The Journal of Open Source Software

Kanapy: A Python package for generating complex synthetic polycrystalline microstructures

Mahesh R.G. Prasad[1], **Napat Vajragupta**[1], **and Alexander Hartmaier**[1]

1 Interdisciplinary Centre for Advanced Materials Simulation, Ruhr-Universität Bochum, Universitätsstr. 150, 44801 Bochum, Germany.

DOI: 10.21105/joss.01732

Software

- Review
- Repository
- Archive

Submitted: 09 August 2019
Published: 08 November 2019

Summary

To study process-structure-property relationships it is essential to understand the contribution of microstructure to material behavior. Micromechanical modeling allows us to understand the influence of microstructural features on the macroscopic mechanical behavior through numerical simulations. At the center of this approach lies the modeling of synthetic microstructures that mimick the important aspects such as grain morphologies and crystallographic orientations.

Metallic material processing (including additive manufacturing) often result in complex microstructures with equiaxed/elongated grains and strong crystallographic textures. The current state-of-the-art of synthetic microstructure generation includes probabilistic methods like spatial tessellation (Quey, Dawson, & Barbe, 2011), which provides sufficiently accurate representations for simple grain morphologies and size distributions, but cannot capture complex morphologies, i.e., irregularly shaped grains. Another widely used approach is the Random Sequential Addition (RSA) (Groeber & Jackson, 2014; Vajragupta et al., 2014), which overcomes the shortcomings of tessellation based methods with its ability to model convex and non-convex grain morphologies, but its computational expense is high as space filling by random addition of particles is not efficient for higher volume fractions (Zhang & Torquato, 2013).

`Kanapy` is a Python package for generating complex synthetic polycrystalline microstructures based on the collision driven particle dynamics approach. `Kanapy` is designed to model irregular shaped grains and developed to provide an alternative to the existing particle packing approach, the RSA technique. This addresses the limitations of the existing microstructure generation techniques described earlier. In this regard, `Kanapy` employs a two-layer collision detection scheme, wherein the outer layer utilizes an octree spatial partitioning data structure to estimate which particles should be checked for collision. The inner layer consists of a bounding spheres hierarchy, which carries out the collision detection only if the bounding spheres between two particles overlap. The computational complexity involved in using the octree data structure is of the order $O(nlog(n))$. The actual collision detection between two static ellipsoidal particles is determined by employing the algebraic separation condition developed by Wang et al. (2001). Using the built-in voxelization routine, complex microstructures like those found in additively manufactured components can be easily created.

`Kanapy` is a modular package and it gives the user the flexibility to create individual workflows for generating specific microstructures. The modules can be executed independently of one another as Kanapy provides easy data storage and handling. It is based on existing implementations of convex hull from the Scipy package (Jones, Oliphant, Peterson, & others, n.d.) together with various Numpy array operations (Oliphant, 2006). A pure Python octree data structure is implemented for efficient collision handling. The performance critical part

of the actual collision detection code is implemented in C++ (using the Eigen library (Guennebaud, Jacob, & others, 2010)) with Python bindings generated using the header-only library pybind11 (Jakob, Rhinelander, & Moldovan, 2017). Examples for generating microstructures with equiaxed and elongated grains along with simulation benchmarks are detailed in the documentation.

References

Groeber, M. A., & Jackson, M. A. (2014). DREAM.3D: A digital representation environment for the analysis of microstructure in 3D. *Integrating Materials and Manufacturing Innovation*, *3*(1), 56–72. doi:10.1186/2193-9772-3-5

Guennebaud, G., Jacob, B., & others. (2010). Eigen v3. http://eigen.tuxfamily.org.

Jakob, W., Rhinelander, J., & Moldovan, D. (2017). Pybind11 – seamless operability between C++11 and Python. https://github.com/pybind/pybind11.

Jones, E., Oliphant, T., Peterson, P., & others. (n.d.). SciPy: Open source scientific tools for Python. Retrieved from http://www.scipy.org/

Oliphant, T. E. (2006). *A guide to numpy* (Vol. 1). Trelgol Publishing USA.

Quey, R., Dawson, P. R., & Barbe, F. (2011). Large-scale 3D random polycrystals for the finite element method: Generation, meshing and remeshing. *Computer Methods in Applied Mechanics and Engineering*, *200*(17), 1729–1745. doi:10.1016/j.cma.2011.01.002

Vajragupta, N., Wechsuwanmanee, P., Lian, J., Sharaf, M., Münstermann, S., Ma, A., Hartmaier, A., et al. (2014). The modeling scheme to evaluate the influence of microstructure features on microcrack formation of dp-steel: The artificial microstructure model and its application to predict the strain hardening behavior. *Computational Materials Science*, *94*, 198–213. doi:10.1016/j.commatsci.2014.04.011

Wang, W., Wang, J., & Kim, M.-S. (2001). An algebraic condition for the separation of two ellipsoids. *Computer Aided Geometric Design*, *18*(6), 531–539. doi:10.1016/S0167-8396(01)00049-8

Zhang, G., & Torquato, S. (2013). Precise algorithm to generate random sequential addition of hard hyperspheres at saturation. *Phys. Rev. E*, *88*(5), 053312. doi:10.1103/PhysRevE.88.053312