The MFrontGenericInterfaceSupport project

Thomas Helfer, Jérémy Bleyer, Tero Frondelius, Ivan Yashchuk, Thomas Nagel, Dmitri Naumov

The Journal of Open Source Software

The MFrontGenericInterfaceSupport project

Thomas Helfer[1], **Jeremy Bleyer**[2], **Tero Frondelius**[3, 4], **Ivan Yashchuk**[5, 6], **Thomas Nagel**[7, 8], **and Dmitri Naumov**[7, 8]

1 CEA, DES, IRESNE, DEC, Cadarache F-13108 Saint-Paul-Lez-Durance, France **2** Laboratoire Navier UMR 8205 (École des Ponts ParisTech-IFSTTAR-CNRS) **3** R&D and Engineering, Wärtsilä, P.O. Box 244, 65101 Vaasa, Finland **4** University of Oulu, Erkki Koiso-Kanttilan katu 1, 90014 Oulu, Finland **5** VTT Technical Research Centre of Finland, Kivimiehentie 3, 02150 Espoo, Finland **6** Department of Computer Science, Aalto University, Konemiehentie 2, 02150 Espoo, Finland **7** Geotechnical Institute, Technische Universität Bergakademie Freiberg, Gustav-Zeuner-Str. 1, 09599 Freiberg, Germany **8** Department of Environmental Informatics, Helmholtz Centre for Environmental Research – UFZ, Permoserstr. 15, 04318 Leipzig, Germany

DOI: 10.21105/joss.02003

Software

- Review
- Repository
- Archive

Editor: Arfon Smith

Reviewers:

- @srmnitc
- @IgorBaratta

Submitted: 12 December 2019
Published: 23 April 2020

Introduction

The ability to easily integrate user-defined constitutive equations plays a major role in the versatility of (mechanical) solvers[1].

Constitutive equations describe how the internal state variables of a material evolve with changing external conditions or mechanical loading. Those state variables can describe many microstructural aspects of the material (grain size, dislocation density, hardening state, etc.) or be phenomenological in nature (equivalent plastic strain). The knowledge of those internal state variables allows the computation of local thermodynamic forces which affect the material equilibrium at the structural scale. Due to the large number of phenomena that can be described in this manner, such as plasticity, viscoplasticity, or damage, computational mechanics is one of the most demanding domains for advanced constitutive equations.

At each time step, the constitutive equations must be integrated to obtain the state of the material at the end of the time step. As most phenomena are nonlinear, an iterative scheme is required at the equilibrium scale to find the local loading of the material: the integration of the constitutive equations is thus therefore called several times with different estimates of the loading of the material. Algorithmic efficiency at the constitutive level is therefore a key aspect for the overall efficiency of a code.

The `MFront` open-source code generator has been designed to simplify the implementation of the integration of the constitutive equations over a time step, to minimize errors during implementation, to facilitate the portability of constitutive equations between solvers, and to help achieve a reproducible and efficient code (CEA & EDF, 2019; Helfer et al., 2015). For that purpose, `MFront` uses a source file with a syntax very close to a physical/engineering description of the constitutive model, and generates `C++` code specific to many well-established (mostly thermo-mechanical) solvers through dedicated interfaces and compiles them into shared libraries. For example, `MFront` provides interfaces for `Cast3M`, `code_aster`, `Europlexus`, `Abaqus/Standard`, `Abaqus/Explicit`, `CalculiX`, etc.

To further facilitate this cross-software integration, `MFront` recently introduced a so-called `generic` interface. This paper describes the `MFrontGenericInterfaceSupport` project, which is denoted `MGIS` in the following. `MGIS` aims at proving tools (functions, classes,

[1] The term solver emphasizes that the numerical method used to discretize the equilibrium equations is not significant in the present context.

bindings to various programming languages) to handle behaviours[2] generated using `MFront`'s `generic` interface (Helfer, 2019a). Those tools alleviate the work required by solver developers. Permissive licences have been chosen to allow integration in open-source and proprietary codes.

This paper is divided into three parts:

1. A brief overview of `MGIS`.
2. A description of the various bindings available.
3. Some examples of usage in various open-source solvers: `FEniCS`, `OpenGeoSys` and `JuliaFEM`.

Overview

The aims of the `MFrontGenericInterfaceSupport` project are twofold:

1. At the pre-processing state `MGIS` shall provide the possibility of retrieving metadata about a particular behaviour and performing proper memory allocation. At the post-processing stage, easy access to internal state variables is desired.
2. During computations, `MGIS` shall simplify the integration of the behaviour at integration points[3] and the update of the internal state variables from one time step to the next.

Preprocessing and post-processing stages

When dealing with user-defined behaviours, most solvers, including `Abaqus/Standard` for example, delegates part of the work to the user. The user must:

1. Describe the behaviour in the input.
2. Take care of the consistency of the behaviour with the hypothesis made during the computation (e.g. a finite strain behaviour must be used in a finite strain analysis based on the appropriate deformation and stress measures as well as reference configurations).

In the authors' experience, this is error-prone in particular for inexperienced users and may lead to spurious or even worse inexact results. For example, when material properties such as the Young Modulus or the Poission ratio, are required by the behaviour, those are generally defined in the solver input file by an array of values and potential checks are limited to the size of the array. An user may thus invert two material properties. If those two material properties have the same order of magnitudes, computations might lead to a physically consistent result, despite this result being false.

`MGIS` introduces a very different approach: the user only declares the shared library, the behaviour and the modelling hypothesis (tridimensional, plane strain, etc.). With this information, the library retrieves various metadata which fully describe how to interact with the behaviour. The solver using `MGIS` can then check if the behaviour is consistent with the computations to be performed and checks that the data provided by the user are correct. The metadata can also be used to allocate the memory required to store the state of the material at each integration point. `MGIS`' design allows the following types of storage:

[2] In the following, we use the term "behaviour" to denote the result of the implementation and compilation of the constitutive equations.

[3] The term "integration points" is used here as a generic placeholder. When using FFT for solving the equilibrium equations, the integration points are voxels. When using FEM, the integration points are typically the Gauss points of the elements.

- An MGIS data structure per integration point. This is the most frequent choice. The memory is automatically allocated by MGIS.
- An MGIS data structure that stores the states of an arbitrary number of integration points. MGIS can allocate the memory associated with the state of all specified integration points or borrow memory allocated by the solver.

For post-processing, MGIS provides a set of functions to retrieve information about the state of the material. For example, one can retrieve the value of a state variable from the described data structures.

Computation stage

MGIS provides a function to integrate the constitutive equations at one integration point or on a set of integration points[4]. The integration of the constitutive equations at different integration points are usually independent of each other; in other words, in most models the constitutive behaviour is local. Thus, when handling a set of integration points, MGIS can parallelize the integration calls using a granularity chosen by the solver.

Main language and available bindings

MGIS is written in C++-11. The C++ API is described in another report, see Helfer (2019b).

The following bindings are currently available:

- python.
- Julia.
- Fortran 2003.
- C.

Examples of usage

FEniCS

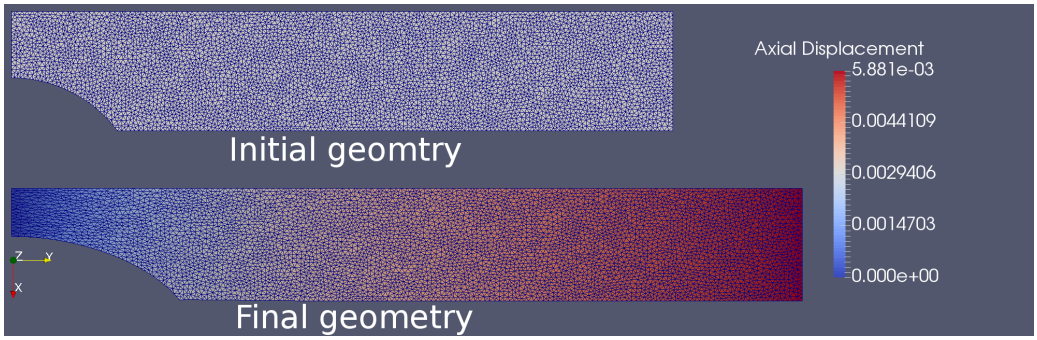

Figure 1: Large strain elasto-plastic modelling of a notched bar.

FEniCS is a popular open-source computing platform for solving partial differential equations (Alnæs et al., 2015; Logg, Mardal, Wells, & others, 2012).

[4] This strongly depends on the data structure chosen to store the internal state variables.

Non linear mechanics computations combining FEniCS at the equilibrium scale and MFront to describe the constitutive equations can be performed through the python bindings of MGIS as demonstrated by Bleyer et al. (see Bleyer & Helfer (2019a); Bleyer & Helfer (2019b)).

Extensions to finite strain elastoplasticity have been recently added as shown in Figure 1 which models a tensile test on a notched bar[5].

OpenGeoSys

OpenGeoSys (OGS) is a scientific open-source initiative for the numerical simulation of thermo-hydro-mechanical/chemical (THMC) processes in porous and fractured media, inspired by FEFLOW and ROCKFLOW concepts and continuously developed since the mid-eighties (see Kolditz (1990); Wollrath (1990); Kroehn (1991); Helmig (1993); Kolditz et al. (2012); Bilke et al. (2019)).

The OGS framework is targeting applications in the environmental geosciences, e.g., in the fields of contaminant hydrology, water resources and waste management, geotechnical applications, geothermal energy systems and energy storage.

The most recent version, OpenGeoSys-6 (OGS-6) (Naumov et al. (2018); Bilke et al. (2019)), is a fundamental re-implementation of the multi-physics code OpenGeoSys-4/5 (Kolditz & Bauer (2004); Wang & Kolditz (2006)) using advanced methods in software engineering and architecture with a focus on code quality, modularity, performance and comprehensive documentation.

Among its recent extensions are the implementation of numerical methods for the propagation of discontinuities, such as enriched finite element function spaces, non-local formulations and phase-field models for fracture (Watanabe, Wang, Taron, Görke, & Kolditz (2012); Parisio et al. (2019); Yoshioka et al. (2019)).

To simplify the implementation of new constitutive models for solids developed with MFront, OGS-6 relies on the C bindings of MGIS.

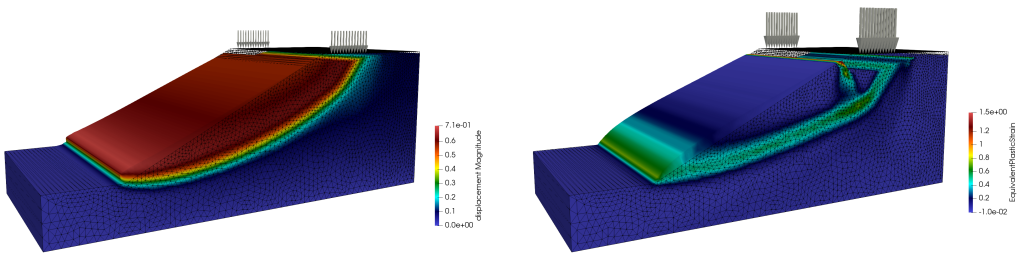

Figure 2: Slope stability analysis with strength reduction performed in OpenGeoSys. The image on the left shows the norm of the displacement vector for a low top load. The image on the right shows the equivalent plastic strain for a setting with an increased top load.

Figure 2 shows the results of a $\varphi - c$ reduction approach to slope stability analysis. The soil is modelled by a non-associated plastic behaviour based on the Mohr-Coulomb yield criterion (Abbo & Sloan, 1995; Zienkiewicz & Pande, 1977). The implementation presented in Nagel et al. (2017) was extended by a tension cut-off. The image on the left shows the norm of the displacement vector for a low top load. The failure kinematics are clearly visible. The image on the right shows the equivalent plastic strain for a setting with an increased top load. It can be seen that the failure mechanism becomes more complex with an additional slip surface forming beneath the load.

[5] This case is adapted from a non-regression test of the Code_Aster finite element solver, see EDF (2011) for details

JuliaFEM

JuliaFEM (Aho et al., 2019a, 2019b; Frondelius & Aho, 2017; Rapo, Aho, & Frondelius, 2017; Rapo, Aho, Koivurova, & Frondelius, 2018; Rapo et al., 2019) is an open-source finite element solver written in the Julia programming language (Bezanson, Edelman, Karpinski, & Shah, 2017). JuliaFEM enables flexible simulation models, takes advantage of the scripting language interface, which is easy to learn and embrace. Besides, it is a real programming environment where other analyses and workflows combine with simulation.

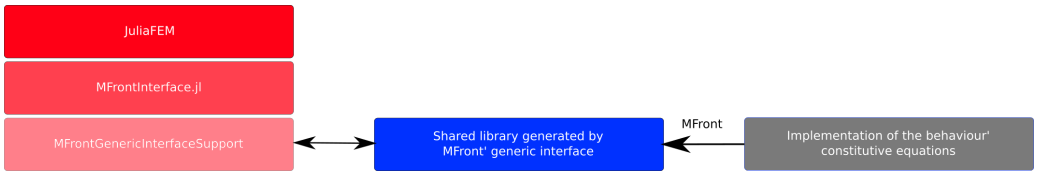

Figure 3: Block diagram showing the software layers involved in using MFront behaviours in JuliaFEM.

The MFrontInterface.jl (Frondelius, Helfer, Yashchuk, Vaara, & Laukkanen, 2019) is a Julia package where MFront material models are brought to Julia via wrapping MGIS, see Figure 3. Installation is, as easy as any julia packages, i.e., pkg> add MFrontInterface. For example TFEL and MGIS cross-compiled binary dependencies are automatically downloaded and extracted. Lastly, Figure 4. shows a simple 3D geometry example using JuliaFEM and MFrontInterface together.

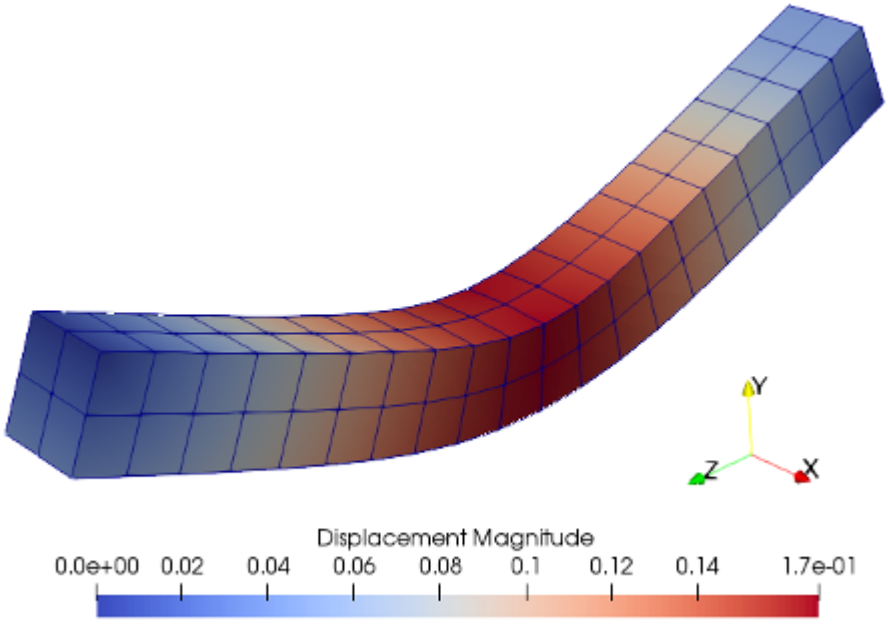

Figure 4: Simple isotropic plasticity modelling of a 3D beam in JuliaFEM with MFrontInterface.

Conclusions

This paper introduces the MFrontGenericInterfaceSupport library which considerably eases the integration of MFront-generated behaviours in any solver. In particular, the library

provides a way of retrieving the metadata associated with a behaviour, data structures to store the physical information and functions to perform the behaviour integration over a time step. Examples of usage in various open-source solvers (`FEniCS`, `OpenGeoSys`, `JuliaFEM`) have been provided.

The models implemented for one code can easily be used in another without the need for re-implementation. This offers great benefits for code quality assurance. Since the constitutive integration is handled by MFront, this step of the computation is equally efficient across the different solver platforms.

Acknowledgements

This research was conducted in the framework of the `PLEIADES` project, which is supported financially by the CEA (Commissariat à l'Energie Atomique et aux Energies Alternatives), EDF (Electricité de France) and Framatome.

We would like to express our thanks to Christoph Lehmann, Francesco Parisio, Olaf Kolditz and the entire community of developers and users of OpenGeoSys(OGS). We thank the Helmholtz Centre for Environmental Research – UFZ for long-term funding and continuous support of the OpenGeoSys initiative. OGS has been supported by various projects funded by Federal Ministries (BMBF, BMWi) as well as the German Research Foundation (DFG). We further thank the Federal Institute for Geosciences and Natural Resources (BGR) for funding.

Also, we would like to acknowledge the financial support of Business Finland for both ISA Wärtsilä Dnro 7734/31/2018, and ISA VTT Dnro 7980/31/2018 projects.

This project uses code extracted from the following projects:

- https://github.com/bitwizeshift/string_view-standalone by Matthew Rodusek
- https://github.com/mpark/variant: by Michael Park
- https://github.com/progschj/ThreadPool by Jakob Progsch and Václav Zeman
- https://github.com/martinmoene/span-lite by Martin Moene
- https://bitbucket.org/fenics-apps/fenics-solid-mechanics/ by Kristian B. Ølgaard and Garth N. Wells.

References

Abbo, A. J., & Sloan, S. W. (1995). A smooth hyperbolic approximation to the mohr-coulomb yield criterion. *Computers & Structures*, *54*(3), 427–441. doi:10.1016/0045-7949(94)00339-5

Aho, J., Jämsä, V., Kuivaniemi, T., Liljenfeldt, N., & Frondelius, T. (2019a). JuliaFEM beam element implementation. *Rakenteiden Mekaniikka*, *52*(3), 160–176. doi:10.23998/rm.76193

Aho, J., Vuotikka, A.-J., & Frondelius, T. (2019b). Introduction to JuliaFEM an open-source fem solver. *Rakenteiden Mekaniikka*, *52*(3), 148–159. doi:10.23998/rm.75103

Alnæs, M., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., Richardson, C., et al. (2015). The FEniCS project version 1.5. *Archive of Numerical Software*, *3*(100). doi:10.11588/ans.2015.100.20553

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, *59*(1), 65–98. doi:10.1137/141000671

Bilke, L., Flemisch, B., Kalbacher, T., Kolditz, O., Helmig, R., & Nagel, T. (2019). Development of Open-Source Porous Media Simulators: Principles and Experiences. *Transport in Porous Media*, *130*(1), 337–361. doi:10.1007/s11242-019-01310-1

Bleyer, J., & Helfer, T. (2019a). Elasto-plastic analysis implemented using the MFront code generator. Numerical tours of continuum mechanics using FEniCS. Retrieved from https://comet-fenics.readthedocs.io/en/latest/demo/plasticity_mfront/plasticity_mfront.py.html

Bleyer, J., & Helfer, T. (2019b). FEniCS and MFront for complex non linear solid mechanics simulation. doi:10.13140/RG.2.2.35501.54247

CEA, & EDF. (2019). MFront web site. Retrieved from http://www.tfel.sourceforge.net/

EDF. (2011). *SSNA303 : Éprouvette entaillée élastoplastique en grandes déformations* (Référence du Code Aster No. V6.01.303 révision : 6972). EDF-R&D/AMA. Retrieved from http://www.code-aster.org

Frondelius, T., & Aho, J. (2017). JuliaFEM - open source solver for both industrial and academia usage. *Rakenteiden Mekaniikka*, *50*(3), 229–233. doi:10.23998/rm.64224

Frondelius, T., Helfer, T., Yashchuk, I., Vaara, J., & Laukkanen, A. (2019). MFrontInterface.jl: MFront material models in juliaFEM. In H. Koivurova & A. H. Niemi (Eds.), *Proceedings of the 32nd nordic seminar on computational mechanics*.

Helfer, T. (2019a). *MGIS*. Retrieved from https://github.com/thelfer/MFrontGenericInterfaceSupport

Helfer, T. (2019b, May). A brief introduction to the MGIS c++ library for mechanical behaviours. doi:10.13140/RG.2.2.22079.76968

Helfer, T., Michel, B., Proix, J.-M., Salvo, M., Sercombe, J., & Casella, M. (2015). Introducing the open-source mfront code generator: Application to mechanical behaviours and material knowledge management within the PLEIADES fuel element modelling platform. *Computers & Mathematics with Applications*, *70*(5), 994–1023. doi:10.1016/j.camwa.2015.06.027

Helmig, R. (1993). *Theorie und Numerik der Mehrphasenströmungen in geklüftet-porösen Medien* (PhD thesis). Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover.

Kolditz, O. (1990). *Zur Modellierung und Simulation geothermischer Transportprozesse in untertägigen Zirkulationssystemen* (PhD thesis). Akademie der Wissenschaften der DDR, Berlin.

Kolditz, O., & Bauer, S. (2004). A process-oriented approach to computing multi-field problems in porous media. *Journal of Hydroinformatics*, *6*(3), 225–244. doi:10.2166/hydro.2004.0017

Kolditz, O., Bauer, S., Bilke, L., Böttcher, N., Delfs, J. O., Fischer, T., Görke, U. J., et al. (2012). OpenGeoSys: An open-source initiative for numerical simulation of thermo-hydro-mechanical/chemical (THM/c) processes in porous media. *Environmental Earth Sciences*, *67*(2), 589–599. doi:10.1007/s12665-012-1546-x

Kroehn, K. P. (1991). *Simulation von Transportvorgängen im klüftigen Gestein mit der Methode der Finiten Elemente* (PhD thesis). Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover.

Logg, A., Mardal, K.-A., Wells, G. N., & others. (2012). *Automated solution of differential equations by the finite element method*. Springer. doi:10.1007/978-3-642-23099-8

Nagel, T., Minkley, W., Böttcher, N., Naumov, D., Görke, U.-J., & Kolditz, O. (2017). Implicit numerical integration and consistent linearization of inelastic constitutive models of rock salt. *Computers & Structures*, *182*, 87–103. doi:10.1016/j.compstruc.2016.11.010

Naumov, D., Bilke, L., Fischer, T., Huang, Y., Lehmann, C., Miao, X.-Y., Nagel, T., et al. (2018). Appendix a: OpenGeoSys-6. In O. Kolditz, T. Nagel, H. Shao, W. Wang, & S. Bauer (Eds.), *Thermo-hydro-mechanical-chemical processes in fractured porous media: Modelling and benchmarking* (pp. 271–277). Springer International Publishing.

Parisio, F., Tarokh, A., Makhnenko, R., Naumov, D., Miao, X.-Y., Kolditz, O., & Nagel, T. (2019). Experimental characterization and numerical modelling of fracture processes in granite. *International Journal of Solids and Structures*, *163*(xxxx), 102–116. doi:10.1016/j.ijsolstr.2018.12.019

Rapo, M., Aho, J., & Frondelius, T. (2017). Natural frequency calculations with JuliaFEM. *Rakenteiden Mekaniikka*, *50*(3), 300–303. doi:10.23998/rm.65040

Rapo, M., Aho, J., Koivurova, H., & Frondelius, T. (2018). Implementing model reduction to the JuliaFEM platform. *Rakenteiden Mekaniikka*, *51*(1), 36–54. doi:10.23998/rm.69026

Rapo, M., Vaara, J., Aho, J., Kuivaniemi, T., Liljenfeldt, N., Vuohijoki, A., & Frondelius, T. (2019). Pipe route optimization to avoid undesired vibration by using juliaFEM. *Rakenteiden Mekaniikka*, *52*(3), 177–191. doi:10.23998/rm.76259

Wang, W., & Kolditz, O. (2006). Object-oriented finite element analysis of thermo-hydro-mechanical (thm) problems in porous media. *International Journal for Numerical Methods in Engineering*, *69*(1), 162–201. doi:10.1002/nme.1770

Watanabe, N., Wang, W., Taron, J., Görke, U., & Kolditz, O. (2012). Lower-dimensional interface elements with local enrichment: application to coupled hydro-mechanical problems in discretely fractured porous media. *International Journal for Numerical Methods in Engineering*, *90*(8), 1010–1034. doi:10.1002/nme.3353

Wollrath, J. (1990). *Ein Strömungs- und Transportmodell für klüftiges Gestein und Untersuchungen zu homogenen Ersatzsystemen* (PhD thesis). Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover.

Yoshioka, K., Parisio, F., Naumov, D., Lu, R., Kolditz, O., & Nagel, T. (2019). Comparative verification of discrete and smeared numerical approaches for the simulation of hydraulic fracturing. *GEM - International Journal on Geomathematics*, *10*(1), 13. doi:10.1007/s13137-019-0126-6

Zienkiewicz, O. C., & Pande, G. N. (1977). Some Useful Forms of Isotropic Yield Surfaces for Soil and Rock Mechanics. In G. Gudehus (Ed.), *Finite elements in geomechanics* (pp. 179–190). John Wiley & Sons, Inc, New York, London, Sydney.