The Journal of Open Source Software

CRATE: A Python package to perform fast material simulations

Bernardo P. Ferreira [1,2], **F. M. Andrade Pires** [2], and **Miguel A. Bessa** [1¶]

1 School of Engineering, Brown University, USA **2** Faculty of Engineering, University of Porto, Portugal
¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

CRATE (Clustering-based Nonlinear Analysis of Materials) is a Python project (package `cratepy`) developed in the context of computational mechanics (B. P. Ferreira, 2022) to aid the design and development of new materials. Its main purpose is performing multi-scale nonlinear analyses of heterogeneous materials through a suitable coupling between first-order computational homogenization and clustering-based reduced-order modeling. This means that, given a representative volume element of the material micro-structure and the corresponding material phase properties, `cratepy` computes the material's effective mechanical response under a given loading by leveraging a so-called clustering-based reduced-order model (CROM).

Figure 1: Logo of CRATE (`cratepy`).

Statement of need

CRATE (Clustering-based Nonlinear Analysis of Materials) is a Python project (package `cratepy`) developed in the field of computational mechanics and material science. To the best of the authors' knowledge, it is a first-of-its-kind open-source software that allows any material development enthusiast to perform multi-scale analyses of materials by taking advantage of the recent family of clustering-based reduced-order models (CROMs). Figure 2 provides a simple illustration of a CRATE simulation. It is worth remarking that CRATE is supported by a rich documentation that provides a conceptual overview of the project, clear installation instructions, a step-by-step basic workflow description, and detailed guidelines for advanced customized developments. Moreover, `cratepy` relies solely on a few well-established third-party Python scientific computing packages, such as `numpy` (Harris et al., 2020) and `scipy` (Virtanen et al., 2020), `cratepy`'s modules are extensively documented, and the automatically generated API provides a complete and updated description of the underlying object-oriented implementation, including LaTeX rendered formulae to improve comprehension.

`cratepy` is essentially a numerical tool for any application that requires material multi-scale simulations. Given the intrinsic clustering-based reduced-order modeling approach (e.g., SCA (Liu et al., 2016), ASCA (Ferreira B. P. et al., 2022)), CRATE is mostly useful in applications where the computational cost of standard simulation methods is prohibitive, namely to solve lower-scales in coupled hierarchical multi-scale simulations (e.g., B. P. Ferreira (2022)) and to generate large material response databases for data-driven frameworks based on machine learning (e.g., Bessa et al. (2017)). CROMs achieve a striking balance between accuracy and computational cost by first performing a clustering-based domain decomposition of the material model and then solving the equilibrium problem formulated over the resulting reduced model. In the particular case of a research environment, `cratepy` is designed to easily accommodate further developments, either by improving the already implemented methods or by including new numerical models and techniques. It also provides all the fundamental means to perform comparisons with alternative methods, both in terms of accuracy and computational cost. In a teaching environment, `cratepy` is a readily available tool for demonstrative purposes and/or academic work proposals in solid mechanics and material-related courses.

We hope that CRATE contributes effectively to the development of new materials and encourages other researchers to share their own projects.

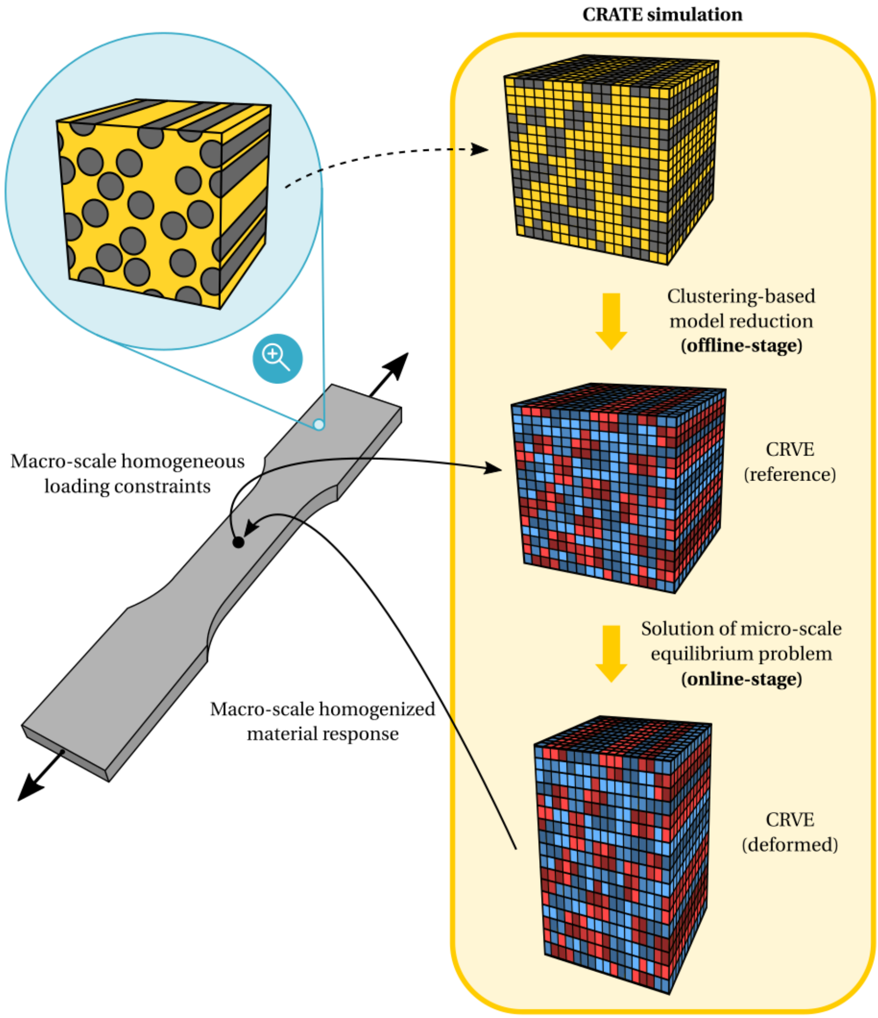

Figure 2: Schematic illustration of CRATE (`cratepy`) simulation.

Acknowledgements

Bernardo P. Ferreira acknowledges the support provided by Fundação para a Ciência e a Tecnologia (FCT, Portugal) through the scholarship with reference SFRH/BD/130593/2017. This research has also been supported by Instituto de Ciência e Inovação em Engenharia

Mecânica e Engenharia Industrial (INEGI, Portugal). Miguel A. Bessa acknowledges the support from the project ‘Artificial intelligence towards a sustainable future: ecodesign of recycled polymers and composites’ (with project number 17260 of the research programme Applied and Engineering Sciences) which is financed by the Dutch Research Council (NWO), The Netherlands.

References

Bessa, M. A., Bostanabad, R., Liu, Z., Hu, A., Apley, D. W., Brinson, C., Chen, W., & Liu, W. K. (2017). A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality. *Computer Methods in Applied Mechanics and Engineering*, *320*, 633–667. https://doi.org/10.1016/j.cma.2017.03.037

Ferreira, B. P. (2022). *Towards data-driven multi-scale optimization of thermoplastic blends: Microstructural generation, constitutive development and clustering-based reduced-order modeling* [PhD thesis, University of Porto]. https://hdl.handle.net/10216/146900

Ferreira, B. P., Andrade Pires, F. M., & Bessa, M. A. (2022). Adaptivity for clustering-based reduced-order modeling of localized history-dependent phenomena. *Computer Methods in Applied Mechanics and Engineering*, *393*, 114726. https://doi.org/https://doi.org/10.1016/j.cma.2022.114726

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Liu, Z., Bessa, M. A., & Liu, W. K. (2016). Self-consistent clustering analysis: An efficient multi-scale scheme for inelastic heterogeneous materials. *Computer Methods in Applied Mechanics and Engineering*, *306*, 319–341. https://doi.org/10.1016/j.cma.2016.04.004

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2