TauFactor 2: A GPU accelerated python tool for microstructural analysis

Steve Kench [1*], **Isaac Squires** [1*], **and Samuel Cooper** [1¶]

1 Imperial College London, United Kingdom ¶ Corresponding author * These authors contributed equally.

DOI: 10.21105/joss.05358

Software

- Review
- Repository
- Archive

Editor: Bonan Zhu

Reviewers:

- @alexsquires
- @ma-sadeghi

Submitted: 23 March 2023
Published: 30 July 2023

Summary

TauFactor 2 is an open-source, GPU accelerated microstructural analysis tool for extracting metrics from voxel based data, including transport properties such as the touristy factor. Tortuosity factor, τ, is a material parameter that defines the reduction in transport arising from the arrangement of the phases in a multiphase medium (see Figure 1). As shown in Equation 1, the effective transport coefficient of a material, D_{eff}, can be calculated from the phases intrinsic transport coefficient, D, volume fraction, ϵ, and τ (Cooper et al., 2016) (note, this value of τ should not be squared (Tjaden et al., 2016)).

$$D_{\text{eff}} = D\frac{\epsilon}{\tau} \tag{1}$$

Tortuosity factor has been a metric of interest in a broad range of fields for many of decades. In geophysics, τ influences groundwater flow through pourous rocks, which has significant environmental contamination impacts (Carey et al., 2016). Electrochemists use τ to solve a reduced-order system of equations describing the electrochemical behaviour of lithium-ion batteries, which influences a cells power rating (Landesfeind et al., 2018). The imaging and subsequent modeling of materials to determine τ is thus commonplace.

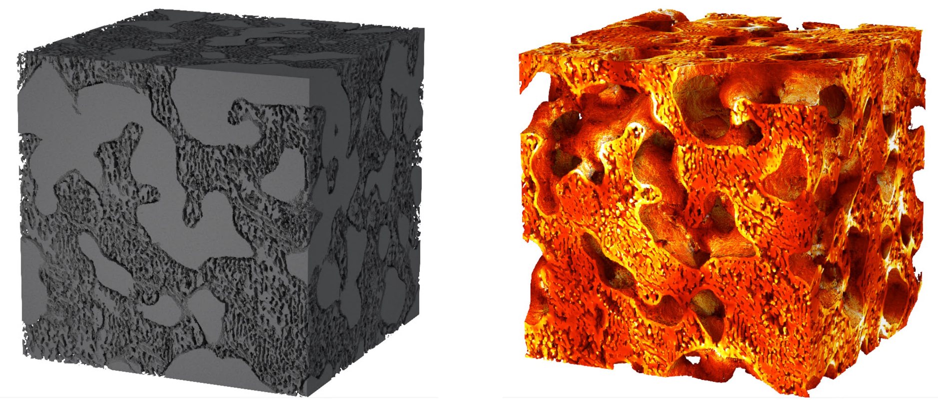

Figure 1: Microstructure and flux field of a sample from the microlib.io library (Kench et al., 2022).

Statement of need

Materials characterisation techniques are constantly improving, allowing the collection of larger field-of-view images with higher resolutions (Withers et al., 2021). Alongside these developments, machine learning algorithms have enabled the generation of arbitrarily large volumes, and can further enhance image quality through super-resolution techniques (Dahari et al., 2023). The resulting high fidelity microstructural datasets can be used to extract statistically representative metrics of a materials composition and performance. However, with increasing dataset size, the computational cost to perform such analysis can lead to prohibitively long run times. This is especially problematic for transport type metrics, such as the touristy factor, as they are inherently 3D and require the use of iterative solvers.

TauFactor 1 (Cooper et al., 2016) provided an open-source MATLAB application for calculating various microstructural metrics, including the touristy factor. However, its implementation as a serial CPU based solver meant that large microstructural dataset could take hours to converge. This made TauFactor 1 unsuitable for use in high-throughput tasks such as materials optimisation. TauFactor 2 provides the necessary efficiency to ensure users can analyse large datasets in reasonable times. The software is built with PyTorch (Paszke et al., 2019), a commonly used and highly optimised python package for machine learning. The GPU acceleration that has enabled the drastic speed up of neural network training proves equally effective for the task of iteratively solving transport equations, where matrix multiplication and addition are the main operations required. The use of Python and PyTorch ensures broad support and easy installation, as well as the option to run the software on CPU if GPU hardware is not available. The ability to run simulations with just a few lines of code ensures accessibility for researchers from the diverse fields where this software may be of use.

The Python implementation is similar to the original TauFactor 1, taking advantage of efficiency gains such as the precalculation of prefactors and the use of over-relaxation. The same convergence criteria are also used, where the vertical flux between each layer is averaged across the planes parallel to the stimulated boundaries. If the percentage error between the minimum and maximum flux is below a given value (default 1%), this indicates convergence has been reached. Once this is satsfied, an extra 100 iterations are performed to confirm the stability of the system. A notable difference in TauFactor 2 is that flux is calculated for all voxels. This replaces an indexing system in TauFactor 1, which solved only in active voxels. We find that the speed of GPU indexing compared to matrix multiplication makes this trade-off worthwhile. As well as the standard solver for a single transport phase, a multi-phase solver is available, where the tortuosity relates D_{eff} to the intrinsic diffusion coefficients and volume fractions of the various phases, p, as follows:

$$D_{\text{eff}} = \frac{\sum_p D_p \epsilon_p}{\tau} = \frac{D_{\text{mean}}}{\tau} \tag{2}$$

D_{mean} is a weighted sum of the active phase transport coefficients according to their volume fractions, which gives a transport coefficient equivalent to prismatic blocks of each phase spanning a test volume, in the direction of transport (i.e. perfectly straight transport paths). Periodic boundary conditions can also be used, which replace no-flux boundary conditions at the outer edges of the control volume. Finally, Taufactor 2 also includes an electrode tortuosity factor solver (see (Nguyen et al., 2020)). There are also GPU accelerated functions for calculating volume fractions, surface areas, triple phase boundaries and the two-point correlation function.

To compare the performance of TauFactor 2 to other available software, a test volume (500x500x500 = 125,000,000 voxels) was created. One of the phases in this two-phase volume fully percolates in all three directions, while the other phase does not percolate at all. The percolating phase has a volume fraction of exactly 30%. The percolating network is anisotropic in the three directions, leading to different transport metrics. Lastly, the structure is periodic at its boundaries, allowing for the exploration of the impact of periodic transport boundaries. This microstructure is available in the GitHub repository, providing a standard against which new software can also be measured. The speed of five different solvers, namely TauFactor 1 (Cooper et al., 2016), TauFactor 1.9 (an updated version of TauFactor 1, available here, that has new solvers such as diffusion impedance and can be called inline as well as from the GUI (Cooper et al., 2017)), TauFactor 2 (CPU), TauFactor 2 (GPU), PoreSpy (Gostick et al., 2019) and Puma (Ferguson et al., 2018), are shown in Figure 2. To check the accuracy of the calaculated tortuosity factors, we overconverge TauFactor 2 to give a 'true value' in each direction. Using default convergence criteria, all five solvers are within 0.5% of the true values other then PuMa's explicit jump solver (5% error), which is thus excluded (note it was still >2x slower than TF2). For this analysis we used a NVIDIA A6000 48GB GPU and AMD Ryzen Threadripper 3970X Gen3 32 Core TRX4 CPU. TauFactor 2 is over 10 times faster than the next best solver, TauFactor 1.9, and over 100 times faster than the original TauFactor 1 solver.

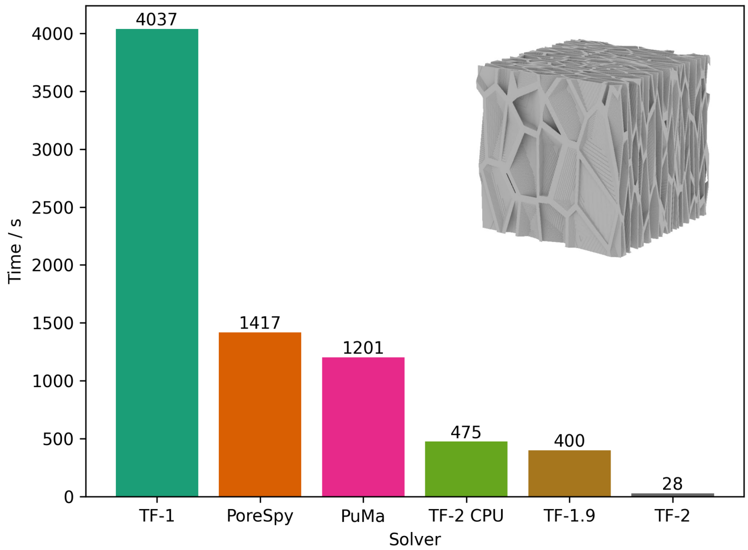

Figure 2: Speed comparison for the four solvers when applied to the test volume. The mean time across all 3 directions is plotted. The values of the overconverged τ in each direction are: 1.1513, 1.3905, 4.2431.

Authorship Contributions

SK wrote the base, periodic and multiphase solvers with input from IS. IS wrote the electrode solver, metric calculations and documentation, and also performed speed tests for other software packages. The project was supervised by SC, and based on his original MATLAB tool. All authors contributed to the writing and editing of the manuscript.

Acknowledgements

This work was supported by funding from the EPSRC Faraday Institution Multi-Scale Modelling project (https://faraday.ac.uk/; EP/S003053/1, grant number FIRG003 received by SK). We acknowledge contributions from Amir Dahari in the testing of the solver and python package installation.

References

Carey, G. R., McBean, E. A., & Feenstra, S. (2016). Estimating tortuosity coefficients based on hydraulic conductivity. *Groundwater*, *54*(4), 476–487. https://doi.org/10.1111/gwat.12406

Cooper, S. J., Bertei, A., Finegan, D. P., & Brandon, N. P. (2017). Simulated impedance of diffusion in porous media. *Electrochimica Acta*, *251*, 681–689. https://doi.org/10.1016/j.electacta.2017.07.152

Cooper, S. J., Bertei, A., Shearing, P. R., Kilner, J., & Brandon, N. P. (2016). TauFactor: An open-source application for calculating tortuosity factors from tomographic data. *SoftwareX*, *5*, 203–210. https://doi.org/10.1016/j.softx.2016.09.002

Dahari, A., Kench, S., Squires, I., & Cooper, S. J. (2023). Fusion of complementary 2D and 3D mesostructural datasets using generative adversarial networks (adv. Energy mater. 2/2023). *Advanced Energy Materials*, *13*(2), 2370009. https://doi.org/10.1002/aenm.202370009

Ferguson, J. C., Panerai, F., Borner, A., & Mansour, N. N. (2018). PuMA: The porous microstructure analysis software. *SoftwareX*, *7*, 81–87.

Gostick, J. T., Khan, Z. A., Tranter, T. G., Kok, M. D., Agnaou, M., Sadeghi, M., & Jervis, R. (2019). PoreSpy: A python toolkit for quantitative analysis of porous media images. *Journal of Open Source Software*, *4*(37), 1296. https://doi.org/10.1051/0004-6361/201629272

Kench, S., Squires, I., Dahari, A., & Cooper, S. J. (2022). MicroLib: A library of 3D microstructures generated from 2D micrographs using SliceGAN. *Scientific Data*, *9*(1), 645. https://doi.org/10.1038/s41597-022-01744-1

Landesfeind, J., Ebner, M., Eldiven, A., Wood, V., & Gasteiger, H. A. (2018). Tortuosity of battery electrodes: Validation of impedance-derived values and critical comparison with 3D tomography. *Journal of The Electrochemical Society*, *165*(3), A469–A476. https://doi.org/10.1149/2.0231803jes

Nguyen, T.-T., Demortière, A., Fleutot, B., Delobel, B., Delacourt, C., & Cooper, S. J. (2020). The electrode tortuosity factor: Why the conventional tortuosity factor is not well suited for quantifying transport in porous li-ion battery electrodes and what to use instead. *Npj Computational Materials*, *6*(1), 123. https://doi.org/10.1038/s41524-020-00386-4

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Tjaden, B., Cooper, S. J., Brett, D. J., Kramer, D., & Shearing, P. R. (2016). On the origin and application of the bruggeman correlation for analysing transport phenomena in electrochemical systems. *Current Opinion in Chemical Engineering*, *12*, 44–51. https://doi.org/10.1016/j.coche.2016.02.006

Withers, P. J., Bouman, C., Carmignato, S., Cnudde, V., Grimaldi, D., Hagen, C. K., Maire, E., Manley, M., Du Plessis, A., & Stock, S. R. (2021). X-ray computed tomography. *Nature Reviews Methods Primers*, *1*(1), 18.