The Journal of Open Source Software

MParT: Monotone Parameterization Toolkit

Matthew Parno [1,4¶], **Paul-Baptiste Rubio** [2], **Daniel Sharp** [2], **Michael Brennan** [2], **Ricardo Baptista** [2], **Henning Bonart** [2,3], **and Youssef Marzouk** [2]

1 Dartmouth College, Hanover, NH USA **2** Massachusetts Institute of Technology, Cambridge, MA USA **3** Technische Universität Darmstadt, Darmstadt, Germany **4** Solea Energy, Overland Park, KS USA ¶ Corresponding author

DOI: 10.21105/joss.04843

Software

- Review
- Repository
- Archive

Editor: Patrick Diehl

Reviewers:

- @ansgarwenzel
- @f-t-s

Submitted: 30 August 2022
Published: 14 December 2022

Summary

Measure transport is a rich area in applied mathematics that involves the construction of deterministic transformations–known as transport maps–between probability distributions (Santambrogio, 2015). These maps characterize a complex target distribution as a (deterministic) transformation of a simple reference distribution (e.g., a standard Gaussian). In the context of probabilistic modeling, transport maps enable easy generation of samples from the target distribution and direct evaluation of the target probability density function. Monotone triangular maps (Baptista et al., 2022) are a specific class of transport maps endowed with several computational advantages over non-triangular maps, such as easy invertibility and training, and yet sufficiently general to represent any absolutely continuous distribution; they are also the building block of many normalizing flow architectures commonly used in the machine learning community (Papamakarios et al., 2021).

Triangular maps are also well suited to many tasks in Bayesian inference, including the modeling of conditional distributions Spantini et al. (2018) and the acceleration of posterior sampling (Bigoni et al., 2016; Cotter et al., 2019; Moselhy & Marzouk, 2012; Parno & Marzouk, 2018). The fundamental idea is to convert the problem of characterizing a probability distribution through Monte Carlo sampling, variational inference, or density estimation into an optimization problem over multivariate monotone functions. The efficient solution of this optimization problem is especially important when using maps as part of online algorithms, as commonly encountered in sequential inference (Spantini et al., 2022).

In practice, working with triangular maps requires the definition of a parametric family of multivariate monotone functions. The Monotone Parameterization Toolkit (`MParT`), pronounced aims to provide performance-portable implementations of such parameterizations. `MParT` is a C++ library with bindings to Python, Julia, and Matlab that emphasizes fast execution and parsimonious parameterizations that can enable near real-time computation on low- and moderate-dimensional problems.

Statement of need

Several existing software packages have the ability to parameterize monotone functions, including TensorFlow Probability (Dillon et al., 2017), TransportMaps (Bigoni, 2015), ATM (Baptista et al., 2021), and MUQ (Parno et al., 2021). TensorFlow probability has a bijection class that allows deep neural network-based functions, such as normalizing flows (Papamakarios et al., 2021) to be easily defined and trained while also leveraging GPU computing resources if available but is focused on deep neural network parameterizations best suited for high dimensional problems. The TransportMaps, ATM, and MUQ packages use an alternative parameterization based on rectified polynomial expansions that is more compact and easier to train on low to moderate dimensional problems. At the core of these packages are scalar-valued functions $T_d : \mathbb{R}^d \to \mathbb{R}$ of the form

$$T_d(\mathbf{x}_{1:d}; \mathbf{w}) = f(x_1, \dots, x_{d-1}, 0; \mathbf{w}) + \int_0^{x_d} g(\partial_d f(x_1, \dots, x_{d-1}, t; \mathbf{w}))\, dt, \qquad (1)$$

where $f(\mathbf{x}_{1:d}; \mathbf{w})$ is a general (non-monotone) function parameterized by coefficients $\mathbf{w}$ and $g : \mathbb{R} \to \mathbb{R}^+$ is a smooth positive function. Typically f takes the form of a multivariate polynomial expansion. The efficient implementation of Equation 1 is non-trivial as it requires the coordination of numerical quadrature, polynomial evaluations, and gradient computations with respect to both the input $\mathbf{x}$ and the parameters $\mathbf{w}$. But the associated optimization problem has many desirable features, such as the absence of spurious local minima (Baptista et al., 2022), which enable efficient and reliable map training.

`MParT` aims to provide a performance portable shared-memory implementation of parameterizations built on Equation 1. `MParT` uses Kokkos (Edwards et al., 2014) to leverage multithreading on either CPUs or GPUs with a common code base. `MParT` provides an efficient low-level library that can then be used to accelerate higher level packages like TransportMaps, ATM, and MUQ that cannot currently leverage GPU resources. Bindings to Python, Julia, and Matlab are also provided to enable a wide variety of users to leverage the fast C++ core from the language of their choice.

Performance and scalability

The following plots show the performance of MParT for the evaluation of a rectified degree-5 polynomial transport map on $\mathbb{R}^5$, using different languages and Kokkos backends. The monotone parameterization is constructed from Hermite polynomials and an adaptive Simpson quadrature rule. Random map coefficients and sample locations are used in this test. At each sample level, the map is evaluated at fifty randomly selected coefficients.

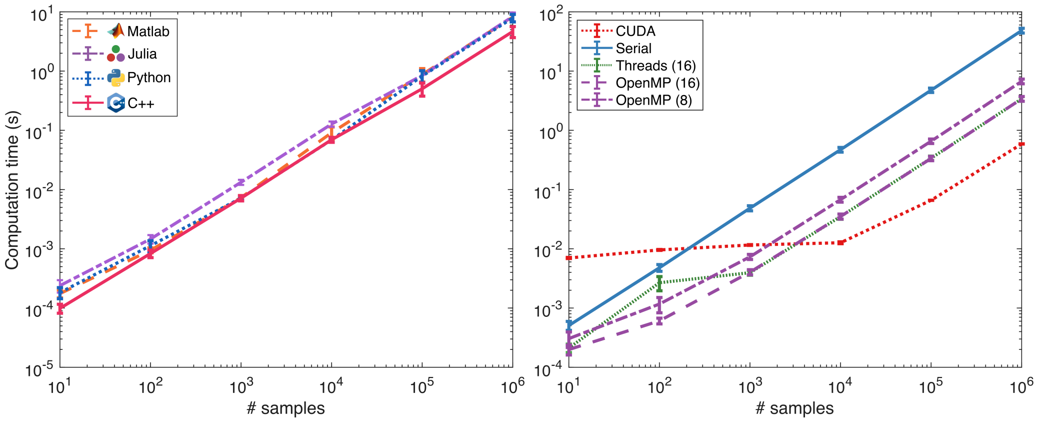

Figure 1: Time to evaluate triangular map from different languages and backends.

The results show similar performance across languages (each using OpenMP backend with 8 threads) and nearly identical performance between the Threads and OpenMP backends. For the evaluation of 10^6 samples, the OpenMP backend with 16 threads is approximately $14\times$ faster than the serial backend. The CUDA backend is approximately $82\times$ faster than the serial backend, or $6\times$ faster than the OpenMP backend. Tests were performed in a Kubernetes container using 8 cores of a Intel(R) Xeon(R) Gold 6248 CPU and a Tesla V100-SXM2 GPU with CUDA version 11.2.

Acknowledgements

We acknowledge support from the US Office of Naval Research under MURI Grant N00014-20-1-2595, the US Department of Energy under grant DE-SC0021226, the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Projektnummern 459970814; 459970841, and computing resources from Dartmouth College and the Massachusetts Institute of Technology.

References

Baptista, R., Rubio, P.-B., Zahm, O., & Marzouk, Y. (2021). Adaptive transport maps (ATM). In *GitHub repository*. GitHub. https://github.com/baptistar/ATM

Baptista, R., Zahm, O., & Marzouk, Y. (2022). On the representation and learning of monotone triangular transport maps. *arXiv Preprint arXiv:2009.10303v2*.

Bigoni, D. (2015). TransportMaps. In *Bitbucket repository*. Bitbucket. https://transportmaps.mit.edu

Bigoni, D., Spantini, A., & Marzouk, Y. (2016). Adaptive construction of measure transports for Bayesian inference. *NIPS Workshop on Approximate Inference*.

Cotter, C., Cotter, S., & Russell, P. (2019). Ensemble transport adaptive importance sampling. *SIAM/ASA Journal on Uncertainty Quantification*, *7*(2), 444–471. https://doi.org/10.1137/17m1114867

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., & Saurous, R. A. (2017). Tensorflow distributions. *arXiv Preprint arXiv:1711.10604*.

Edwards, H. C., Trott, C. R., & Sunderland, D. (2014). Kokkos: Enabling manycore performance portability through polymorphic memory access patterns. *Journal of Parallel and Distributed Computing*, *74*(12), 3202–3216. https://doi.org/10.1016/j.jpdc.2014.07.003

Marzouk, Y., Moselhy, T., Parno, M., & Spantini, A. (2016). Sampling via measure transport: An introduction. In R. Ghanem, D. Higdon, & H. Owhadi (Eds.), *Handbook of uncertainty quantification* (pp. 1–41). Springer International Publishing. https://doi.org/10.1007/978-3-319-11259-6_23-1

Moselhy, T., & Marzouk, Y. (2012). Bayesian inference with optimal maps. *Journal of Computational Physics*, *231*(23), 7815–7850.

Papamakarios, G., Nalisnick, E. T., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, *22*(57), 1–64.

Parno, M., Davis, A., & Seelinger, L. (2021). MUQ: The MIT uncertainty quantification library. *Journal of Open Source Software*, *6*(68), 3076. https://doi.org/10.21105/joss.03076

Parno, M., & Marzouk, Y. (2018). Transport map accelerated Markov chain Monte Carlo. *SIAM/ASA Journal on Uncertainty Quantification*, *6*(2), 645–682. https://doi.org/10.1137/17m1134640

Santambrogio, F. (2015). *Optimal transport for applied mathematicians* (1st ed., Vol. 87). Birkäuser Cham. https://doi.org/10.1007/978-3-319-20828-2

Spantini, A., Baptista, R., & Marzouk, Y. (2022). Coupling techniques for nonlinear ensemble filtering. *SIAM Review (to Appear)*.

Spantini, A., Bigoni, D., & Marzouk, Y. (2018). Inference via low-dimensional couplings. *The Journal of Machine Learning Research*, *19*(1), 2639–2709.