Recent applications of Ferrite.jl at the IKM

Dustin R. Jantos, Ilayda Kök,
Hendrik Geisler, Miriam Kick,
Max von Zabiensky

Institute of Continuum Mechanics,
Leibniz Universität Hannover

Leibniz Universität Hannover

06.10.2023

Ferrite.jl User & Developer Conference 2023

Overview

- complex material models in Ferrite.jl
  - time-separated **stochastic** mechanics → Geisler
  - **damage** and fatigue modelling → Kök
  - topology **optimization** with plasticity → Kick
  - topology **optimization** for large deformations → von Zabiensky

Uncertainty quantification for inelastic material models

Flight > Airlines

Up to 1,200 Airbus Jet Engines Recalled

The recall was caused by on impurities in powdered metal.

BY SÉBASTIEN ROBLIN PUBLISHED: JUL 27, 2023

BUSINESS

Pratt & Whitney Engines on Hundreds of Airbus Jets Recalled for Inspection

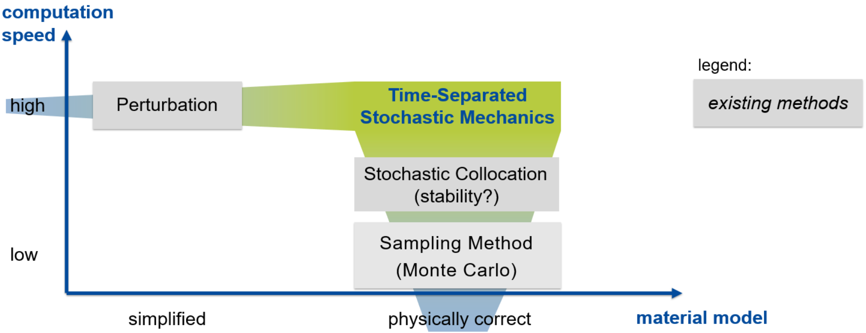
RTX says engines are affected by contaminated metal parts that could crack over time

source: prattwhitney.com

source: prattwhitney.com

How to **predict** the stochastic behavior **efficiently and accurately**?

State-of-the-art



Dustin R. Jantos, Ilayda Kök, Hendrik Geisler, Miriam Kick, Max von Zabiensky

Implementation in Ferrite

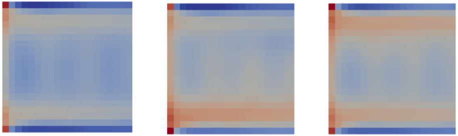
Approach: split into **deterministic** "0" and **stochastic** part "I":
$\rightarrow \mathbb{E} = \mathbb{E}^0 + \xi \mathbb{E}^I, \; \langle \xi \rangle = 0$
$\rightarrow \varepsilon = \varepsilon^0 + \xi \varepsilon^I, \quad \varepsilon^{inelas} = \varepsilon^{inelas,0} + \xi \varepsilon^{inelas,I}$

Seperate set of PDEs for deterministic and stochastic part
$\rightarrow$ Evolution equations stem naturally from standard set of equations

for each loadstep **do**
    do FEM "0" $\rightarrow \varepsilon^0, \varepsilon^{inelas,0}$
    do FEM "I" $\rightarrow \varepsilon^I, \varepsilon^{inelas,I}$
    calculate expectation & standard deviation
     of stresses, reaction forces, etc.
end for

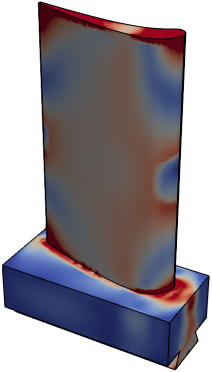
Results for standard deviation of stress

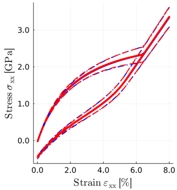
TSM	1000 MC	10000 MC

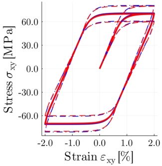


0min 17s	18min 40s	3h 3min

18h (TSM) vs.
0.7 years (MC)







TSM simulations
MC simulations (reference)

TSM speed-up > 100

→ viscoplasticity, damage, phasetransformations ...

Hamilton-based damage model (HDM)

- extended Hamilton functional e.g. for damage modelling

$$\mathcal{H}[\boldsymbol{u},\boldsymbol{\alpha}] \;:=\; \underbrace{\mathcal{G}[\boldsymbol{u},\boldsymbol{\alpha}]}_{\text{Gibbs energy}} + \underbrace{\mathcal{D}[\boldsymbol{\alpha}]}_{\text{dissipation}} - \underbrace{\mathcal{R}[\boldsymbol{\alpha}]}_{\text{hardening}} + \underbrace{\mathcal{C}[\boldsymbol{\alpha}]}_{\text{constraints}}$$

- Internal variables: $\boldsymbol{\alpha} = \{d, \varepsilon_p\}$

↳ $\Psi(\varepsilon, \boldsymbol{\alpha}) = \frac{1}{2}(\varepsilon - \varepsilon_p) : f(d)\mathbb{C} : (\varepsilon - \varepsilon_p) + \frac{1}{2}\beta \,\|\nabla f\|^2$

↳ $\mathcal{D}[\boldsymbol{\alpha}] = \int_\Omega \frac{\partial \Delta}{\partial \dot{\boldsymbol{\alpha}}} \cdot \boldsymbol{\alpha} \,\mathrm{d}V$

  ↳ $\Delta[\boldsymbol{\alpha}] = r_d|\dot{d}| + \frac{1}{2}\,\eta_d|\dot{d}|^2 + r_p\,\|\dot{\varepsilon}_p\| + \frac{1}{2}\,\eta_p\,\|\dot{\varepsilon}_p\|^2$

↳ $\mathcal{R}[\boldsymbol{\alpha}] = \int_\Omega \Psi_h \,\mathrm{d}V$ with $\Psi_h = \Psi_h(\alpha)$.

↳ $\mathcal{C}[\boldsymbol{\alpha}] = \int_\Omega p_c^p : \varepsilon_p + p_c^\alpha \,\mathrm{d}V$

Hamilton-based damage modelling (HDM)

- requiring stationarity: $\mathcal{H}\left[\boldsymbol{u}, \boldsymbol{\alpha},\right] \rightarrow \underset{\boldsymbol{u},\boldsymbol{\alpha}}{\text{stat}}$
- stationarity condition yields:

$$
\begin{cases}
\int\limits_{\Omega} \frac{\partial \Psi_m}{\partial \varepsilon}\, \delta \boldsymbol{u} \,\mathrm{dV} - \int\limits_{\Omega} \boldsymbol{b}^* \cdot \boldsymbol{u}\, \delta \boldsymbol{u} \,\mathrm{dV} - \int\limits_{\partial\Omega} \boldsymbol{t}^* \cdot \boldsymbol{u}\, \delta \boldsymbol{u} \,\mathrm{dA} & = 0 \qquad \forall\ \delta \boldsymbol{u} \\
\int\limits_{\Omega} \left(-\mathbb{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_p) + \frac{\partial \Delta}{\partial \boldsymbol{\varepsilon}_p} + \kappa \mathbf{I}\right) : \delta \boldsymbol{\varepsilon}_p \,\mathrm{dV} & = 0 \qquad \forall\ \delta \boldsymbol{\varepsilon}_p \\
\int\limits_{\Omega} \Psi_0 f' \delta d \,\mathrm{dV} - \int\limits_{\Omega} \beta \nabla f \cdot \nabla (f' \delta d) \,\mathrm{dV} + \int\limits_{\Omega} \frac{\partial \Delta}{\partial d} \,\mathrm{dA} \delta d & = 0 \qquad \forall\ \delta d
\end{cases}
$$

Hamilton-based damage modelling (HDM)

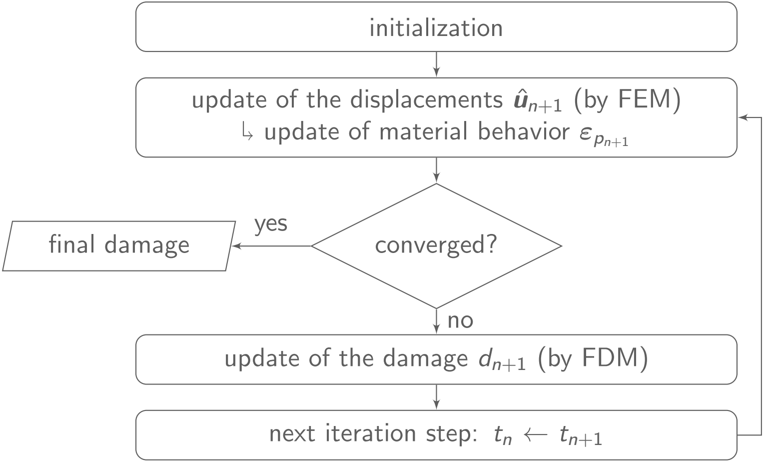
- solve system of stationary conditions
  - $\delta_{\boldsymbol{u}}\mathcal{H} = 0 \quad \Rightarrow$ weak form of balance of linear momentum
  - $\delta_{\boldsymbol{\varepsilon}_p}\mathcal{H} = 0 \quad \Rightarrow$ material model (evolution equation)

    $\Rightarrow$ solved by **finite element method** (FEM)
  - $\delta_d\mathcal{H} = 0 \quad \Rightarrow$ damage equation (strong form)

    $$\Psi_0 f(d) - \beta f(d)\nabla^2 f(d) - r - \eta\dot{d} \leq 0$$

    $\Rightarrow$ solved by **finite difference method** (FDM)
- staggered FEM + FDM $\Rightarrow$ **NEM** (neighbored element method)

Damage Process



Algorithm: update of $\boldsymbol{u}_{n+1}$ and ε_{pn+1} by FEM

while true **do**
  for each *element* $\in$ *mesh* **do**
    call reinit!(*mesh*, *elementvalues*)
    call $\boldsymbol{K}_e, \boldsymbol{r}_e$ = assembleCell!(*elementvalues*, $\boldsymbol{u}$) ▷ see next slide
    call assemble!(*assembler*, $\boldsymbol{K}_e$, $\boldsymbol{r}_e$)
  end for
  call apply_zero!($\boldsymbol{K}$, $\boldsymbol{r}$, *constraints*)
  if $\|\boldsymbol{r}\| <$ tol **then** break
  end if
  update $\boldsymbol{u}_{i+1} = \boldsymbol{u}_i - \boldsymbol{K}^{-1}\boldsymbol{r}$
  update $i = i + 1$
end while
$\boldsymbol{u}_{n+1} \leftarrow \boldsymbol{u}_{i+1}$

Algorithm: update of $\boldsymbol{u}_{n+1}$ and ε_{pn+1} by FEM

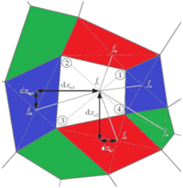
- assembleCell!($elementvalues$, $\boldsymbol{u}$)

for each $ip \in element$ **do**
  compute ε_{n+1} = function_symmetric_gradient($elementvalues$, ip, $\boldsymbol{u}_e$)
  compute $\boldsymbol{\sigma}_{n+1}(d_n)$, $\mathbb{D}_{0,n+1}$, ε_{pn+1} and $\Psi_{0,n+1} \rightarrow$ material model
  compute $\Omega^\star$ = getdetJdV($elementvalues$, ip)
  for i to number base shape functions **do**
    compute $\boldsymbol{B}^{\mathrm{T}}$ = shape_symmetric_gradient($elementvalues$, ip, i)
    compute $\boldsymbol{r}_e\,[i] \mathrel{+}= \left(\boldsymbol{B}^{\mathrm{T}} : \boldsymbol{\sigma}_{n+1}\right)\, \Omega^\star$
    for j to number base shape functions **do**
      compute $\boldsymbol{B}$ = shape_symmetric_gradient($elementvalues$, ip, j)
      compute $\boldsymbol{K}_e\,[i,j] \mathrel{+}= \left(\boldsymbol{B}^{\mathrm{T}} : \mathbb{D}_{0,n+1} : \boldsymbol{B}\right)\, \Omega^\star$
    end for
  end for
  compute $\boldsymbol{r}_e = \boldsymbol{r}_e - \boldsymbol{f}_{e,\mathrm{ext}}$
end for

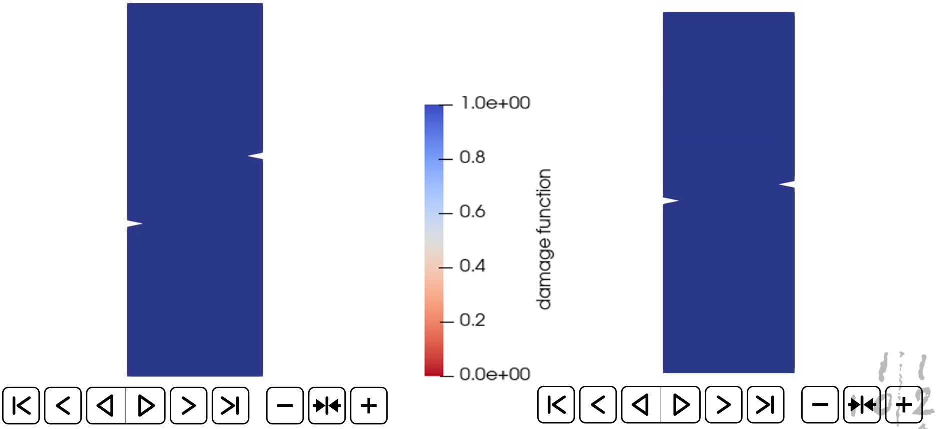
Update of d_{n+1} via "external" FDM

- input:
  - strain energy: $\Psi_{0,n+1}$ (from FEM)
  - structure volume (user input)
  - regularization parameter (user input) $\beta \rightarrow$ minimum member size

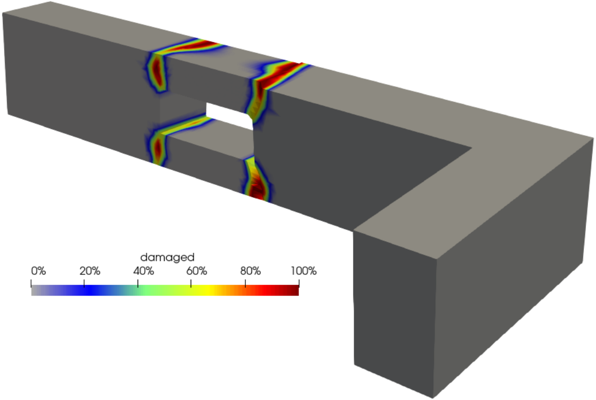
- compute
  - Laplace operator $\nabla^2 f(d)$
    $\rightarrow$ requires mesh data
  - new design d_{n+1} by solving PDE via FDM
    $\rightarrow$ within two nested loops independent of FEM



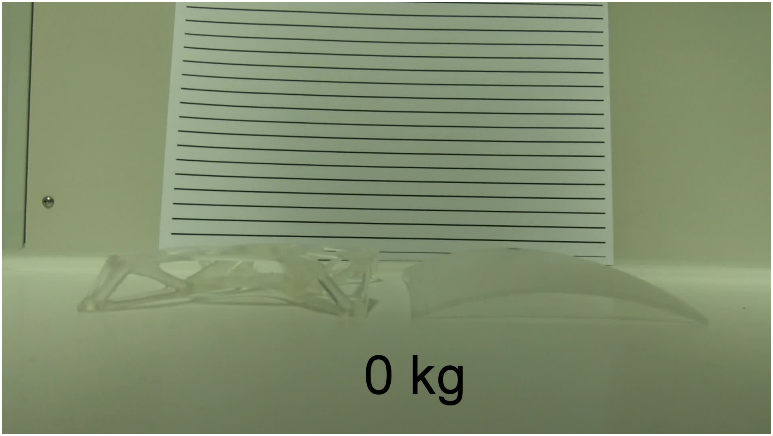
Damage simulation results



Damage simulation results



The potency of topology optimization



Thermodynamic topology optimization (TTO)

- extended Hamilton functional e.g. for topology optimization

$$\mathcal{H}[\boldsymbol{u},\boldsymbol{\alpha},\chi] := \underbrace{\mathcal{G}[\boldsymbol{u},\boldsymbol{\alpha},\chi]}_{\text{Gibbs energy}} + \underbrace{\mathcal{D}[\boldsymbol{\alpha}]}_{\text{dissipation}} - \underbrace{\mathcal{R}[\chi]}_{\text{rearrangement}} + \underbrace{\mathcal{C}[\boldsymbol{\alpha},\chi]}_{\text{constraints}}$$

↳ **design variable(s)** χ besides internal variables $\boldsymbol{\alpha}$
e.g. $\chi \rightarrow$ material distribution = topology

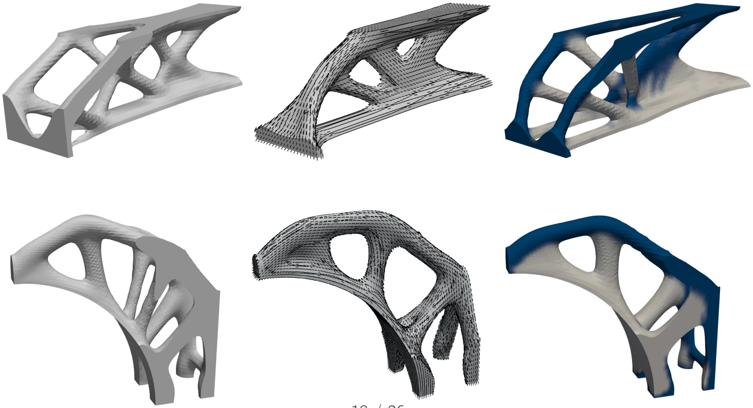
$$\left.\begin{array}{l} \mathcal{G} = \ldots \\ \mathcal{D} = \ldots \\ \mathcal{R} = \ldots \\ \mathcal{C} = \ldots \end{array}\right\} \Rightarrow$$

Already well documented example in Ferrite.jl! :-)

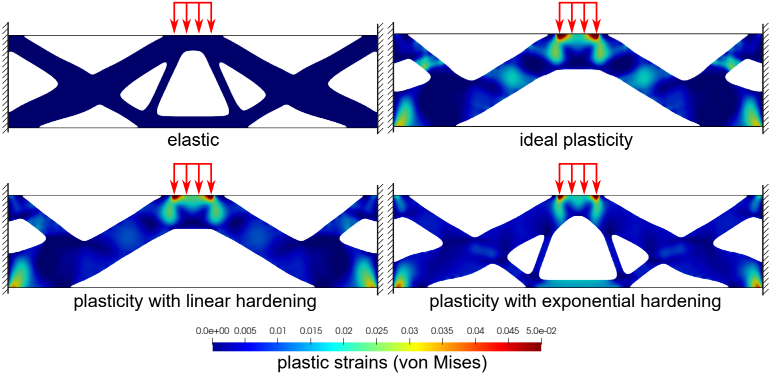
↳ Solution via NEM analogous to damage model

Different elastaic materials

linear-elastic | anisotropic | mixed material

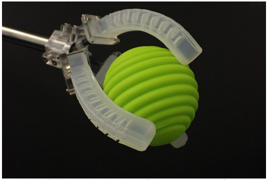


Plasticity with different kinds of hardening



Implementation of large deformations in TTO

- if a **Hyperelastic Material Model** is used:
  - **large** displacements $\boldsymbol{u}$ and strains ε
  - $\delta_{\boldsymbol{u}}\mathcal{H} = 0$ is **non-linear**
  - **snap-back** and **snap-through** possible
  - geometrical part of B-operator necessary

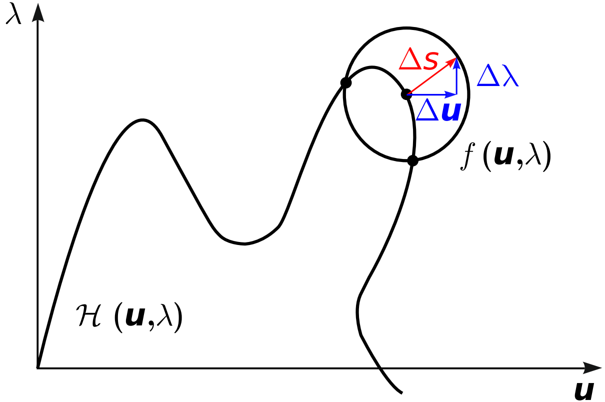


https://3druck.com/forschung/harvard-forscher-drucken-softroboter-mit-integrierten-sensoren-5968339/

- discretized system for **arc-length method**
  - $\begin{pmatrix} \boldsymbol{K} & -\boldsymbol{P} \\ f^T & f_{,\lambda} \end{pmatrix}_i \begin{Bmatrix} \Delta\boldsymbol{u} \\ \Delta\lambda \end{Bmatrix}_{i+1} = \begin{Bmatrix} \mathcal{H} \\ f \end{Bmatrix}_i$
- with an additional constrain $f(\boldsymbol{u}, \lambda)$
  - $f = \sqrt{(\Delta\boldsymbol{u})^T(\Delta\boldsymbol{u}) + (\Delta\lambda)^2} - \Delta s$

Visualization of the arc-length method

- solve problems with snap-back and snap-through



Dustin R. Jantos, Ilayda Kök, Hendrik Geisler, Miriam Kick, Max von Zabiensky

Algorithm: arc-lenght method according to [1]

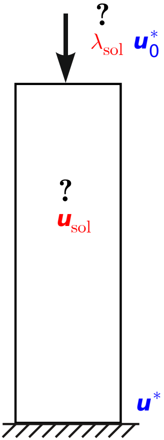
set $\boldsymbol{u}_0 = \boldsymbol{u}_k$ and Δs ▷ First Settings by User
compute $\boldsymbol{u}_{P_0} = \boldsymbol{P}/\boldsymbol{K}_0$ ▷ Predictor Step
compute $\lambda_0 = \lambda_k \pm \frac{\Delta s}{\sqrt{(\Delta \boldsymbol{u}_{P_0})^T \Delta \boldsymbol{u}_{P_0}}}$
while $\|\mathcal{H}(\boldsymbol{u}, \lambda)\| \leq \mathrm{TOL}$ **do**
  for $i = 0, 1, 2, \ldots$ **do**
    compute $\Delta \boldsymbol{u}_{P_{i+1}} = \boldsymbol{P}/\boldsymbol{K}_i$
    compute $\Delta \boldsymbol{u}_{\mathcal{H}_{i+1}} = -\mathcal{H}(\boldsymbol{u}_i, \lambda_i)/\boldsymbol{K}_i$
    compute $\Delta \lambda_{i+1} = \frac{f_i + \boldsymbol{f}_i^T \Delta \boldsymbol{u}_{\mathcal{H}_{i+1}}}{f_{,\lambda_i} + \boldsymbol{f}_i^T \Delta \boldsymbol{u}_{P_{i+1}}}$ ▷ Increments for next Step
    compute $\Delta \boldsymbol{u}_{i+1} = \Delta \lambda_{i+1} \Delta \boldsymbol{u}_{P_{i+1}} + \Delta \boldsymbol{u}_{\mathcal{H}_{i+1}}$
    update $\lambda_{i+1} = \lambda_i + \Delta \lambda_{i+1}$
    update $\boldsymbol{u}_{i+1} = \boldsymbol{u}_i + \Delta \boldsymbol{u}_{i+1}$
    update $\mathcal{H}(\boldsymbol{u}_{i+1}, \lambda_{i+1})$ and $i = i + 1$
  end for
end while

A new extended solution method according to [2]

- boundary conditions based on **displacements only**

- $$\boldsymbol{u} = \begin{cases} \boldsymbol{u}_{\text{sol}} \\ \boldsymbol{u}^* \\ \lambda_{\text{sol}}\ \boldsymbol{u}_0^* \end{cases}$$

- how to implement $\boldsymbol{u}$ in DoF-Handler?



Conclusions

- Developed at our institute in Ferrite.jl
  - **staggered models**
    - not "simply" coupled FEM problems
    - **external algorithms for material update**
    - "add-ons" to FEM with **modifications to core**

- Wishlist and eager to discuss
  - large deformations
    - geometrical part of shape derivatives
    - **arc length method** inlc. displacement based approach
  - 4D Space-Time → for the next FerriteCon :-)

References

[1] Peter Wriggers. *Nichtlineare finite-element-methoden*. Springer-Verlag, 2013.

[2] Giuliano Pretti, William M. Coombs, and Charles E. Augarde. A displacement-controlled arc-length solution scheme. *Computers & Structures*, 258:106674, 2022.

[3] Miriam Kick and Philipp Junker. Thermodynamic topology optimization including plasticity. *arXiv preprint arXiv:2103.03567*, 2021.

[4] Hendrik Geisler, Jan Nagel, and Philipp Junker. Simulation of the dynamic behavior of viscoelastic structures with random material parameters using time-separated stochastic mechanics. *submitted*, 2022.

[5] Dustin R. Jantos, Klaus Hackl, and Philipp Junker. An accurate and fast regularization approach to thermodynamic topology optimization. *International Journal for Numerical Methods in Engineering*, 117(9):991–1017, 2019.

[6] Andreas Vogel and Philipp Junker. Adaptive and highly accurate numerical treatment for a gradient-enhanced brittle damage model. *International Journal for Numerical Methods in Engineering*, 121(14):3108–3131, July 2020.

Acknowledgements

Thank you for your kind attention!

We highly acknowledge the funding of this research by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through the project grant JU 3096/2-1 and FOR5250/449916462 as well as the International Research Training Group IRTG 2657 "Computational Mechanics Techniques in High Dimensions" (Reference: GRK 2657/1).

www.ikm.uni-hannover.de

@ikm.luh