

Mechkit: A continuum mechanics toolkit in Python

Julian Karl Bauer ¹✉, Philipp Lothar Kinon ¹, Jonas Hund ², Lisa Latussek ¹, Nils Meyer ³, and Thomas Böhlke ⁴

¹ Institute of Mechanics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany ² Department of Structural Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway ³ Institute of Vehicle System Technology, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany ⁴ Chair for Continuum Mechanics, Institute of Engineering Mechanics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany ✉ Corresponding author

DOI: [10.21105/joss.04389](https://doi.org/10.21105/joss.04389)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Kevin M. Moerman](#)  

Reviewers:

- [@nicoguardo](#)
- [@likask](#)
- [@lizarett](#)

Submitted: 23 March 2022

Published: 11 October 2022

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

The Python package `mechkit` is a toolkit for researchers in the field of continuum mechanics and material modeling. `mechkit` contains methods and operators for elementary tasks concerning tensor algebra and tensor notation, e.g., linear mapping, base transformations and active as well as passive transformations of tensors.

In the context of engineering applications in three spatial dimensions, strains and stresses in solids are usually described by second-order tensors. In linear elasticity, mappings between stresses and strains are important. To this end, the methods in `mechkit` are focussed on second- and fourth-order tensors. Main motivations can thus be found in the research concerning linear elasticity ([Bertram & Glüge, 2015](#)), ([Mandel, 1965](#)), ([Fedorov, 1968](#)), ([Mehrabadi & Cowin, 1990](#)), ([Thomson, 1856](#)), ([Cowin & Mehrabadi, 1992](#)), ([Rychlewski, 2000](#)), ([Spencer, 1970](#)), ([Böhlke & Brüggemann, 2001](#)), ([Brannon, 2018](#)) and the description of microstructures of fiber-reinforced composite materials ([Bauer & Böhlke, 2021](#)), ([Ken-Ichi, 1984](#)), ([Advani & Tucker III, 1987](#)).

The implementations in `mechkit` aim at usability, seek to provide understandable source code, and do not put major emphasis on performance. Furthermore, the implementations follow, as directly as possible, the notation and formulation in the respective primary scientific sources. A redundant implementation of identical operations based on different sources is aimed at, for validation reasons.

Statement of need

The mathematical operators for material modeling and linear elasticity can be expressed in a compact manner. However, their representation is not unique and the relation between different representations is nontrivial. To the best knowledge of the authors, there is no common library for the mathematical operators to this day. This presents an obstacle with regard to the exchange and the reliability of research code and leads to negative consequences for the interpretation and comparison of results.

Several multi-purpose tensor packages in Python, e.g., ([Kossaifi et al., 2016](#)), ([Hauschild & Pollmann, 2018](#)) exist, but mainly address the manipulation of multi-dimensional matrices which are not related to physical bases. In contrast, `mechkit` focuses on linear elasticity and provides a limited scope of notations for second- and fourth-order tensors.

The main goal of `mechkit` is to provide reusable research code that increases the reliability of results. It is intended to accelerate and simplify further research. `mechkit` is inspired by ([Meyer, 2021](#)), ([Schlömer, 2022a](#)), ([Schlömer, n.d.](#)) and ([Schlömer, 2022b](#)).

Motivation by example: Isotropic material and notations

In the overlapping area of theoretical continuum mechanics, experimental material modeling, and numerical solution methods for boundary value problems, a multitude of different notations exist. As an example, one may consider the description of the mechanical properties of a homogeneous and isotropic, i.e. direction-independent, material within the framework of linear elasticity. Such a material can be described completely by two independent scalar parameters. However, in the disciplines mentioned above, at least six different material parameters are commonly used, motivated by different applications, measurement techniques and tensor decompositions. This results in fifteen possible combinations of scalar descriptions of an isotropic material, which can be combined to the corresponding fourth-order elasticity tensor. For this tensor, again, different representations exist, referring to different basis systems and notations. The notations either follow the Voigt or Kelvin-Mandel notation or take account of interfaces of finite element codes.

The translation between different notations is often tedious and prone to errors. `mechkit` allows an easy exchange between different notations with user-friendly interfaces, thereby preventing errors. The necessary number of translation functions between different notations increases drastically with an increasing number of notations. Consequentially, even for a small number of different notations, the implementation of all corresponding translation methods is not feasible. Therefore, `mechkit` does not necessarily directly translate one notation into another. Instead, in the case of the translation of second- and fourth-order tensors, `mechkit` determines the shortest path between source and target in the set of implemented notations as illustrated in the graph of currently supported notations, see [Figure 1](#). This procedure greatly facilitates the addition of further notations to `mechkit`. Essentially, only a translation function from and to a new notation has to be added to the existing code, to make translations from and to this new notation universally available in `mechkit`.

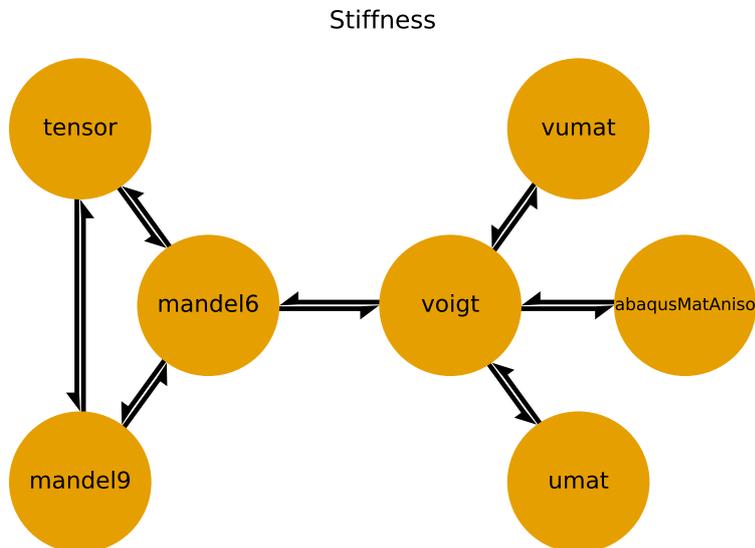


Figure 1: Currently supported notations and translations of fourth-order stiffness tensors.

Acknowledgements

The research documented in this manuscript has been funded by the German Research Foundation (DFG) within the International Research Training Group “Integrated engineering of continuous-discontinuous long fiber-reinforced polymer structures” (GRK 2078/2). The support by the German Research Foundation (DFG) is gratefully acknowledged.

References

- Advani, S. G., & Tucker III, C. L. (1987). The use of tensors to describe and predict fiber orientation in short fiber composites. *Journal of Rheology*, 31(8), 751–784. <https://doi.org/10.1122/1.549945>
- Bauer, J. K., & Böhlke, T. (2021). Variety of fiber orientation tensors. *Math. Mech. Solids*, 0(0), 108128652110576. <https://doi.org/10.1177/10812865211057602>
- Bertram, A., & Glüge, R. (2015). *Solid mechanics*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-19566-7>
- Böhlke, T., & Brüggemann, C. (2001). Graphical representation of the generalized Hooke's law. *Technische Mechanik*, 21(2), 145–158.
- Brannon, R. (2018). *Rotation reflection and frame changes orthogonal tensors in computational engineering mechanics*. IOP Publishing. <https://doi.org/10.1088/978-0-7503-1454-1>
- Cowin, S. C., & Mehrabadi, M. M. (1992). The structure of the linear anisotropic elastic symmetries. *J. Mech. Phys. Solids*, 40(7), 1459–1471. [https://doi.org/10.1016/0022-5096\(92\)90029-2](https://doi.org/10.1016/0022-5096(92)90029-2)
- Fedorov, F. I. (1968). *Theory of elastic waves in crystals*. Springer US. <https://doi.org/10.1007/978-1-4757-1275-9>
- Hauschild, J., & Pollmann, F. (2018). Efficient numerical simulations with Tensor Networks: Tensor Network Python (TeNPy). *SciPost Phys. Lect. Notes*, 5. <https://doi.org/10.21468/SciPostPhysLectNotes.5>
- Ken-Ichi, K. (1984). Distribution of directional data and fabric tensors. *Int. J. Eng. Sci.*, 22(2), 149–164. [https://doi.org/10.1016/0020-7225\(84\)90090-9](https://doi.org/10.1016/0020-7225(84)90090-9)
- Kossaifi, J., Panagakis, Y., Anandkumar, A., & Pantic, M. (2016). Tensorly: Tensor learning in python. *arXiv Preprint arXiv:1610.09555*.
- Mandel, J. (1965). Généralisation de la théorie de plasticité de WT koiter. *Int. J. Solids Struct.*, 1(3), 273–295.
- Mehrabadi, M. M., & Cowin, S. C. (1990). Eigentensors of linear anisotropic elastic materials. *Quart. J. Mech. Appl. Math.*, 43(1), 15–41. <https://doi.org/10.1093/qjmam/43.1.15>
- Meyer, N. (2021). Fiberoripy. In *GitHub repository* (Version v1.0.12) [Computer software]. GitHub. <https://doi.org/10.5281/zenodo.4679756>
- Rychlewski, J. (2000). A qualitative approach to Hooke's tensors. Part I. *Archives of Mechanics*, 52(4-5), 737–759.
- Schlömer, N. (n.d.). *pygalmesh: Python interface for CGAL's meshing tools*. <https://doi.org/10.5281/zenodo.5564818>
- Schlömer, N. (2022a). Meshio: Tools for mesh files. In *GitHub repository*. GitHub. <https://doi.org/10.5281/zenodo.1173115>
- Schlömer, N. (2022b). Quadpy: Your one-stop shop for numerical integration in python. In *GitHub repository*. GitHub. <https://github.com/nschloe/quadpy>
- Spencer, A. (1970). A note on the decomposition of tensors into traceless symmetric tensors. *International Journal of Engineering Science*, 8(6), 475–481. [https://doi.org/10.1016/0020-7225\(70\)90024-8](https://doi.org/10.1016/0020-7225(70)90024-8)
- Thomson, W. (1856). Elements of a mathematical theory of elasticity. *Philosophical Transactions of the Royal Society of London*, 146, 481–498.