Contents lists available at ScienceDirect

SoftwareX

Original software publication

PuMA: the Porous Microstructure Analysis software

Joseph C. Ferguson [a,*], Francesco Panerai [b], Arnaud Borner [a], Nagi N. Mansour [c]

[a] *Science and Technology Corp., NASA Ames Research Center, Moffett Field, CA 94035, United States*
[b] *AMA Inc., NASA Ames Research Center, Moffett Field, CA 94035, United States*
[c] *NASA Ames Research Center, Moffett Field, CA 94035, United States*

ARTICLE INFO

ABSTRACT

The Porous Microstructure Analysis (PuMA) software has been developed in order to compute effective material properties and perform material response simulations on digitized microstructures of porous media. PuMA is able to import digital three-dimensional images obtained from X-ray microtomography or to generate artificial microstructures. PuMA also provides a module for interactive 3D visualizations. Version 2.1 includes modules to compute porosity, volume fractions, and surface area. Two finite difference Laplace solvers have been implemented to compute the continuum tortuosity factor, effective thermal conductivity, and effective electrical conductivity. A random method has been developed to compute tortuosity factors from the continuum to rarefied regimes. Representative elementary volume analysis can be performed on each property. The software also includes a time-dependent, particle-based model for the oxidation of fibrous materials. PuMA was developed for Linux operating systems and is available as a NASA software under a US & Foreign release.

© 2018 Published by Elsevier B.V.

Code metadata

Current code version	2.1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-17-00084
Legal Code License	NASA US & Foreign Release
Code versioning system used	git
Software code languages, tools, and services used	C++, QT, OpenMP, FFTW, LibTiff, OpenGL.
Compilation requirements, operating environments & dependencies	Linux OS, GCC 4.4.7 or above, QT 4.6 or above
If available Link to developer documentation/manual	https://software.nasa.gov/software/ARC-17920-1
Support email for questions	joseph.c.ferguson@nasa.gov

1. Motivation and significance

X-ray microtomography (micro-CT) has become an established tool in material science [1,2], allowing for the non-destructive characterization of microstructures at scales from hundreds of nanometers to centimeters. The resulting digital representations of the three-dimensional material microstructures constitute a high fidelity framework for calculating effective material properties [3,4] and simulating material response [5].

The development of the Porous Microstructure Analysis (PuMA) software is motivated by the possibility to characterize, from tomographic images, microscale properties and response of ablative heat shield materials used on NASA spacecraft. These microscale based computations are used to quantify material properties used in macroscale material response models. [6]

A common challenge when dealing with 3D tomography images is the size of the datasets, often on the order of several billion voxels; a dedicated computational infrastructure is needed to analyze and run simulations based on these images. Several commercial software packages are available for microstructure analysis, including GeoDict [7] (Math2Market, Kaiserslautern, Germany), Avizo XLab [8] (FEI, Hillsboro, Oregon, USA) and Simpleware [9] (Synopsys, Mountain View, California, USA). Avizo XLab and Simpleware are popular image processing and visualization packages that have recently included the option of computing thermal, mechanical and morphological properties of material structures based on X-ray tomography. GeoDict is an extensive material design platform that enables the calculation of properties from either tomography or artificially generated geometries.

---

* Corresponding author.
E-mail address: joseph.c.ferguson@nasa.gov (J.C. Ferguson).

Open-source and academic software is also witnessing a rapid development. The TauFactor software by Cooper et al. [10] has the capability of computing tortuosity factors for continuum flows in porous media, and to perform representative elementary volume (REV) analysis. Another software, iMorph [11], can compute continuum tortuosity factors and morphological properties based on tomographic images. For strain analysis of in situ X-ray microtomography experiments, another open source software, TomoWarp2 [12], can determine displacement fields through 3D digital image correlation.

The listed commercial and academic software primarily focus on low temperature applications, outside the realm of simulation capabilities needed for entry-systems modeling, where high enthalpies and chemistry play a fundamental role. Additionally, commercial software lacks the flexibility and access to add custom solvers, perform advanced statistics (e.g. robust sensitivity analysis and uncertainty quantification), or couple with other simulation tools.

With this goal in mind, the Porous Microstructure Analysis (PuMA) software was developed at the NASA Ames Research Center in order to calculate material properties and response on large digital microstructures.

The latest version of PuMA is available from NASA under a US & Foreign release and includes the ability to import X-ray microtomography datasets; computationally generate a number of ideal geometries; visualize the materials 3D microstructure; determine effective material properties such as porosity, surface area, thermal conductivity and tortuosity factors; and perform representative elementary volume (REV) analysis for each property. A particle-based microscale oxidation model [5,13] was also implemented and parallelized [14] to simulate high-temperature oxidation of carbon fiber materials.

This paper presents an overview of the PuMA (Version 2.1) software. Section 2 covers the four overarching modules of PuMA's architecture, namely the domain generation module (Section 2.2), the visualization module (Section 2.3), the material properties module (Section 2.4) and the material response module (Section 2.6).

2. Software description

2.1. Software architecture

The functional structure of PuMA is shown in Fig. 1a. It consists of four major modules: domain generation, visualization, material properties, and material response. These are described in detail in Sections 2.2–2.6.

PuMA is written in C++ with a graphical user interface (GUI) built on Qt. A snapshot of the GUI is presented in Fig. 1b, showing the tomography import window. The visualization toolkit is based on OpenGL [15]. The current version of the software (Version 2.1) uses two third party libraries: LibTiff [16] is used to import tiff stacks from microtomography and the LibFFTW [17] is used for Fast Fourier Transform (FFT) operations. Every computationally intensive element in the code has been parallelized with OpenMP [18] for shared memory architectures.

The software is developed for Linux systems, optimized for multi-processor workstations.

2.2. Domain generation

Computational domains in PuMA can be constructed artificially or imported as a 3D TIFF (stack) from microtomography. PuMA does not currently include image filtering capabilities, though this will likely be incorporated into future versions. In our typical tomography pipeline, datasets that need filtering for artifact removal or edge detection enhancement are processed with the Fiji software [19] after tomographic reconstruction. Processed TIFF stacks can be imported in PuMA and a user defined subset can be selected for visualization and analysis. A simple histogram-based thresholding segmentation method is provided, where a single material phase can be isolated using a grayscale thresholding method. Fig. 1b provides an example in which a carbon-fiber microtomography sample is imported for analysis and a typical bimodal grayscale distribution is shown. In many cases, threshold-based segmentation techniques may not be appropriate. For such cases, segmentation can be performed in dedicated software that allow for more advanced methods such as random walk or machine learning algorithms. Examples of software with advanced segmentation features are Fiji [19], scikit-image [20], and Biomedisa [21].

The artificial material generator allows for the creation of random 3D fiber structures, packed sphere beds, periodic foam structures, as well as a series of ideal geometries, such as spheres, cylinders or slabs, used for testing purposes. Examples of artificial domains generated in PuMA are shown in Fig. 2.

2.3. Visualization

The visualization toolkit in PuMA provides an interactive 3D view of the computational domain. The visualization is based on the marching cubes algorithm [22,23], in which an approximate iso-surface is constructed as a collection of triangles. From this set of triangles, a surface rendering is generated using OpenGL, with triangle normals used for shading.

Fig. 3 shows subsets of two synchrotron X-ray tomography images of a carbon fiber material [24] collected at the Advanced Light Source of the Lawrence Berkeley National Laboratory, taken at different resolutions. In Fig. 3, the domain is visualized using a surface rendering in PuMA V2.1, with approximately 450 million triangles in the largest of the visualizations. The sample is shown at multiple scales, demonstrating the level of detail that the tomography is able to capture.

2.4. Material properties

In this section we describe a number of morphological, thermal, and transport properties that PuMA is able to compute from a given material microstructure (either tomographic or artificially generated).

2.4.1. Porosity and volume fraction

Based on the thresholding segmentation method, the volume fraction ϕ_i of a given phase i can be determined as a ratio of the number of voxels, N_i, in the phase to the total number of voxels, N_{tot}, in the domain: $\phi_i = N_i/N_{tot}$. The porosity, ϵ, is defined as the volume fraction of the void phase.

2.4.2. Specific surface area

The specific surface area (S) is defined as the ratio between the total surface area, and the volume, V. To numerically estimate the surface area, two methods [25] were included in the PuMA software: a voxel-based approach, where the total surface area is computed as the sum of exposed solid face areas of each voxel, and an iso-surface based approximation, where the total surface area is determined as a sum of the individual triangle areas in the computed triangulation.

$$S = \frac{\frac{1}{2}l_v^2 \sum_i | \vec{u}_i \times \vec{v}_i |}{V} \tag{1}$$

where $\vec{u}_i$ and $\vec{v}_i$ are the vectors defining triangle i, and l_v is the voxel length. The iso-surface approximation is generated with a marching cubes [22] or marching tetrahedra [26] algorithm. The