SoftwareX 15 (2021) 100775

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Software update

# Update 3.0 to "PuMA: The Porous Microstructure Analysis software", (PII:S2352711018300281)

Joseph C. Ferguson [a,*], Federico Semeraro [b], John M. Thornton [b], Francesco Panerai [c,d], Arnaud Borner [b], Nagi N. Mansour [b,d]

[a] Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, United States of America
[b] AMA Inc. at NASA Ames Research Center, Moffett Field, CA 94035, United States of America
[c] Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, IL 61801, United States of America
[d] Center for Hypersonics and Entry Systems Studies, University of Illinois at Urbana-Champaign, IL 61801, United States of America

A R T I C L E I N F O

*Article history:*
Received 16 July 2021
Accepted 19 July 2021

*Keywords:*
Microtomography
Tortuosity
Permeability
Conductivity
Segmentation
Microscale modeling

A B S T R A C T

A major update of the Porous Microstructure Analysis (PuMA) software is presented. PuMA is a framework for computing effective material properties and response based on material microstructures. Version 3.0 of the software extends the PuMA capabilities to include computation of anisotropic conductivity, elasticity, permeability, per-voxel material or fiber orientation estimation, as well as expanded computational material generation capabilities. Version 3.0 also introduces *pumapy*, a Python version of the PuMA software.



## Code metadata

| | |
|---|---|
| Current code version | 3.0 |
| Permanent link to code/repository used for this code version | https://github.com/ElsevierSoftwareX/SOFTX-D-21-00131 |
| Code Ocean compute capsule | N/A |
| Legal Code License | NASA Open Source License v1.3 |
| Code versioning system used | git |
| Software code languages, tools, and services used | C++, Python, Qt, Paraview, FEniCS |
| Compilation requirements, operating environments & dependencies | Linux or MacOS |
| If available Link to developer documentation/manual | https://puma-nasa.readthedocs.io |
| Support email for questions | joseph.ferguson@stanford.edu<br>federico.semeraro@nasa.gov |

## 1. Introduction

Advances in high-resolution 3D imaging techniques, such as X-ray micro-Computed Tomography (μ-CT) and Focused Ion Beam Scanning Electron Microscopy, have opened the door for the numerical simulation of material response and effective properties based directly on materials' meso- and microstructures. The Porous Microstructure Analysis software (PuMA) has been under development at the NASA Ames Research Center since 2014 in order to provide a computational framework for material modeling from tomographic voxel data. In the original publication [1], PuMA v2.1 was able to import or generate microstructural data, create interactive 3D visualizations, perform basic representative volume analysis, and compute material properties, including volume fraction, specific surface area, effective conductivity, and continuum and rarefied tortuosity [2]. Additionally, PuMA v2.1 included a particle-based decomposition model, relevant to high-temperature material ablation [3].

In this update, we present a major refactoring of the software and the addition of new capabilities, including anisotropic material analysis, computation of Darcy's permeability, and improved microstructure generation capabilities. Notably, a new Python distribution of the software was added, named *pumapy*, and the overall architecture was refactored to achieve ease of use and collaboration.

DOI of original article: https://doi.org/10.1016/j.softx.2018.03.001.
* Corresponding author.
*E-mail address:* joseph.ferguson@stanford.edu (Joseph C. Ferguson).

https://doi.org/10.1016/j.softx.2021.100775
2352-7110/

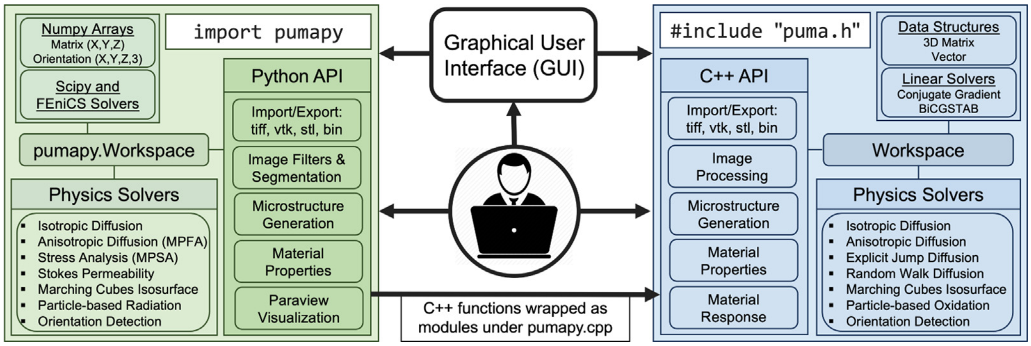


**Fig. 1.** PuMA's software architecture.

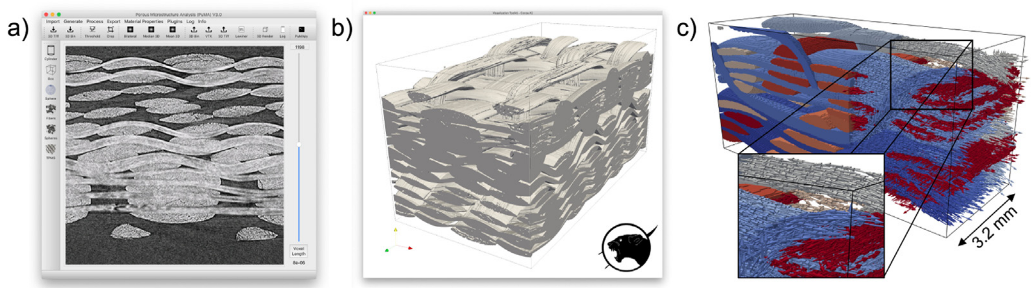


**Fig. 2.** μ-CT woven TPS imported in the PuMA GUI (a), 3D rendered (b), and post-processed with tow segmentation and orientation detection (c).

## 2. Porous microstructure analysis in Python: pumapy

The main C++ PuMA code has been expanded with a Python package, pumapy, which includes all of the features available in the PuMA software. The majority of these features are implemented directly in Python, while some are available as Python wrappers of the C++ functionalities. By taking advantage of the wide array of open source libraries in Python, pumapy allows for more rapid development and testing of new physics capabilities. For example, pumapy utilizes scikit-image [4] for basic 3D image manipulation, Numpy [5] and scipy [6] for fast matrix operations and sparse linear solvers, Paraview [7] for visualization, FEniCS [8] for Finite Element Analysis and Detectron2 [9] for image segmentation. The Python interface also allows users to easily perform statistical analysis, such as uncertainty quantification (UQ) and representative elementary volume (REV) analysis based on pumapy simulations.

## 3. Anisotropic analysis

PuMA v3.0 integrates tools for the analysis of anisotropic materials. In image-based simulations, anisotropy can exist at multiple scales. For instance, the effective properties of a material can be anisotropic due to its mesoscale architecture (e.g. a weave), even if all the constituents at the microscale are isotropic. However, for most materials, such as fiber-based architectures, properties are anisotropic even at the fiber level; take carbon fibers as an example, which have vastly different thermal conductivity values parallel and normal to the fiber direction [10]. For thermal or mechanical modeling based on μ-CT, it is critical to capture this multiscale anisotropy with accuracy, by identifying the local material orientation at each voxel, and in turn solve the governing equations accounting for the local anisotropy.

### 3.1. Material orientation estimation

Three methods were implemented in PuMA to estimate, from a μ-CT scan, the per-voxel local orientation of a fibrous or woven material. A structure tensor method based on grayscale gradients, an isotropic diffusion method that determines the dominant local flux direction, and a ray-casting method that estimates the material orientation by the longest line-of-sight. The detailed implementation and critical assessment of these methods for different fiber architectures can be found in Semeraro et al. [11]

### 3.2. Conductivity and elasticity

A solver for the conductivity of a material based on a microstructure with anisotropic constituents has been implemented into PuMA v3.0. [12]. The solver utilizes the Multi-Point Flux Approximation (MPFA-O) [13] approach and the BiCGSTAB iterative method to solve the steady-state conduction equation. The local material orientation and anisotropic local conductivities are used to homogenize the composite material. In addition to solid conductivity, an important component of the total conductivity is radiation, which becomes prevailing at high temperatures. For this reason, a ray-tracing method has been added to PuMA in order to statistically estimate the optical radiative extinction coefficient of the μ-CT reconstructions, as described in [14].

The micromechanical properties of a μ-CT domain can also be estimated through PuMA's linear static solver. The finite volume stress analysis approach is based on the Multi-Point Stress Approximation with weak symmetry (MPSA-W) [15]. Using a similar concept as the MPFA-O, the MPSA-W computes a homogenized elasticity tensor, as well as stress and deformation fields, based on the input voxel microstructure.

## 4. Permeability

The permeability quantifies the resistance of a material to a pressure-driven flow, and is a critical property in volume-averaged modeling of momentum transport in porous media. In PuMA v3.0, the permeability of a material is calculated by solving the Stokes equation with finite element analysis, using the FEniCS [8] library. For a given pressure gradient or forcing function, the pore-resolved solution to the Stokes equation yields a mean flow rate, which can be inserted into Darcy's law to calculate the effective permeability. Two discretization methods are implemented in PuMA: the first uses first-order finite elements in both pressure and velocity, which requires the addition of a stabilization term to the mixed variational form of Stokes equation; the second approach uses Taylor-Hood second-order elements in the pressure, which results in a more stable scheme and a higher computational requirement. Dirichlet or periodic boundary conditions can be used in the simulation direction, and free-slip, no-slip, or periodic boundary conditions can be used for the side boundary conditions.

## 5. Generation of synthetic microstructures

The ability to generate artificial materials has been expanded in PuMA v3.0 to include fibrous structures with complex cross-sections and hollow regions, triply periodic minimal surfaces (TPMS), and advanced woven structures via the coupling with TexGen's Python API [16].

## 6. Software architecture updates

PuMA's architecture was re-designed to run on a variety of computing systems, ranging from laptops to workstations and supercomputing clusters. The installation of the C++ PuMA library, pumapy Python package and GUI (Fig. 2) are handled through the Conda package management system, which alleviates the need of linking dependencies. Compared to v2.1, the v3.0 refactoring features a highly modular architecture to promote modifications and contributions by outside researchers (Fig. 1). A series of tutorials in the form of Jupyter notebooks and C++ examples have been added to the repository, covering all software functionalities, in order to accelerate learning of the software and development by new users.

## 7. Conclusions and outlook

PuMA v3.0 presents a full refactoring and open source release of the software, designed for ease of use and contribution. The Python version of PuMA, called pumapy, is also introduced. The updated PuMA software includes solvers for the effective permeability, anisotropic analysis, and improved material generation capabilities.

Future releases will include a weave machine learning segmentation technique, built upon the Detectron2 library and a technique called video panoptic segmentation [17] to segment and track the tow instances in 3D space.

## 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 work was supported by the NASA Entry System Modeling project, United States of America (M.D. Barnahrdt project manager, A.M. Brandis principal investigator). The work was also supported by the NASA Space Technology Research Fellowship program, United States of America (Award No. 80NSSC19K1134). F. S., J. T. and A.B are supported by NASA, United States of America contract NNA15BB15C to Analytical Mechanical Associates, Inc. Sergio Fraile Izquierdo, Sander Visser and Jeremie Meurisse (AMA Inc. at NASA Ames Research Center) are acknowledged for their review of the manuscript and useful discussions.

## References

[1] Ferguson JC, Panerai F, Borner A, Mansour NN. PuMA: the porous microstructure analysis software. SoftwareX 2018;7:81–7.
[2] Panerai F, Ferguson JC, Lachaud J, Martin A, Gasch MJ, Mansour NN. Micro-tomography based analysis of thermal conductivity, diffusivity and oxidation behavior of rigid and flexible fibrous insulators. Int J Heat Mass Transfer 2017;108:801–11.
[3] Ferguson JC, Panerai F, Lachaud J, Martin A, Bailey SC, Mansour NN. Modeling the oxidation of low-density carbon fiber material based on micro-tomography. Carbon 2016;96:57–65.
[4] Van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, et al. Scikit-image: Image processing in Python. PeerJ 2014;2:e453.
[5] Van Der Walt S, Colbert SC, Varoquaux G. The NumPy array: A structure for efficient numerical computation. Comput Sci Eng 2011;13(2):22–30.
[6] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods 2020;17(3):261–72.
[7] Ahrens J, Geveci B, Law C. Paraview: An end-user tool for large data visualization. Vis Handb 2005;717(8).
[8] Alnæs M, Blechta J, Hake J, Johansson A, Kehlet B, Logg A, et al. The FEniCS project version 1.5. Arch Numer Softw 2015;3(100).
[9] Wu Y, Kirillov A, Massa F, Lo W-Y, Girshick R. Detectron2. 2019, https://github.com/facebookresearch/detectron2.
[10] Grujicic M, Zhao C, Dusel E, Morgan D, Miller R, Beasley D. Computational analysis of the thermal conductivity of the carbon–carbon composite materials. J Mater Sci 2006;41(24):8244–56.
[11] Semeraro F, Ferguson JC, Panerai F, King RJ, Mansour NN. Anisotropic analysis of fibrous and woven materials part 1: Estimation of local orientation. Comput Mater Sci 2020;178:109631.
[12] Semeraro F, Ferguson JC, Acin M, Panerai F, Mansour NN. Anisotropic analysis of fibrous and woven materials part 2: Computation of effective conductivity. Comput Mater Sci 2021;186:109956.
[13] Aavatsmark I. An introduction to multipoint flux approximations for quadrilateral grids. Comput Geosci 2002;6(3):405–32.
[14] Owiti BO, Sakai T, Ishida Y. Radiative transfer in a low-density ablative material under arcjet flow conditions. J Thermophys Heat Transfer 2020;34(1):181–92.
[15] Keilegavlen E, Nordbotten JM. Finite volume methods for elasticity with weak symmetry. Internat J Numer Methods Engrg 2017;112(8):939–62.
[16] Lin H, Brown LP, Long AC. Modelling and simulating textile structures using TexGen. In: Advanced materials research. vol. 331. Trans Tech Publ; 2011. p. 44–7.
[17] Kim D, Woo S, Lee J-Y, Kweon IS. Video panoptic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020. p. 9859–9868.