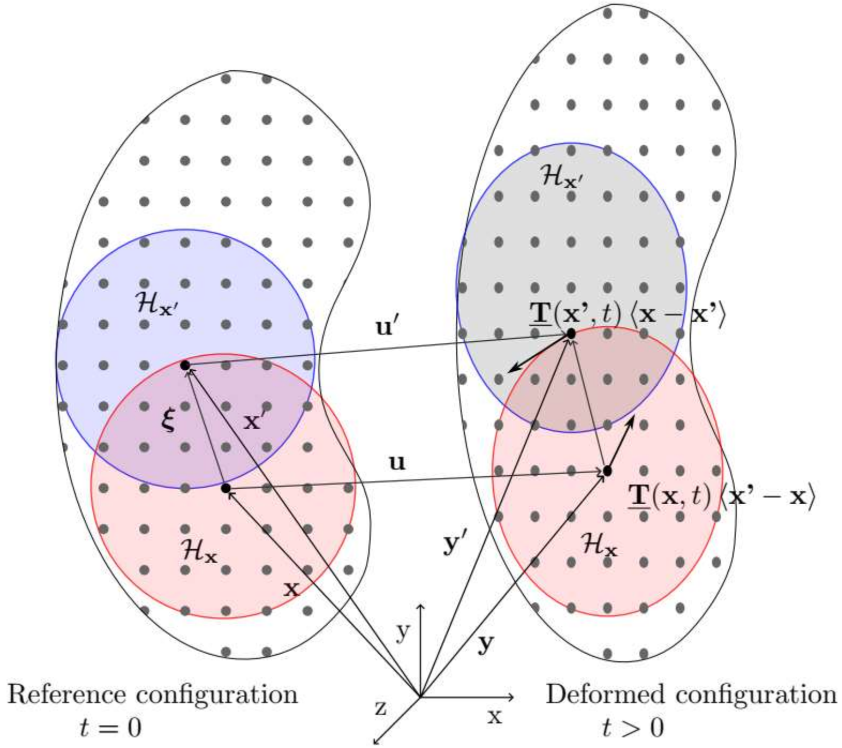
Peridynamics

Formulations

$$\int_H \left[\underline{\mathbf{T}}(\mathbf{x},t)\langle \mathbf{x}' - \boldsymbol{x}\rangle - \underline{\mathbf{T}}(\mathbf{x}',t)\langle \mathbf{x} - \mathbf{x}'\rangle\right] dV + \mathbf{b} = \rho\ddot{\mathbf{u}}$$

Type	conservation of momentum	conservation of angular momentum
Bond-based	Bond	Bond

- Only one material parameter
- Fix Poisson's ratio
- Non-local spring formulation



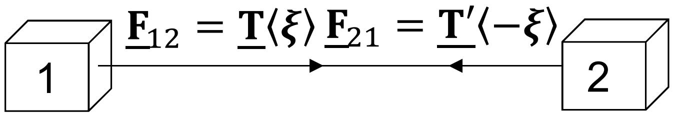
$$\mathbf{f}\left(\mathbf{u}' - \mathbf{u}, \mathbf{x}' - \mathbf{x}\right) = c\,s\,\frac{\mathbf{y}' - \mathbf{y}}{|\mathbf{y}' - \mathbf{y}|}$$

Peridynamics

Formulations

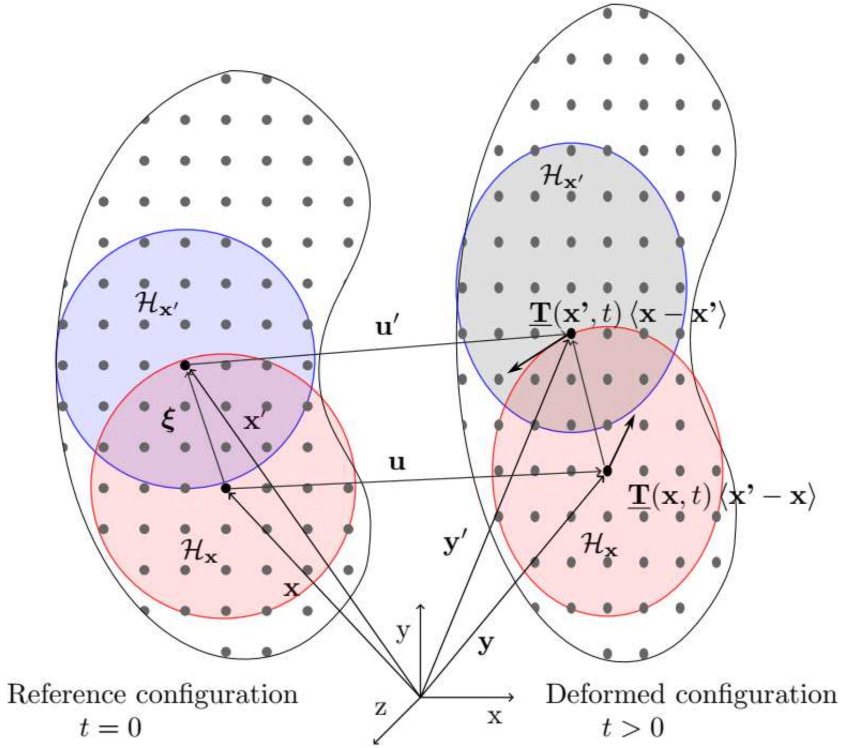
$$\int_H \left[\underline{\mathbf{T}}(\mathbf{x},t)\langle \mathbf{x}' - x\rangle - \underline{\mathbf{T}}(\mathbf{x}',t)\langle \mathbf{x} - \mathbf{x}'\rangle\right] dV + \mathbf{b} = \rho\ddot{\mathbf{u}}$$

Type	conservation of momentum	conservation of angular momentum
Bond-based	Bond	Bond
Ordinary stated based	Integral	Bond



$$\underline{t}\langle \boldsymbol{\xi}, t\rangle = \frac{\underline{\omega}\langle \boldsymbol{\xi}\rangle}{m_V}\left[3K\theta \underline{x} + 15G\underline{e}^d\right]$$

$$\underline{\mathbf{T}} = \underline{t}\frac{\underline{\mathbf{Y}}}{|\underline{\mathbf{Y}}|}$$

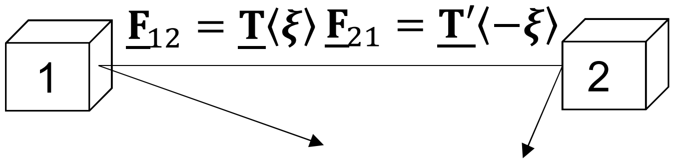


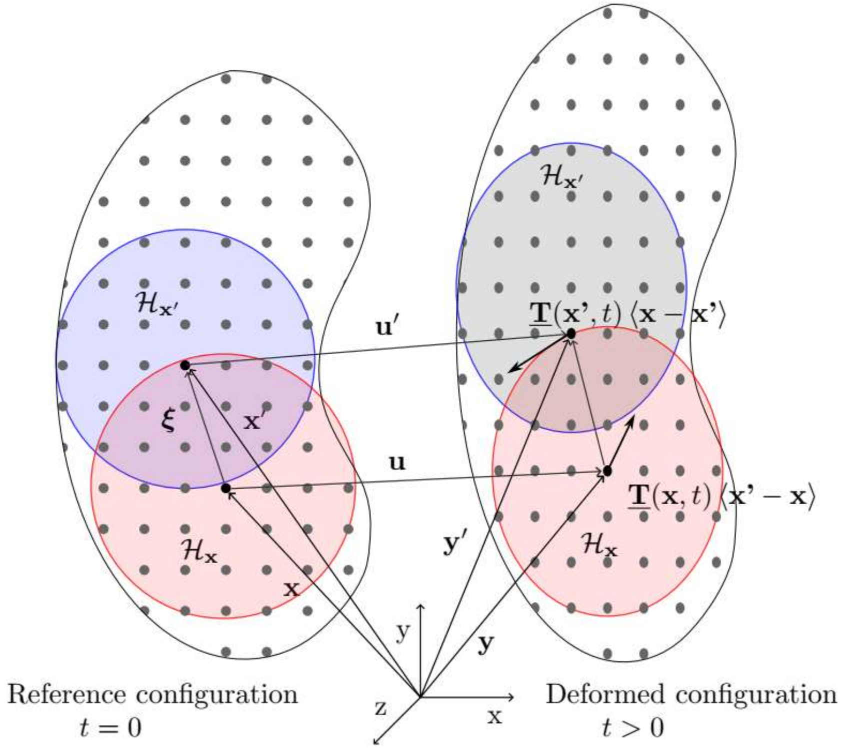
Peridynamics

Formulations

$$\int_H \left[\underline{\mathbf{T}}(\mathbf{x},t)\langle \mathbf{x}' - \boldsymbol{x}\rangle - \underline{\mathbf{T}}(\mathbf{x}',t)\langle \mathbf{x} - \mathbf{x}'\rangle\right] dV + \mathbf{b} = \rho\ddot{\mathbf{u}}$$

Type	conservation of momentum	conservation of angular momentum
Bond-based	Bond	Bond
Ordinary stated based	Integral	Bond
Non-ordinary stated based	Integral	Integral





Peridynamics

Modelling classical continuum mechanics models in Peridynamics

$$\int_{\boldsymbol{H}} \left[\underline{\mathbf{T}}(\mathbf{x},t)\langle\mathbf{x}'-\boldsymbol{x}\rangle-\underline{\mathbf{T}}(\mathbf{x}',t)\langle\mathbf{x}-\mathbf{x}'\rangle\right]dV+\mathbf{b}=\rho\ddot{\mathbf{u}}$$

$$\underline{\mathbf{T}}\langle\boldsymbol{\xi}\rangle=\underline{\omega}\langle\boldsymbol{\xi}\rangle\mathbf{P}\mathbf{K}^{-1}\boldsymbol{\xi}$$

$$\mathbf{P}=\det\mathbf{F}\boldsymbol{\sigma}\mathbf{F}^{-1}$$

$$\mathbf{F}=\left[\int_{H}\underline{\omega}\langle\boldsymbol{\xi}\rangle\,\underline{\mathbf{Y}}\langle\boldsymbol{\xi}\rangle\otimes\underline{\mathbf{X}}\langle\boldsymbol{\xi}\rangle dV\right]\mathbf{K}^{-1}$$

$$\mathbf{K}=\int_{H}\underline{\omega}\langle\boldsymbol{\xi}\rangle\,\underline{\mathbf{X}}\langle\boldsymbol{\xi}\rangle\otimes\underline{\mathbf{X}}\langle\boldsymbol{\xi}\rangle dV$$

$$\boldsymbol{\sigma}=\mathbf{C}\cdot\cdot\,\boldsymbol{\varepsilon}$$

$$\boldsymbol{\varepsilon}=\tfrac{1}{2}(\mathbf{F}^{T}\mathbf{F}-\mathbf{I})$$

$$\boldsymbol{\sigma}=\mathbf{R}\frac{\partial\sigma_{local}}{\partial\varepsilon_{local}}\mathbf{R}^{\boldsymbol{T}}\boldsymbol{\varepsilon}\mathbf{R}\mathbf{R}^{\boldsymbol{T}}$$

Peridynamics

UMAT Fortran Routine

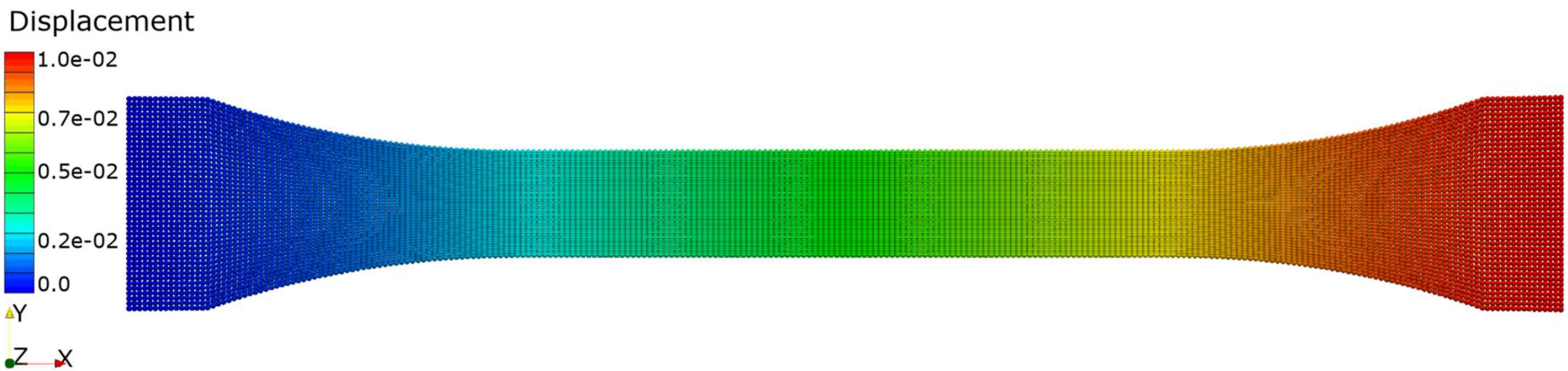
$$\mathrm{d}\boldsymbol{\sigma} = f(\boldsymbol{\varepsilon}_i, d\boldsymbol{\varepsilon}, \boldsymbol{F}_i, \boldsymbol{F}_{i+1}, \boldsymbol{status}, \boldsymbol{prop}, \ldots)$$

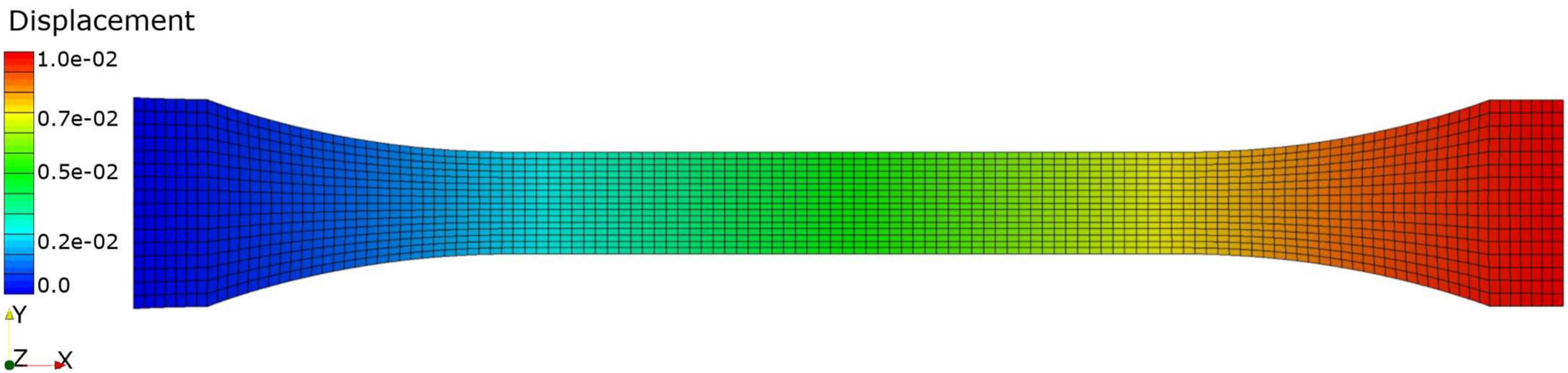
$$\boldsymbol{\sigma}_{i+1} = \boldsymbol{\sigma}_i + \mathrm{d}\boldsymbol{\sigma}$$

```fortran
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,
     2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
     3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
     4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,JSTEP,KINC)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 CMNAME
      DOUBLE PRECISION, DIMENSION(NTENS) :: STRANNP1
      DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3),
     4 JSTEP(4)

      ...

      RETURN
      END
```

Verification

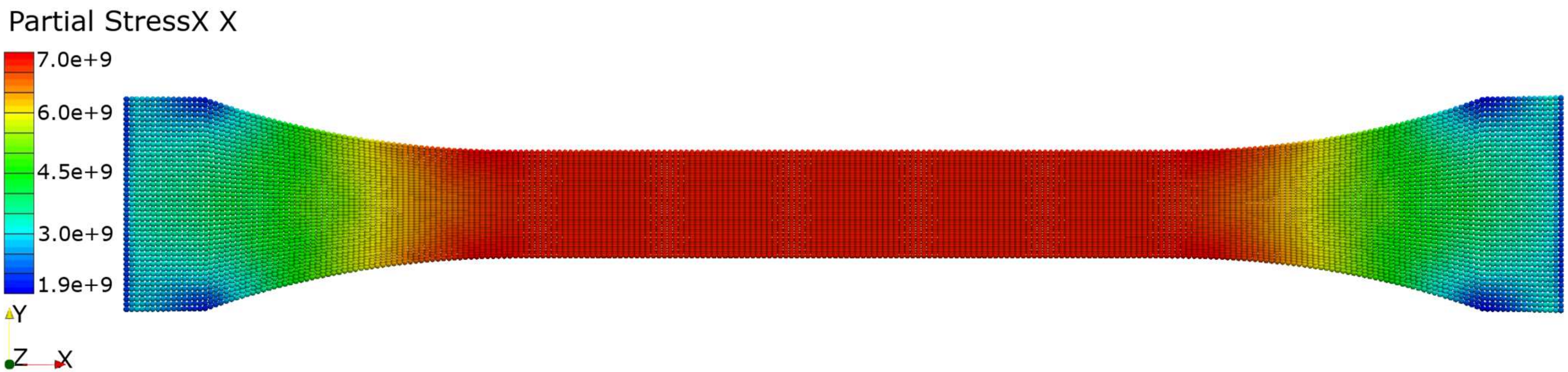
Displacements

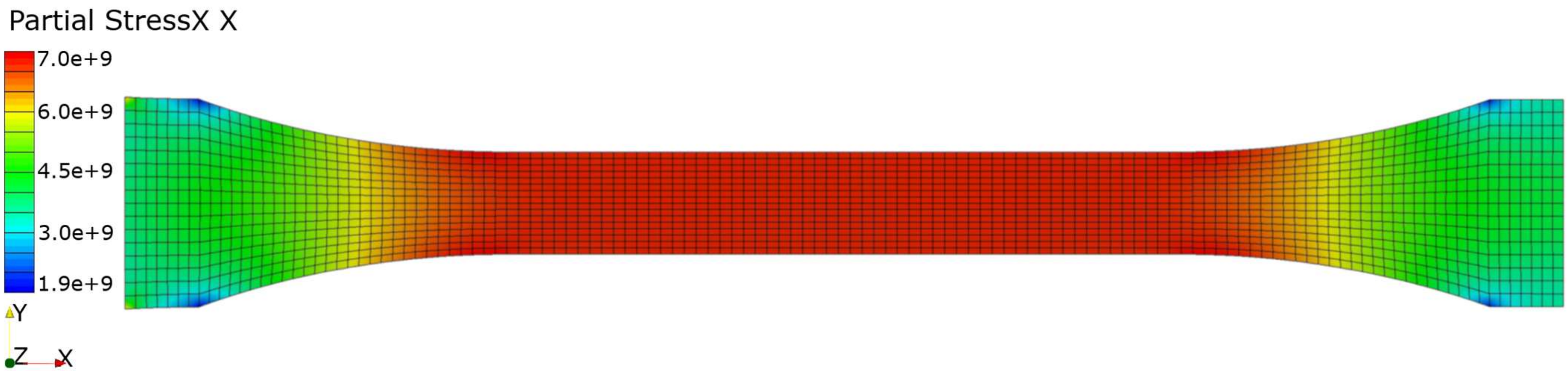




Verification

σ_{xx}





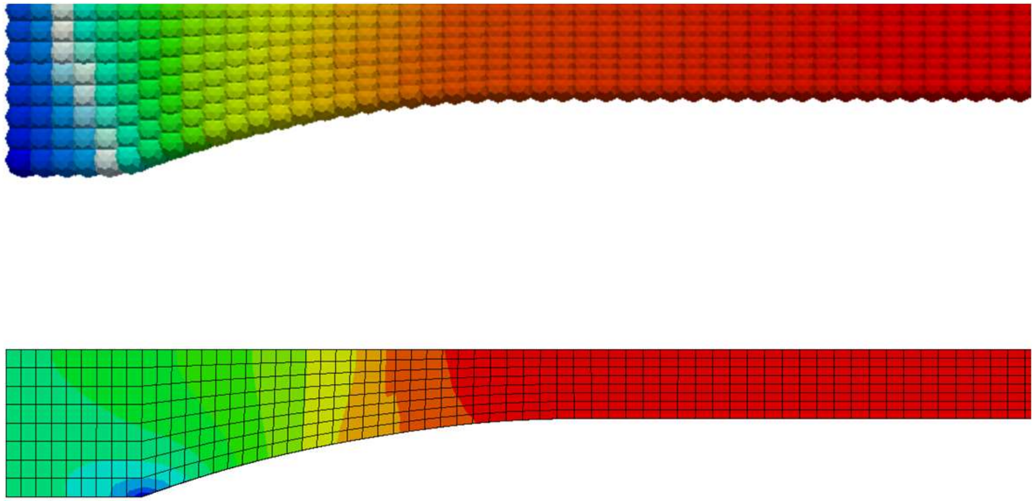
Plasticity

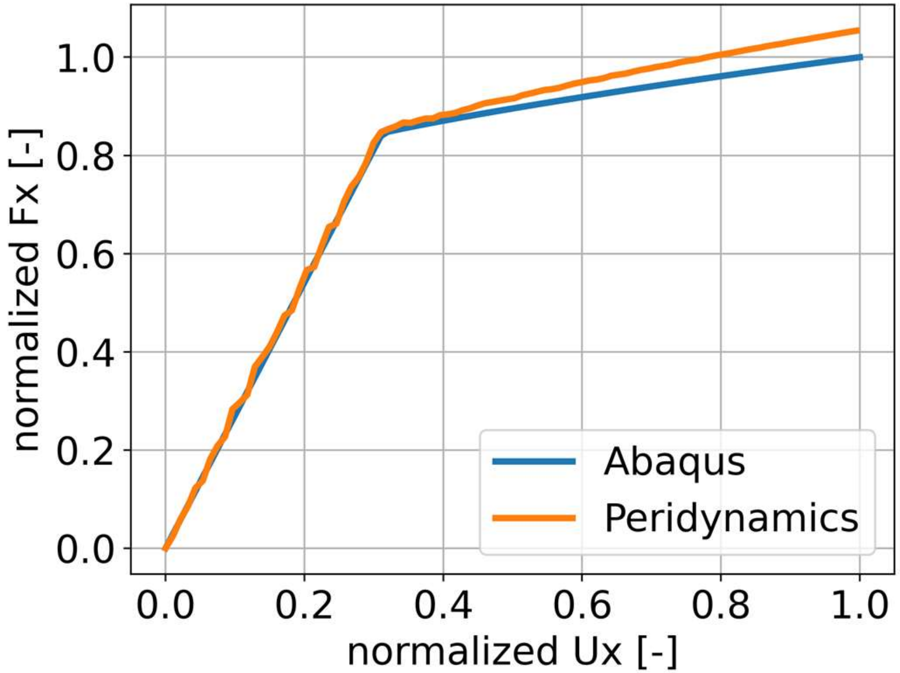
- E (Elastic modulus)
- nu (Poissons ratio)
- Y0 (Initial yield limit)
- Hiso (Isotropic hardening modulus)
- invYiso (Inverse of isotropic saturation stress)
- Hk1 (Kinematic hardening modulus nr. 1)
- invYk1 (Inverse of kinematic saturation stress nr 1)
- Hk2 (Kinematic hardening modulus nr. 2)
- invYk2 (Inverse of kinematic saturation stress nr 2)

https://github.com/KnutAM/MaterialModels
K. A. Meyer, M. Ekh, and J. Ahlström (2018) "Modeling of kinematic hardening at large biaxial deformations in pearlitic rail steel," Int. J. Solids Struct., vol. 130–131, pp. 122–132. https://doi.org/10.1016/j.ijsolstr.2017.10.007

Plasticity

σ_{xx}





https://github.com/KnutAM/MaterialModels

K. A. Meyer, M. Ekh, and J. Ahlström (2018) "Modeling of kinematic hardening at large biaxial deformations in pearlitic rail steel," Int. J. Solids Struct., vol. 130–131, pp. 122–132. https://doi.org/10.1016/j.ijsolstr.2017.10.007

Conclusion

- UMAT interface as a standard works with Peridynamics
- Complex material routines are usable
- Verification and dual use will speed up research and increase quality
- Future work:
  - Multiphysical models, e.g. for additive manufacturing

Further Reading

Willberg, Christian und Hesse, Jan-Timo und Garbade, Marc und Rädel, Martin und Heinecke, Falk und Schuster, Andreas und Pernatii, Anna (2023) *A user material interface for the Peridynamic Peridigm framework.* SoftwareX, 21. Elsevier. doi: 10.1016/j.softx.2023.101322.

Hesse, Jan-Timo und Willberg, Christian (2022) *Peridigm User Material Interface Dataset.* *d*oi: 10.5281/zenodo.6418264

Hesse, Jan-Timo und Willberg, Christian und Heinecke, Falk (2022) *Peridynamic Simulation Platform to Determine Virtual Allowables of Manufacturing Deviations.* 9th GACM Colloquium on Computational Mechanics for Young Scientists from Academia and Industrie, 21.-23. September, Essen, Germany.

Thank you!

The work was done in 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 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 on the basis of the budget passed by the Saxon state parlament. Grant number: **3028223**.

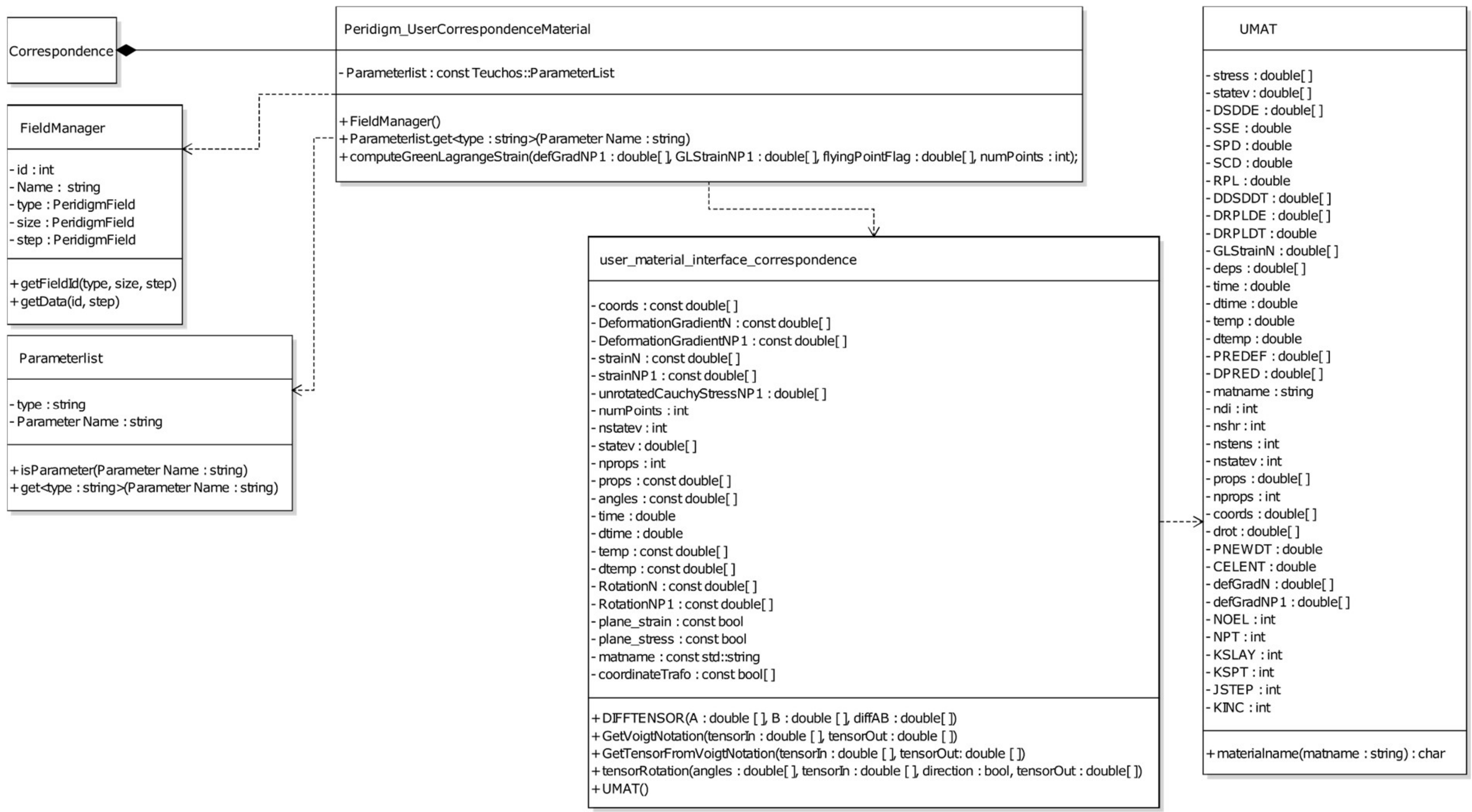
Thank you for your Attention!

Dr.-Ing. Christian Willberg

DLR
Institute of Lightweight Systems

Lilienthalplatz 7
38108 Braunschweig
Germany

Phone: +49 531 295-2489
Email: christian.willberg@dlr.de


Correspondence
Peridigm_UserCorrespondenceMaterial
- Parameterlist : const Teuchos::ParameterList
+ FieldManager()
+ Parameterlist.get<type : string>(Parameter Name : string)
+ computeGreenLagrangeStrain(defGradNP1 : double[ ], GLStrainNP1 : double[ ], flyingPointFlag : double[ ], numPoints : int);
FieldManager
- id : int
- Name : string
- type : PeridigmField
- size : PeridigmField
- step : PeridigmField
+ getFieldId(type, size, step)
+ getData(id, step)
Parameterlist
- type : string
- Parameter Name : string
+ isParameter(Parameter Name : string)
+ get<type : string>(Parameter Name : string)
user_material_interface_correspondence
- coords : const double[ ]
- DeformationGradientN : const double[ ]
- DeformationGradientNP1 : const double[ ]
- strainN : const double[ ]
- strainNP1 : const double[ ]
- unrotatedCauchyStressNP1 : double[ ]
- numPoints : int
- nstatev : int
- statev : double[ ]
- nprops : int
- props : const double[ ]
- angles : const double[ ]
- time : double
- dtime : double
- temp : const double[ ]
- dtemp : const double[ ]
- RotationN : const double[ ]
- RotationNP1 : const double[ ]
- plane_strain : const bool
- plane_stress : const bool
- matname : const std::string
- coordinateTrafo : const bool[ ]
+ DIFFTENSOR(A : double [ ], B : double [ ], diffAB : double[ ])
+ GetVoigtNotation(tensorIn : double [ ], tensorOut : double [ ])
+ GetTensorFromVoigtNotation(tensorIn : double [ ], tensorOut: double [ ])
+ tensorRotation(angles : double[ ], tensorIn : double [ ], direction : bool, tensorOut : double[ ])
+ UMAT()
UMAT
- stress : double[ ]
- statev : double[ ]
- DSDDE : double[ ]
- SSE : double
- SPD : double
- SCD : double
- RPL : double
- DDSDDT : double[ ]
- DRPLDE : double[ ]
- DRPLDT : double
- GLStrainN : double[ ]
- deps : double[ ]
- time : double
- dtime : double
- temp : double
- dtemp : double
- PREDEF : double[ ]
- DPRED : double[ ]
- matname : string
- ndi : int
- nshr : int
- nstens : int
- nstatev : int
- props : double[ ]
- nprops : int
- coords : double[ ]
- drot : double[ ]
- PNEWDT : double
- CELENT : double
- defGradN : double[ ]
- defGradNP1 : double[ ]
- NOEL : int
- NPT : int
- KSLAY : int
- KSPT : int
- JSTEP : int
- KINC : int
+ materialname(matname : string) : char