Software Impacts 18 (2023) 100595

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

NOCAL-FEA: A NonlOCAL results processor for Finite Element Analysis

John A. Moore [a,*], Caitlin Martinez [a], Ayushi Chandel [b]

[a] Marquette University, 1515 W Wisconsin Ave, Milwaukee, WI 53233, United States of America
[b] Brookfield Academy, 3462 N Brookfield Rd, Brookfield, WI 53045, United States of America

ARTICLE INFO

Keywords:
Engineering
Nonlocal
Finite element analysis
Python
Parallelized

ABSTRACT

In engineering, thermal, and mechanical field quantities (i.e., stress, deformation, temperature) are calculated at every point in a complex structure to ensure quality performance before costly manufacturing. These calculations are often performed using finite element analysis. However, for determination of some performance metrics (usually relating to fracture), a local measure at every point is insufficient—as a larger (nonlocal) region of the structure affects values at a single point. The code here calculates nonlocal results without modifying the finite element software source code. The code is parallelized for large calculations typical of finite element analysis problems.

Code metadata

Current code version	*v.0.1.0*
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2023-430
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/2919660/tree/v1
Legal Code License	GNU General Public License (GPL)
Code versioning system used	git
Software code languages, tools, and services used	Python, Dask
Compilation requirements, operating environments & dependencies	Matplotlib, Numpy
Link to developer documentation/manual	https://github.com/johnallanmoore/NOCAL-FEA/docs
Support email for questions	john.a.moore@marquette.edu

1. Introduction

1.1. Motivation

In engineering, thermal, and mechanical field quantities (i.e., stress, deformation, temperature) are calculated at every point in a complex structure (i.e., part, component, device, etc.) to ensure quality performance before costly manufacturing. The software here is designed to allow engineers to calculate nonlocal field quantities (usually stress, strain, or related quantities).

To determine if a structure will fail, the most important mechanical field quantities are stress and strain. Stress (σ) is a force (F) per area (A) where $\sigma = F/A$; strain (ε) is a change in length (ΔL) per length L where $\varepsilon = \Delta L/L$. Two examples illustrate how these quantities relate to fracture. (1) Your weight (force) is the same whether you stand on a book or a nail, but the pain is much higher when your weight is distributed over the low area nail. (2) If a machine were to stretch a 100 m steel girder 1 cm it would not fail, but if a machine stretched a 1 cm slice of steel wire an additional 1 cm, it would fail. Thus stress/strain rather than force/displacement govern failure. In finite element analysis stress/strain are calculated in elements; whereas force/displacement are calculated at nodes (see Figs. 1 and 2 for examples of elements and nodes).

Stress and strain are important in determining if a structure will fail under a static load (fracture) or fail under repeated cyclic loads (fatigue). Increasingly, engineers use software, analysis, and experiments to design structural (i.e., load bearing) materials for specific applications. This is called Integrated Computational Materials Engineering (ICME) [1–3]. These designers look under a microscope and calculate

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail address: john.a.moore@marquette.edu (J.A. Moore).

https://doi.org/10.1016/j.simpa.2023.100595
Received 2 October 2023; Received in revised form 22 October 2023; Accepted 25 October 2023

2665-9638/© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

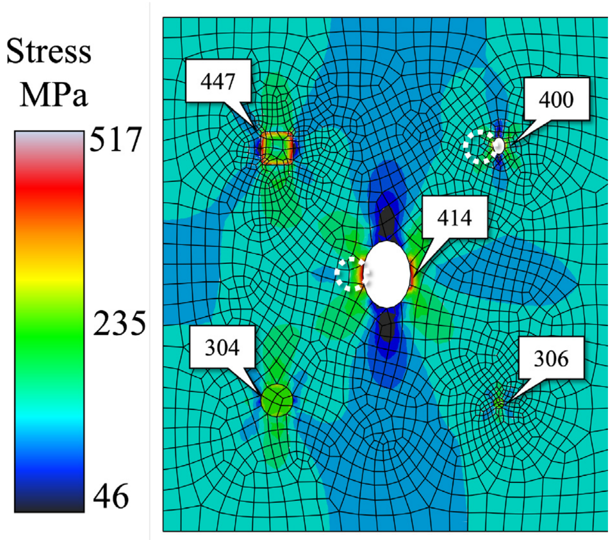

Fig. 1. Stress profile in an example microstructure modeled using FEA. The microstructure contains two voids and three defects of dissimilar material each with the stress value annotated. The white dashed circles are nonlocal volumes.

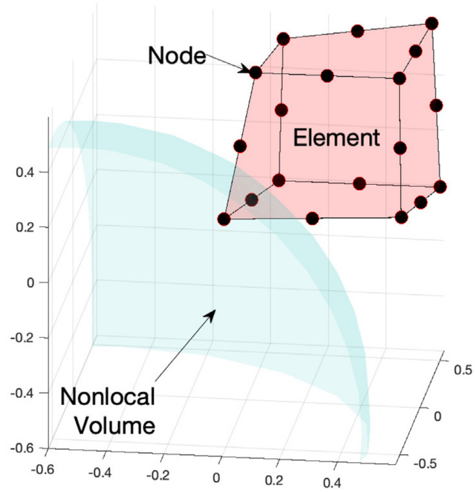

Fig. 2. A 20-node finite element and the boundary of a nonlocal volume.

the stress around material imperfections they see. From this information they can process the material differently to avoid the defects that most limit a material's life.

A very common approach to calculating stress and strain is finite element analysis (FEA) [4]. FEA is particularly well suited for modeling structures with complex shapes [4]. Rather than produce a stress profile that has infinite resolution, FEA divides a structure into a finite number of *elements* over which stress/strain are calculated. These elements are illustrated by a multitude of quadrilateral shapes in Fig. 1. The entire set of elements is called a *mesh*.

In fracture analysis, it is known that larger material defects, impurities, pores, or cracks will drive a structure to failure more readily than smaller features [5–8]. Fig. 1 shows five different microstructural features that one might see under a microscope in a structural material: two pores (voids), and three inclusions of dissimilar material. The square inclusion has a different stress than the round inclusions; and the pores have different stresses than the round inclusions. However, the stresses between round pores are very similar. The same is true for inclusions. For example, the large void has a 3% higher stress than the small void; however, it is known from the references above that the

Table 1
Connectivity matrix example.

Element number	Node #1	Node #2	…	Node #n
1	1,	2,	3,4,5,6,7,	8
2	5,	6,	7,8,9,10,11,	12

Table 2
Node position example.

Node number	x	y	z
1	0.0,	0.0,	0.0
2	0.1,	0.0,	0.0

driving force for crack formation of the large void is much greater than 3% of that of the small void. However, if the stress is average over a constant volume (see dashed lines in Fig. 1) then the large void gives a higher nonlocal stress than the small void—which is more consistent with observation. The software here calculates these nonlocal volumes, such that they can be used in FEA, and averages results over these volumes.

1.2. Nonlocal quantity of interest

Many nonlocal quantities related to stress and strain can be of interest for fracture analysis [8–10]. The software here calculates a function of both stress and strain known as a fatigue indicator parameter (FIP) but any nonlocal output of FEA could be calculated. The FIP is given by:

$$FIP = \frac{\Delta\epsilon^p}{2}\left(1+\kappa\frac{\sigma^n}{\sigma_y}\right), \tag{1}$$

where $\Delta\epsilon^p$ is the change in plastic strain over a fatigue cycle, σ^n is the stress normal to a critical plane on which a crack will form, σ_y is the yield stress of the material and κ is a calibration parameter.

1.3. FEA software

Many FEA software exist. To name only a few: Ansys, Abaqus, Comsol, and Nastran are commercially available, ALE3D and Sierra are government owned, and MFEM, FEniCS, and JAX are open source. However, the user rarely has access to source code for commercial software, government software is restricted in use, and open-source software can be difficult for non-software engineers to modify while also focusing on broader engineering goals. The software here not only calculates nonlocal volumes, but it also performs this task without modifying the source code of the FEA software. The current code examples are for Abaqus but any FEA software could be used to generate the mesh and run the analysis.

2. Methods

The goal of the software is to efficiently determine a nonlocal volume of elements surrounding any given single finite element. In FEA, every finite element has several associated nodes at the corners or along edges of elements as shown in Fig. 2. Each node represents a set of equations that must be solved by FEA software. So, for a single finite element there will be several nodes with different positions in space.

FEA codes hold element and node information in an array known as a connectivity matrix as shown in Table 1. The associated node positions are then stored in a separate array shown in Table 2.

The software here includes a method to extract the connectivity matrix from an Abaqus input deck; however, this matrix can be extracted from any FEA software.

Determining which elements to include in the nonlocal volume shown in Fig. 2 requires establishing the nearest neighbor elements to every element in a mesh. Therefore, using the connectivity matrix and node position array only, a code must search the position of every node in every element and establish if a majority of nodes in any given element are within the nonlocal volume. This is then repeated for every element in the mesh which becomes computationally inefficient—especially for the common 20-node elements as shown in Fig. 2. Fig. 2 also shows that simply establishing if a single node is in the nonlocal volume might greatly overestimate the number of elements that should lie in the volume. Thus, the more efficient approach employed here is to first find the geometric centroid of each element, then store this single value in a matrix and search the position of element centroids rather than position of several nodes.

Finally, determining the nonlocal volume for each element is an *embarrassingly* parallel task. Thus, the software here is written for parallelization using Python's `multiprocessing` module [11,12]. Further parallelization uses the open-source Python library Dask [13] (specifically the `dask.distributed` library) which can scale to a large number of high-performance computing nodes for large meshes, but this version has yet to be released.

3. Code usage

NOCAL-FEA is controlled by a python dictionary where all the user inputs (i.e., *cards*) are stored. This file must be called dictionary.py .

In pre-processing the user can choose to extract the connectivity matrix from an Abaqus input using the `deckName` card. The user can set the `dx` card to modify the length of an edge of cubic nonlocal volumes, for spherical nonconformal volumes NOCAL-FEA will convert this value into a sphere with an equivalent volume to the cube. The user can choose whether to employ a cubic conformal or a spherical nonconformal volume using the `isNonConf` card and also choose how many processors to use in determining volumes. The determined nonlocal volumes can also be visualized by storing them as images via the `plotVolumes` and `vol2Plot` cards.

For post-processing the user can choose a weighted or non-weighted volume average (see [14]). The user can also choose to use only one of these operations at a time via the `runOnly` card.

4. Impact overview

This software allows the user to ask *scientific questions* about nonlocal length scales, nonlocal volumes, etc. and receive *answers* quickly, because many types of volumes are easily folded into FEA. As shown in [14], the nonconformal volumes produced by this software also allow for a more accurate prediction of the spread in statistical fatigue life data.

The software here can determine conformal volumes (that do not overlap) or nonconformal volumes (that do overlap). This software is being used extensively on the project: Determining the Driving Force for Fatigue Crack Nucleation in a Superelastic Nickel Titanium Alloy (see Acknowledgements). The conformal version is used in [15–17] and the nonconformal version is used in [14].

Materials design companies such as QuesTek Innovations often use computational FEA models to steer designs. The software here is useful in that process, specifically for design of fatigue resistant materials, as in [18–23]. The authors have previously collaborated with QuesTek [24,25] and future collaborations could support their use of the software here.

Specifically for analyzing mechanical fatigue, the software here has potential for industrial impact in the area of biomedical fatigue resistant devices. Companies such as Confluent Medical (and even government entities, such as the U.S. Food and Drug Administration [26, 27]), often perform research in the area of microscale or microstrain analysis using FEA (where nonlocal fatigue analysis is needed [26–28]). As of 2023, the Google Scholar search for the exact phrase "Fatigue Indicator Parameter" yielded 412 results [29] which illustrates the potential for impact of the software here on fatigue analysis.

This software also has the propensity for incorporation (or at least coupled usage) with other engineering software. The Thermocalc Additive Manufacturing Module "Allows you to calculate ... size of melt pool [and]...Property variations through the melt pool (viscosity, thermal conductivity, density)" [30] of which size dependance necessitates nonlocal analysis facilitated by this software. The software JAX AM [31,32] which "runs on CPU/GPU with high performance ... [and] is natural for research of AI for science/engineering, because JAX is intended for machine learning research" [33] and The Exascale Project [34] both model microstructural features where nonlocal effects are present and can benefit from the scalable nonlocal tools in the software here. Finally, Abaqus FEA software has user defined field `usdfld` capabilities which "allows users to define field variables at material points as a function of time" [35]; the nonlocal volumes created by the software could be incorporated in a real-time simulation process (as opposed to post-process). This would allow for real-time nonlocal calculations (as in [8]) using `usdfld`. Neper software [36–38] – from which the microstructures used in the current software were created – could also incorporate the software here's nonlocal volumes directly.

As a result of the software here, new research questions in several fields can be pursued. As shown in Fig. 1, those researching the structure–property relationship in materials can use nonlocal volumes to apply size-effect to microstructural feature and determine the resulting fatigue life accurately. Also, nonlocal modeling (and thus, this software) is valuable to research in other disciplines in engineering and science such as: "geomechanics" [39], "image analysis" [39], "signal and image processing and medical imaging digital image correlation" [39], "discrete data analysis" [39], "inverse problems" [39], and "mathematical biology" [39].

This software will also be developed in the future by the authors. The focus of future development is on scalability and extensibility. As mentioned, ideally the software would scale to an unlimited number of computing nodes for large FEA meshes. This work is already in progress, and code with `dask` implemented has been *committed* to the repo in Code metadata. However, this `dask` code has yet to be released. Future implementation with `dask` will focus on user-friendly environments which load the required `openssl` toolkit [40] and memory management for scaling to a large number of nodes.

Other future development by the authors will focus on extensibility. First a graphical user interface (GUI) will be developed using Tkinker to increase ease-of-use for those more comfortable with software GUIs. Currently, a `writeResults` branch has been created for implementation of verification tools. These tools will produce sample results data that will allow users to check their code (i.e., these results will allow the user to know what the nonlocal results should be). This will hopefully make it easier to implement new features such as alternative averaging schemes and nonlocal volume shapes. Finally, the authors also study porosity and ductile failure modeling [41–43], and would like to extend this work to porosity evolution for fracture problems which has been shown to be nonlocal [8].

CRediT authorship contribution statement

John A. Moore: Conceptualization, Writing – original draft, Software. **Caitlin Martinez:** Methodology, Software. **Ayushi Chandel:** Software, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This material is based upon work supported by the National Science Foundation, United States of America under Grant No. Award Number 1934753. Support was provided by the Division of Civil, Mechanical and Manufacturing Innovation's (CMMI) Mechanics of Materials and Structures (MOMS) program, United States of America. The authors would like to thank Dr. Dinc Erdeniz of the University of Cincinnati for his discussion about and support of this work.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.simpa.2023.100595.

References

[1] M.F. Horstemeyer, Integrated Computational Materials Engineering (ICME) for Metals: Concepts and Case Studies, John Wiley & Sons, 2018.
[2] W.Y. Wang, J. Li, W. Liu, Z.-K. Liu, Integrated computational materials engineering for advanced materials: A brief review, Comput. Mater. Sci. 158 (2019) 42–48.
[3] M.F. Horstemeyer, Integrated Computational Materials Engineering (ICME) for Metals: Using Multiscale Modeling To Invigorate Engineering Design with Science, John Wiley & Sons, 2012.
[4] T. Belytschko, W.K. Liu, B. Moran, K. Elkhodary, Nonlinear Finite Elements for Continua and Structures, John Wiley & Sons, 2014.
[5] Z.P. Bažant, Y. Xi, Statistical size effect in quasi-brittle structures: II. nonlocal theory, J. Eng. Mech. 117 (11) (1991) 2623–2640.
[6] Z.P. Bazant, J. Planas, Fracture and Size Effect in Concrete and Other Quasibrittle Materials, Vol. 16, CRC Press, 1997.
[7] R. Kumar, A.J. Wang, D. McDowell, Effects of microstructure variability on intrinsic fatigue resistance of nickel-base superalloys–a computational micromechanics approach, Int. J. Fract. 137 (2006) 173–210.
[8] K. Enakoutsa, J. Leblond, G. Perrin, Numerical implementation and assessment of a phenomenological nonlocal model of ductile rupture, Comput. Methods Appl. Mech. Engrg. 196 (13–16) (2007) 1946–1957.
[9] N. Fleck, J. Hutchinson, A phenomenological theory for strain gradient effects in plasticity, J. Mech. Phys. Solids 41 (12) (1993) 1825–1857.
[10] J.A. Moore, Y. Li, D.T. O'Connor, W. Stroberg, W.K. Liu, Advancements in multiresolution analysis, Internat. J. Numer. Methods Engrg. 102 (3–4) (2015) 784–807.
[11] Python Software Foundation, Multiprocessing–process-based parallelism, 2023, https://docs.python.org/3/library/multiprocessing.html.
[12] J.A. Moore, Nocal-fea: Nocal-fea! v.0.1.0, 2023, http://dx.doi.org/10.5281/zenodo.8323192.
[13] Dask Development Team, Dask: Library for dynamic task scheduling, 2016, https://dask.org.
[14] J. Moore, C. Martinez, A nonconformal nonlocal approach to calculating statistical spread in fatigue indicator parameters for polycrystals, Fatigue Fract. Eng. Mater. Struct. http://dx.doi.org/10.1111/ffe.14158, in-press.
[15] J.A. Moore, J.P. Rusch, P.S. Nezhad, S. Manchiraju, D. Erdeniz, Effects of martensitic phase transformation on fatigue indicator parameters determined by a crystal plasticity model, Int. J. Fatigue 168 (2023) 107457.
[16] J.A. Moore, J.P. Rusch, P.S. Nezhad, S. Manchiraju, D. Erdeniz, Effects of martensitic phase transformation on fatigue indicator parameters determined by a crystal plasticity model, in: 17th US National Congress on Computational Mechanics, Albuquerque, NM, 2023.
[17] J. Rusch, J.A. Moore, Using finite element analysis and bayesian machine learning to make predictions of crack nucleation in metals with non-metallic inclusions, in: Mechanistic Machine Learning and Digital Twins for Computational Science, Engineering & Technology, San Diego, CA, 2021.
[18] J. Sebastian, G. Olson, Examples of questek innovations' application ICME to materials design, development, and rapid qualification, in: 55th AIAA/ASMe/ASCE/AHS/SC Structures, Structural Dynamics, and Materials Conference, 2014, p. 0797.
[19] P. Nezhadfar, S. Thompson, A. Saharan, N. Phan, N. Shamsaei, Structural integrity of additively manufactured aluminum alloys: Effects of build orientation on microstructure, porosity, and fatigue behavior, Addit. Manuf. 47 (2021) 102292.
[20] M. Muhammad, P. Nezhadfar, S. Thompson, A. Saharan, N. Phan, N. Shamsaei, A comparative investigation on the microstructure and mechanical properties of additively manufactured aluminum alloys, Int. J. Fatigue 146 (2021) 106165.
[21] S. Vunnam, A. Saboo, C. Sudbrack, T.L. Starr, Effect of powder chemical composition on the as-built microstructure of 17-4 PH stainless steel processed by selective laser melting, Addit. Manuf. 30 (2019) 100876.
[22] S.M. Orzolek, J.K. Semple, C.R. Fisher, Influence of processing on the microstructure of nickel aluminum bronze (nab), Addit. Manuf. 56 (2022) 102859.
[23] Y. Du, R. Schmid-Fetzer, J. Wang, S. Liu, J. Wang, Z. Jin, Computational Design of Engineering Materials: Fundamentals and Case Studies, Cambridge University Press, 2023.
[24] J.A. Moore, A Micromechanics-Based Method for Multiscale Fatigue Prediction (Ph.D. thesis), Northwestern University, 2015.
[25] J.A. Moore, D. Frankel, R. Prasannavenkatesan, A.G. Domel, G.B. Olson, W.K. Liu, A crystal plasticity-based study of the relationship between microstructure and ultra-high-cycle fatigue life in nickel titanium alloys, Int. J. Fatigue 91 (2016) 183–194.
[26] H.M. Paranjape, K.I. Aycock, C. Bonsignore, J.D. Weaver, B.A. Craven, T.W. Duerig, A probabilistic approach with built-in uncertainty quantification for the calibration of a superelastic constitutive model from full-field strain data, Comput. Mater. Sci. 192 (2021) 110357.
[27] I. Carr, K. Aycock, C. Bonsignore, H. Paranjape, J. Weaver, B. Craven, Two-Step Uncertainty Quantification Methodology for Medical Device Design: Influence of Input Parameter Probability Distribution on Output Uncertainty, Bulletin of the American Physical Society.
[28] K. Senthilnathan, A. Shamimi, C. Bonsignore, H. Paranjape, T. Duerig, Effect of prestrain on the fatigue life of superelastic nitinol, J. Mater. Eng. Perform. 28 (2019) 5946–5958.
[29] Google scholar, 2023, https://scholar.google.com/scholar?hl=en&as_sdt=0%2C50&q=%22Fatigue+Indicator+Parameter%22&btnG=. (Accessed: 08 September 2023).
[30] Thermo-Calc Software, Powder bed fusion additive manufacturing module, 2023, https://thermocalc.com/products/add-on-modules/additive-manufacturing-module/.
[31] T. Xue, S. Liao, Z. Gan, C. Park, X. Xie, W.K. Liu, J. Cao, JAX-FEM: A differentiable GPU-accelerated 3D finite element solver for automatic inverse design and mechanistic data science, Comput. Phys. Comm. (2023) 108802.
[32] J.Y. Choi, T. Xue, S. Liao, J. Cao, Accelerating phase-field simulation of three-dimensional microstructure evolution in laser powder bed fusion with composable machine learning predictions, Available at SSRN 4502223.
[33] T. Xue, Welcome to JAX-AM's documentation, 2022, https://jax-am.readthedocs.io/en/latest/.
[34] L. Diachin, A. Barker, Transforming additive manufacturing through exascale simulation objective, 2023, https://exascaleproject.org.
[35] Abaqus, ABAQUS/Standard User's Manual, Version 6.6, Dassault Systèmes Simulia Corp, 2009.
[36] R. Quey, P. Dawson, F. Barbe, Large-scale 3D random polycrystals for the finite element method: Generation, meshing and remeshing, Comput. Methods Appl. Mech. Engrg. 200 (17–20) (2011) 1729–1745.
[37] R. Quey, L. Renversade, Optimal polyhedral description of 3D polycrystals: Method and application to statistical and synchrotron x-ray diffraction data, Comput. Methods Appl. Mech. Engrg. 330 (2018) 308–333.
[38] R. Quey, A. Villani, C. Maurice, Nearly uniform sampling of crystal orientations, J. Appl. Crystallogr. 51 (4) (2018) 1162–1173.
[39] Journal of peridynamics and nonlocal modeling. https://www.springer.com/journal/42102.
[40] OpenSSL Project Authors, Cryptography and SSL/TLS toolkit, 2023, https://www.openssl.org/.
[41] J.A. Moore, S.F. Li, M. Rhee, N.R. Barton, Modeling the effects of grain and porosity structure on copper spall response, J. Dyn. Behav. Mater. 4 (2018) 464–480.
[42] J.A. Moore, A degradation function consistent with Cocks–Ashby porosity kinetics, Int. J. Fract. 209 (2018) 231–234.
[43] J.A. Moore, A. Frasca, A comparison of gurson and Cocks–Ashby porosity kinetics and degradation functions, Int. J. Fract. 229 (2) (2021) 253–268.