Using Tensor Trains to Solve High-Dimensional PDE

Ryan Zapp[1]

William Schill[1], Pierson Guthrey[1]

[1]Lawrence Livermore National Laboratory

August 3, 2023

LLNL DSTI Summer Poster Session

We aim to use tensor trains to accurately solve high-dimensional PDE. To do this, we develop ways to efficiently compress operators and functions in the PDE into the tensor train format. We then directly solve the PDE in this compressed format on a reasonably fine grid and sample the solution without unzipping any information.

Motivation:

- Fluid flow PDE have extreme resolution requirements
- Adequately resolving all scales is far too expensive
- One solution is to recast the PDE into a statistical transport equation:

$$\frac{\partial \boldsymbol{V}}{\partial t} + \boldsymbol{V}\cdot\nabla_x \boldsymbol{V} = -\frac{1}{\rho}\nabla_x P + \nu\nabla_x^2 \boldsymbol{V} \quad , \quad \nabla_x \cdot \boldsymbol{V} = 0$$

$$\boldsymbol{V} = [u(x,y,z,t)\,, v(x,y,z,t)\,, w(x,y,z,t)]$$

Recast PDE: trade extreme resolution requirements for high dimensionality

$$\frac{\partial f}{\partial t} + \mathcal{F}(f) = \nabla_{\boldsymbol{\omega}}\cdot(f\boldsymbol{D}) - 2\nu\nabla_{\boldsymbol{\omega}}\cdot(f\boldsymbol{b}) + \nu\frac{\partial}{\partial\omega^i}\left(\frac{\partial(f a_k^i)}{\partial x^k}\right)$$

$$\mathcal{F} = \nabla_{\boldsymbol{x}}\cdot\boldsymbol{B} + \boldsymbol{B}\cdot\nabla_{\boldsymbol{x}} - \nu\nabla_{\boldsymbol{x}}^2$$

$$f = f(x,y,z,\omega_1,\omega_2,\omega_3,t)$$

- Instead of solving the Navier-Stokes equations for a velocity field $\boldsymbol{V}$, we solve a transport equation for a probability density function f

Benefits of Tensor Trains:

- Tensor trains allow us to reduce the negative impact of the curse of dimensionality when solving high dimensional PDE

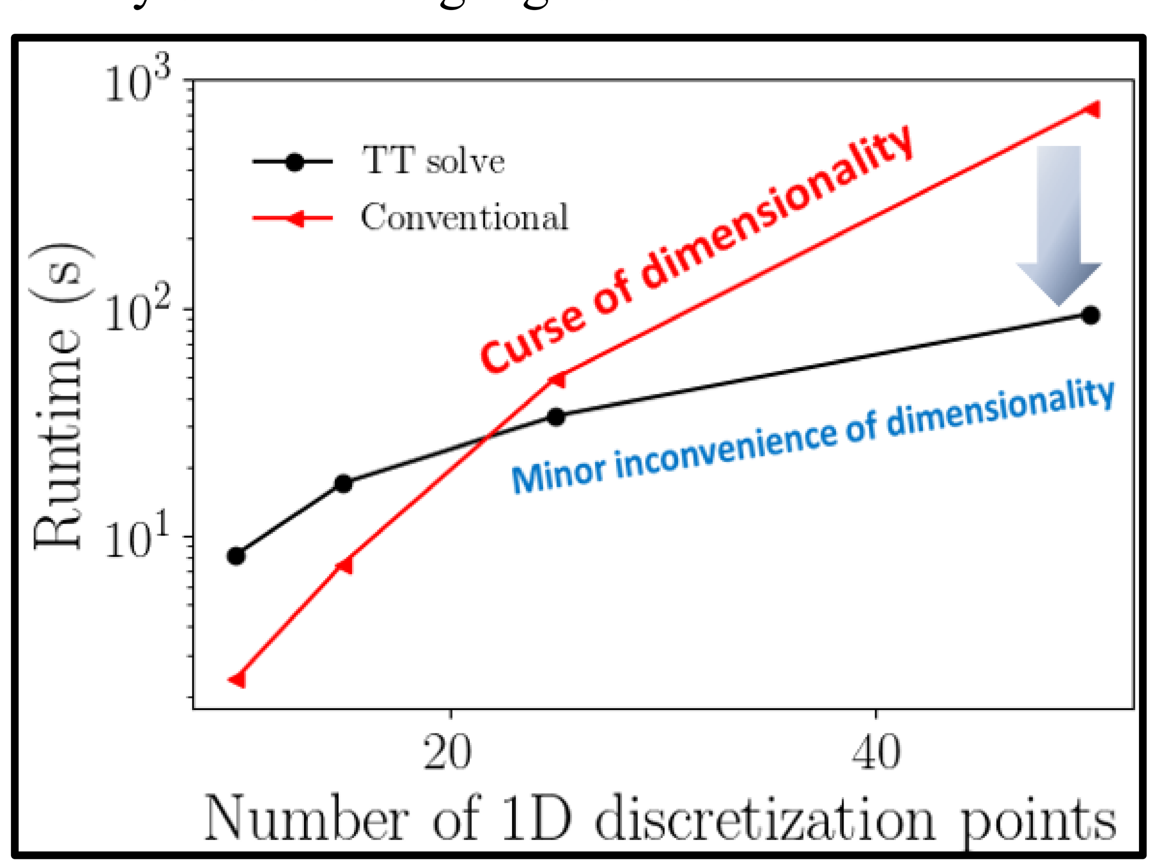

Figure 1: Runtime vs number of nodes in 1D for a conventional PDE solve vs a tensor train PDE solve

Definition of a Tensor Train:

- Consider the discrete NxN periodic Laplacian in 1D:

$$\nabla_x^2 = \frac{1}{\Delta x^2}\begin{bmatrix} -2 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & -2 \end{bmatrix}$$

- Standard representation of this operator requires $\mathcal{O}(N^2)$ storage
- We can compress our representation of this operator:

$$\nabla_x^2 = \frac{\partial^2}{\partial x^2}\otimes I = \frac{1}{\Delta x^2}\begin{bmatrix} -2 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & -2 \end{bmatrix} \otimes \overbrace{\begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}}^{\sqrt{N}} \left.\vphantom{\begin{bmatrix} 1 \\ \vdots \\ 0 \end{bmatrix}}\right\}\sqrt{N}$$

- With this representation, we only require $\mathcal{O}(2N)$ storage
- The generalization of this compression process to d dimensions is known as a tensor train

Efficient Tensor Train Assembly:

- To solve a PDE with tensor trains, we must convert the terms in the PDE to the tensor train format
- Efficient assembly of these terms in the tensor train format is essential for maximizing code performance
- Leveraging symmetries in functions and operators often allows us to efficiently assemble tensor trains:

$$\nabla_x^2 = \frac{\partial^2}{\partial x^2}\otimes I \otimes I + I \otimes \frac{\partial^2}{\partial y^2}\otimes I + I \otimes I \otimes \frac{\partial^2}{\partial z^2}$$

- Unlike the Laplacian, many operators and functions do not contain obvious symmetries

Current Work:

- Our immediate goal is to compute integrals of the following form:

$$F(\boldsymbol{x},\boldsymbol{\omega},t) = \int_{\mathbb{R}^3}\frac{1}{r(\boldsymbol{x},\boldsymbol{y})^p} f(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\omega},t)\,d\boldsymbol{y}$$

$$\frac{1}{r^p} = \frac{1}{\sqrt{(x_0-x)^2+(y_0-y)^2+(z_0-z)^2}}$$

- Integrals of this form appear in the terms $\boldsymbol{B}$ and $\boldsymbol{D}$ in the transport PDE we wish to solve
- By leveraging the properties of exponentials, we can represent this function as a tensor train:

$$\frac{1}{r} = \frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty}\otimes_{i=1}^{3} A(t)\,dt$$

$$A_{i,j} = e^{-t^2(x_j - x_i)^2}$$

- We can then perform a change of interval and approximate this integral using Gauss-Legendre quadrature

Next Steps:

- Test convergence rate of the numerical tensor train integration
- Use convergence rate data to develop optimal method for integrating the tensor train matrix
- Use this method to efficiently assemble $\boldsymbol{B}$ and $\boldsymbol{D}$ in the statistical transport PDE and solve for f

Works Cited:

1) Li J, Qian Z, Zhou M. 2022 On the transport equation for probability density functions of turbulent vorticity fields. *Proc. R. Soc. A* **478**: 20210534. https://doi.org/10.1098/rspa.2021.0534
2) Oseledets, I. V., 2011 Tensor-Train Decomposition. *DIAM journal on Scientific Computing* https://doi.org/10.1137/090752286

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.