Step Truncation Methods for Nonlinear Evolution Equations on Tensor Manifolds

Bram Rodgers

NASA ARC-TNA

Advanced Modeling & Simulation (AMS) Seminar Series NASA Ames Research Center, December 14th, 2023

1. Background and Motivation
2. Tensor Methods for Function Approximation
   - Two-Dimensional Functions
   - Higher Dimensional Functions
   - Data Parallelization
3. Tensor Methods for Time-Dependent PDEs
   - Convergence of Global Error
   - Explicit Step-Truncation Methods
   - Implicit Step-Truncation Methods
   - Absolute Stability
4. Numerical Examples
   - Fokker-Planck Equation
   - Nonlinear Schrödinger
   - Burgers-Hopf FDE

Introduction

Equations of the form

$$\frac{\partial f}{\partial t} = \mathcal{G}(f) \tag{1}$$

where $\mathcal{G}$ is a (time independent, maybe nonlinear) differential operator on functions $\Omega \mapsto \mathbb{R}$. We consider simple domains, $\Omega = [0, L_1] \times \cdots \times [0, L_d]$. Time is defined on $[0, T]$.

GOAL: Study (1) with preferably small computer memory footprint. We follow our recent approaches [1,2,3].

Roughly, $\boldsymbol{f}_{k+1} = \mathfrak{T}(\boldsymbol{f}_k + \Delta t \boldsymbol{G}(\boldsymbol{f}_k))$, $\mathfrak{T} =$ compression algorithm

---

[1] A. Rodgers and D. Venturi, "Stability analysis of hierarchical tensor methods for time-dependent PDEs", *J. Comput. Phys.*, **409** (2020) p. 109341

[2] A. Rodgers, A. Dektor, and D. Venturi, "Adaptive integration of nonlinear evolution equations on tensor manifolds", *J. Sci. Comput.*, **92(2)** (2022), p.39.

[3] A. Rodgers, and D. Venturi, "Implicit step-truncation integration of nonlinear PDEs on low-rank tensor manifolds", *J. Sci. Comput.*, **97** (2023)

High Dimensional PDEs: Motivation

Problems in this class are common in engineering and physics.

Fokker-Planck Equation (Systems with noise)

$$\frac{\partial p(\boldsymbol{x},t)}{\partial t} = -\nabla \cdot (\boldsymbol{g}(\boldsymbol{x})p(\boldsymbol{x},t)) + \sum_{i,j=1}^{d} \frac{\partial^2 D_{ij}(\boldsymbol{x})p(\boldsymbol{x},t)}{\partial x_i \partial x_j}, \quad p(\boldsymbol{x},0) = p_0(\boldsymbol{x}). \tag{2}$$

Where $\boldsymbol{D}(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{C}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x})^\top$ is the diffusion tensor for an Ito SDE

$$d\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{x})dt + \boldsymbol{C}\,d\boldsymbol{W} \tag{3}$$

with initial distribution $\boldsymbol{x}(0) \sim p_0(\boldsymbol{x}_0)$.

High Dimensional PDEs: Motivation

Many body Schrödinger Equation (Quantum Mechanics)

$$i\frac{\partial \psi}{\partial t} = \sum_{j=1}^{d} a_j \frac{\partial^2 \psi}{\partial x_j^2} = H\psi, \quad \psi(\boldsymbol{x}, 0) = \psi_0(\boldsymbol{x}) \tag{4}$$

For d particles interacting in a single spacial dimension.

Conserved quantities:
Particle number: $\int_\Omega \psi^* \psi \; d\boldsymbol{x}$
Momentum: $\int_\Omega (H\psi)^* \, H\psi \; d\boldsymbol{x}$

High Dimensional PDEs: Motivation

Functional Differential Equations, such as the Burgers-Hopf equation

$$\begin{cases} \frac{\partial \Phi([\theta],t)}{\partial t} = \int_0^{2\pi} \left[\frac{i}{2} \frac{\partial}{\partial x} \frac{\delta^2 \Phi([\theta],t)}{\delta \theta(x)^2} + \nu \frac{\partial^2}{\partial x^2} \frac{\delta \Phi([\theta],t)}{\delta \theta(x)} \right] \theta(x) dx \\ \Phi([\theta],0) = \Phi_0([\theta]). \end{cases} \tag{5}$$

Can be approximated as a PDE with a large number of space-like variables $a_1, a_2, \ldots a_d$, e.g. Fourier coefficients.

$$\frac{\partial \phi}{\partial t} = \sum_{p=1}^{d} a_p \left(i \sum_{j,k=1}^{d} A_{pjk} \frac{\partial^2 \phi}{\partial a_k \partial a_j} + \nu \sum_{k=1}^{d} B_{pk} \frac{\partial \phi}{\partial a_k} \right) \tag{6}$$

ϕ's domain is a finite dimensional subspace of Φ's domain.

Recent Approaches in high dimensional PDEs

Some recent approaches are

- Sparse Collocation methods [1]
- High dimensional polynomial interpolation [2]
- High dimensional model representations [3]
- Tensor Methods for High Dimensional PDEs [4]

---

[1] J. Foo and G. E. Karniadakis. "Multi-element probabilistic collocation method in high dimensions". *J. Comput. Phys.* **2**29 (2010), p. 1536.

[2] V. Barthelmann, E. Novak, and K. Ritter. "High dimensional polynomial interpolation on sparse grids". *Advances in Computational Mechanics* **1**2 (2000), p. 273.

[3] G. Li and H. Rabitz. "Regularized random-sampling high dimensional model representation (RS-HDMR)". *Journal of Mathematical Chemistry* **4**3.3 (2008), p. 1207

[4] C. Lubich et al. "Dynamical approximation by hierarchical Tucker and tensor-train tensors". *SIAM J. on Matrix Anal. and Appl.* **3**4.2 (2013), p. 470

Paradigm of Choice

Clearly, a diversity of applications results in a diversity of approaches to high dimensional PDEs.
Tensor methods have the following properties:

- Avoid exponential scaling in d. (Curse of $d$imensionality.)
- Convergence with small time step.
- Straightforward implementation.
- Wide availability of numerical packages.
- Somewhat problem independent.

1. Background and Motivation
2. Tensor Methods for Function Approximation
   - Two-Dimensional Functions
   - Higher Dimensional Functions
   - Data Parallelization
3. Tensor Methods for Time-Dependent PDEs
   - Convergence of Global Error
   - Explicit Step-Truncation Methods
   - Implicit Step-Truncation Methods
   - Absolute Stability
4. Numerical Examples
   - Fokker-Planck Equation
   - Nonlinear Schrödinger
   - Burgers-Hopf FDE

Multivariate Function Approximation

We assume $\boldsymbol{x} \in \Omega = [0, L_1] \times [0, L_2] \times \cdots \times [0, L_d]$.
A well known way to represent a scalar valued function f is with a series expansion.

$$f(x_1, x_2, \ldots, x_d) \approx \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} \cdots \sum_{i_d=1}^{N} \boldsymbol{f}[i_1, i_2, \ldots, i_d] \varphi_{i_1}^1(x_1) \varphi_{i_2}^2(x_2) \cdots \varphi_{i_d}^d(x_3) \tag{7}$$

$\boldsymbol{f}$ is a tensor of coefficients, the φ_i^j are basis functions.
If $\{\varphi_i^j\}$ is a cardinal basis, then $\boldsymbol{f}[i_1, i_2, \ldots, i_d]$ represent values of f at the collocation nodes.

Number of stored floating point numbers: N^d (Curse of $d$imensionality.)

To keep this formulation, we need some kind of compression of the data.

1. Background and Motivation
2. Tensor Methods for Function Approximation
   - Two-Dimensional Functions
   - Higher Dimensional Functions
   - Data Parallelization
3. Tensor Methods for Time-Dependent PDEs
   - Convergence of Global Error
   - Explicit Step-Truncation Methods
   - Implicit Step-Truncation Methods
   - Absolute Stability
4. Numerical Examples
   - Fokker-Planck Equation
   - Nonlinear Schrödinger
   - Burgers-Hopf FDE

Low Rank Matrix Factorization

We first describe $d = 2$ for clarity.

$$f(x_1, x_2) \approx \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} \boldsymbol{f}[i_1, i_2] \varphi_{i_1}^1(x_1) \varphi_{i_2}^2(x_2)$$

$\rightarrow$ Approximate $\boldsymbol{f}$ with a matrix $\widetilde{\boldsymbol{f}}$ of rank $r \leq rank(\boldsymbol{f})$.
$\rightarrow$ This can be formulated as an optimization problem:

$$\min_{\boldsymbol{Q}, \boldsymbol{\Sigma}, \boldsymbol{V}} ||\boldsymbol{f} - \widetilde{\boldsymbol{f}}||_2$$

subject to

$$\widetilde{\boldsymbol{f}} = \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{V}^\top, \quad \boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}_r, \quad \boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}_r, \quad \boldsymbol{\Sigma} \text{ invertible.}$$

The solution consists of dropping the last $rank(\boldsymbol{f}) - r$ singular values of $\boldsymbol{f}$.
A function achieving this min is denoted by $\widetilde{\boldsymbol{f}} = \mathcal{T}_r(\boldsymbol{f})$.

Low Rank Matrix Operations

Let $\boldsymbol{f} = \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, $\boldsymbol{g} = \boldsymbol{Z}\boldsymbol{\Xi}\boldsymbol{Y}^\top$.

Summation: $\boldsymbol{f} + \boldsymbol{g} = \begin{bmatrix}\boldsymbol{Q} & \boldsymbol{Z}\end{bmatrix} \begin{bmatrix}\boldsymbol{\Sigma} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Xi}\end{bmatrix} \begin{bmatrix}\boldsymbol{V}^\top \\ \boldsymbol{Y}^\top\end{bmatrix}$ (Rank increase.)

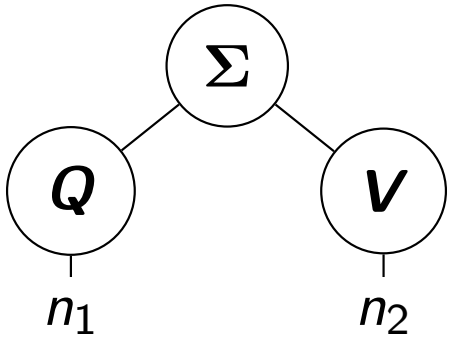
Scalar multiply: $\alpha\boldsymbol{f} = \boldsymbol{Q}(\alpha\boldsymbol{\Sigma})\boldsymbol{V}^\top$ (Keeps $\boldsymbol{Q}, \boldsymbol{V}$ orthogonal.)

Matrix multiplication:
$\boldsymbol{D}\boldsymbol{f} = (\boldsymbol{D}\boldsymbol{Q})\boldsymbol{\Sigma}\boldsymbol{V}^\top$
$\boldsymbol{f}\boldsymbol{D}^\top = \boldsymbol{Q}\boldsymbol{\Sigma}(\boldsymbol{D}\boldsymbol{V})^\top$ (Fewer column vectors to multiply with.)

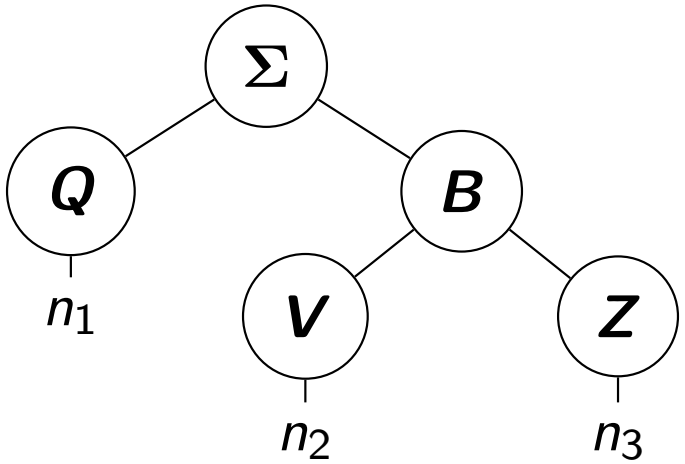
1. Background and Motivation
2. Tensor Methods for Function Approximation
   - Two-Dimensional Functions
   - Higher Dimensional Functions
   - Data Parallelization
3. Tensor Methods for Time-Dependent PDEs
   - Convergence of Global Error
   - Explicit Step-Truncation Methods
   - Implicit Step-Truncation Methods
   - Absolute Stability
4. Numerical Examples
   - Fokker-Planck Equation
   - Nonlinear Schrödinger
   - Burgers-Hopf FDE

Tensor Tree Graphs

Graphical representation: $\boldsymbol{f} = \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{V}^\top \in \mathbb{R}^{n_1 \times n_2}$



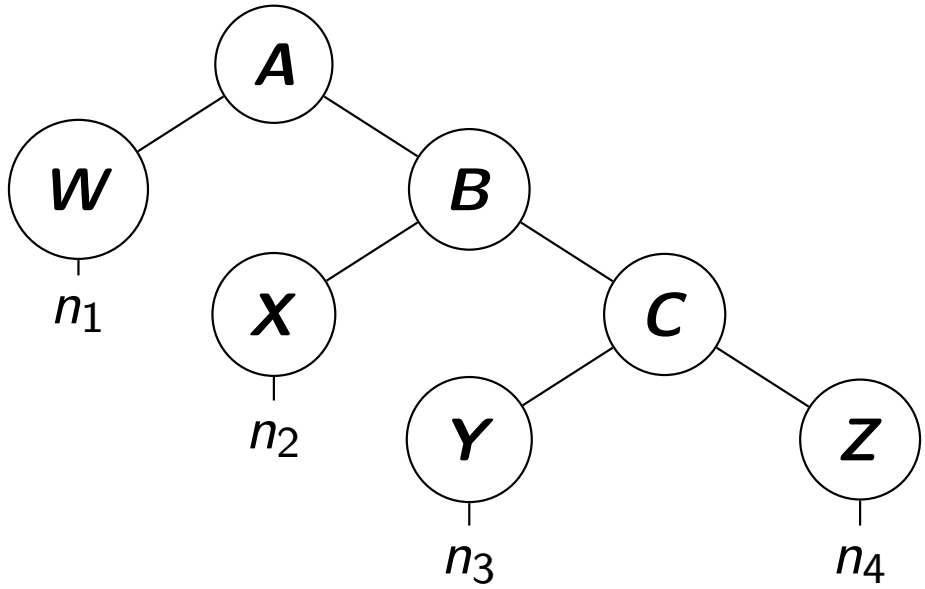
Every branching of a node represents a truncated SVD. For 3d arrays:



General process is called the Hierarchical Tucker (HT) decomposition.

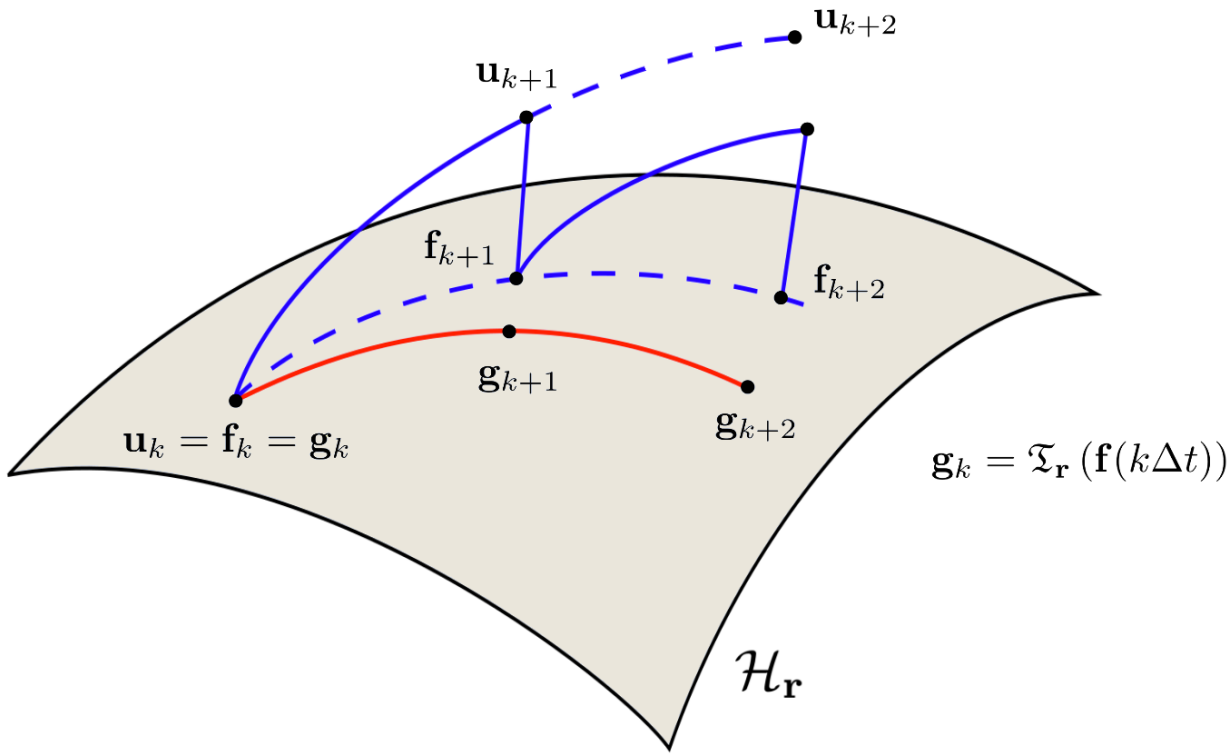
Tensor Tree Graphs

A tree graph for $\boldsymbol{f} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4}$ (Special Case called Tensor Train)



Tree rank: an array of integers containing rank of each matrix above.

Geometric Interpretation



Fixed rank constraint defines a manifold of tensors $\mathcal{H}_{\boldsymbol{r}}$

Main points of tensor methods for function approximation

Key features:

- Approximation accuracy based on decay of singular values.
- Full array storage is n^d, tensor methods are $O(dr^2(n+r))$ or $O(dnr^2)$
  $\rightarrow$ rank r is biggest concern on storage.
- Approximations are based on well studied linear algebra.
  $\rightarrow$ Matlab packages are available [1].
  $\rightarrow$ Notably, no harder to use than standard matrix arithmetic.
- Amenable to parallelization [2].

---

[1] D. Kressner and C. Tobler. "Algorithm 941: htucker - A Matlab toolbox for tensors in hierarchical Tucker format". *ACM Transactions on Mathematical Software* **4**0.3 (2014), pp. 1.

[2] H. A. Daas, G. Ballard, P. Benner. "Parallel Algorithms for Tensor Train Arithmetic." (2020) Arxiv: 2011.06532

1. Background and Motivation
2. Tensor Methods for Function Approximation
   - Two-Dimensional Functions
   - Higher Dimensional Functions
   - Data Parallelization
3. Tensor Methods for Time-Dependent PDEs
   - Convergence of Global Error
   - Explicit Step-Truncation Methods
   - Implicit Step-Truncation Methods
   - Absolute Stability
4. Numerical Examples
   - Fokker-Planck Equation
   - Nonlinear Schrödinger
   - Burgers-Hopf FDE

Parallel Memory Layout for Tensor Trains

Special case dimension tree implies TT may be written as

$$\boldsymbol{f}[i_1, i_2, \ldots, i_d] = \boldsymbol{C}_1[1, i_1, :]\boldsymbol{C}_2[:, i_2, :] \cdots \boldsymbol{C}_{d-1}[:, i_{d-1}, :]\boldsymbol{C}_d[:, i_d, 1].$$

Intuition: these are the coefficients of a truncated series expansion.

To parallelize, write each core $\boldsymbol{C}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$ as

$$\boldsymbol{C}_k[i, :, j] = \begin{bmatrix} \boldsymbol{C}_k^1[i, :, j] \\ \boldsymbol{C}_k^2[i, :, j] \\ \vdots \\ \boldsymbol{C}_k^P[i, :, j] \end{bmatrix}.$$

Each processor $p = 1, 2, \ldots, P$ gets one block $\boldsymbol{C}_k^p[i, :, j]$.
From here, efficient parallelization of linear algebra is achieved by use of the Tall-Skinny QR factorization algorithm.

1. Background and Motivation
2. Tensor Methods for Function Approximation
   - Two-Dimensional Functions
   - Higher Dimensional Functions
   - Data Parallelization
3. Tensor Methods for Time-Dependent PDEs
   - Convergence of Global Error
   - Explicit Step-Truncation Methods
   - Implicit Step-Truncation Methods
   - Absolute Stability
4. Numerical Examples
   - Fokker-Planck Equation
   - Nonlinear Schrödinger
   - Burgers-Hopf FDE

Discretization of PDEs

Consider the following PDE

$$\frac{\partial f}{\partial t} = \mathcal{G}(f) \quad \longrightarrow \begin{matrix} \textit{Discretizing} \\ \textit{in space} \end{matrix} \longrightarrow \quad \frac{d\boldsymbol{f}}{dt} = \boldsymbol{G}(\boldsymbol{f})$$

If $\boldsymbol{\Phi}_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}_k)$ is a local approximation of the flow map, we get a recurrence

$$\boldsymbol{f}_{k+1} = \boldsymbol{\Phi}_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}_k). \tag{8}$$

E.g., $\boldsymbol{\Phi}_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}_k) = \boldsymbol{f}_k + \Delta t \boldsymbol{G}(\boldsymbol{f}_k)$ is the Euler method.

Tensor rank of $\boldsymbol{f}_k$ can easily grow, raising computer memory usage.

$\rightarrow$Truncation $\mathcal{T}_{\boldsymbol{r}}(\boldsymbol{f}_{k+1})$ at each k required.

Step-Truncation Integrator

Generally, we call $\mathfrak{T}_{\boldsymbol{r}}$ a rank truncation if:
$\forall \varepsilon > 0$, there is a tensor rank $\boldsymbol{r}$ with

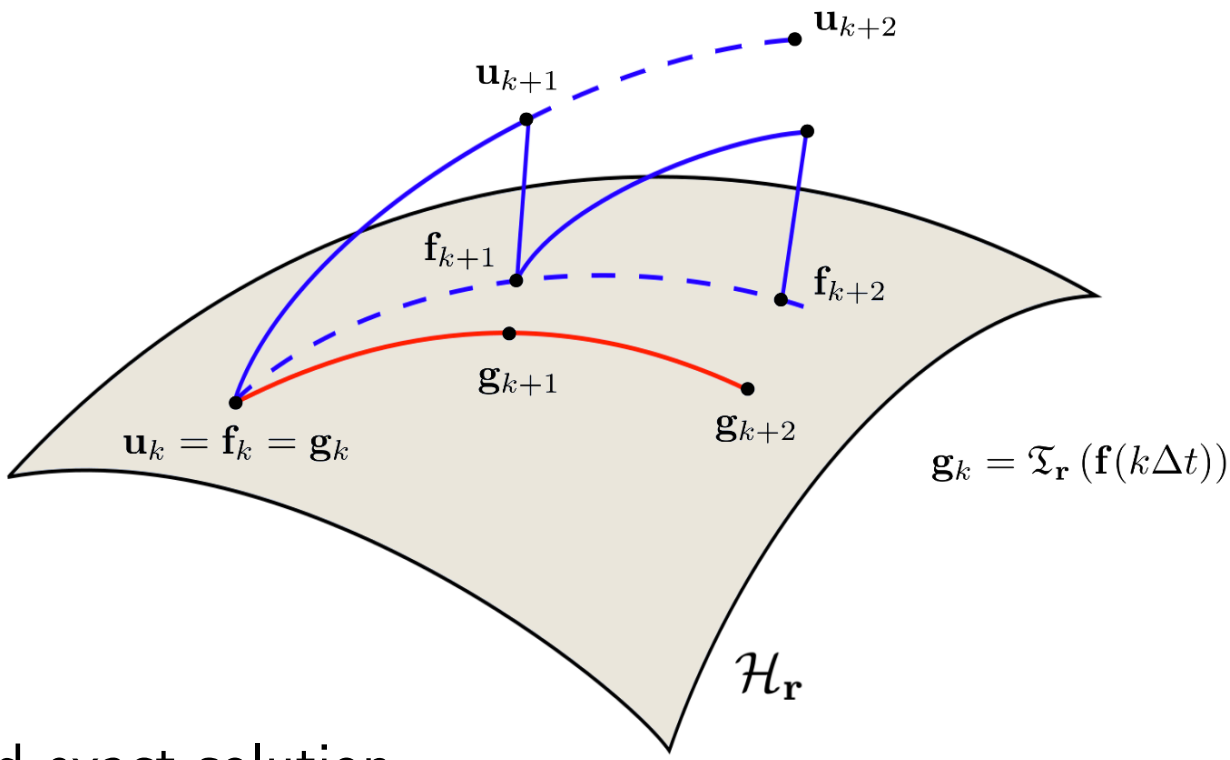
$$\|\boldsymbol{f} - \mathfrak{T}_{\boldsymbol{r}}(\boldsymbol{f})\| \leq \varepsilon \tag{9}$$

and $\text{rank}(\mathfrak{T}_{\boldsymbol{r}}(\boldsymbol{f})) \leq \boldsymbol{r}$.
Let $\boldsymbol{\Phi}_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}_k)$ be an approximate flow.

$$\boldsymbol{f}_{k+1} = \hat{\boldsymbol{\Phi}}_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}_k, \varepsilon) = \mathfrak{T}_{\boldsymbol{r}}\left(\boldsymbol{\Phi}_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}_k)\right) \tag{10}$$

is called a step-truncation integrator.

Geometric Interpretation



$\boldsymbol{g}_k$: Truncated exact solution.
$\boldsymbol{f}_k$: Step-Truncation method.
$\boldsymbol{u}_k$: Conventional integrator.
$\mathcal{H}_{\boldsymbol{r}}$: Tensor manifold of rank $\boldsymbol{r}$.

Step-Truncation Integrator

More generally, given an order p consistent scheme $\boldsymbol{f}_{k+1} = \boldsymbol{\Phi}_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}_k)$, a step truncation method

$$\boldsymbol{f}_{k+1} = \hat{\boldsymbol{\Phi}}_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}_k, \boldsymbol{e})$$

satisfies

$$\|\hat{\boldsymbol{\Phi}}_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}_k, \boldsymbol{e}) - \boldsymbol{\Phi}_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}_k)\| \leq R(\boldsymbol{e})$$

- $\boldsymbol{e} = (e_1, e_2, \ldots, e_s)$ is a list of low rank compression errors.
- $R(\boldsymbol{e})$ is the residual which scales with the total compression error.

E.g. explicit Euler
$\boldsymbol{f}_{k+1} = \mathcal{T}_{\boldsymbol{r}_2}\left(\boldsymbol{f}_k + \Delta t \mathcal{T}_{\boldsymbol{r}_1}\left(\boldsymbol{G}(\boldsymbol{f}_k)\right)\right)$ has $R(e_1, e_2) = e_1 \cdot \Delta t + e_2$

Equivalence of Rank Adaptivity and Consistency

We proved in [1] that the order of the rank-adaptive integrator is the same as the order of the scheme without truncation.

Lemma (Consistency)

Suppose $\|\boldsymbol{f}(\Delta t) - \boldsymbol{\Phi}_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}_0)\|_2 = O(\Delta t^{p+1})$. *Then*

$$\|\boldsymbol{f}(\Delta t) - \hat{\boldsymbol{\Phi}}_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}_0, \boldsymbol{e})\| = O(\Delta t^{p+1})$$

$$\Longleftrightarrow$$

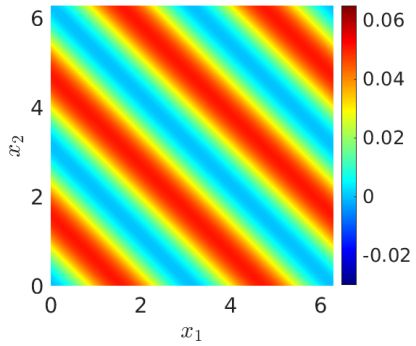
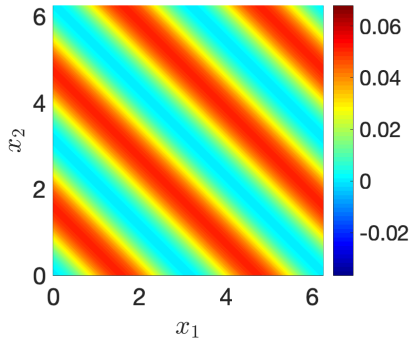
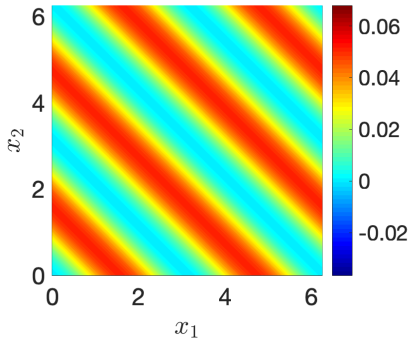
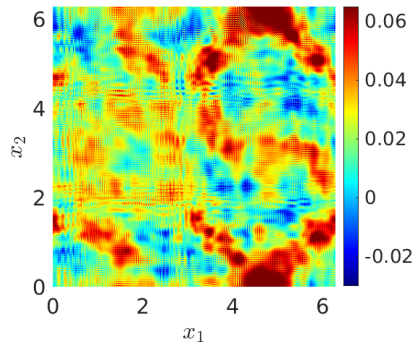
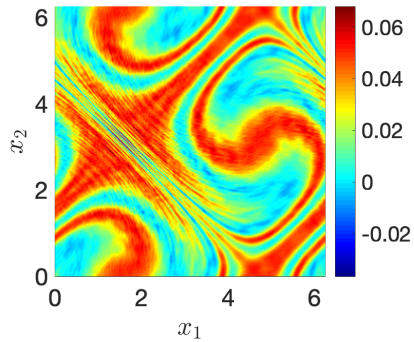
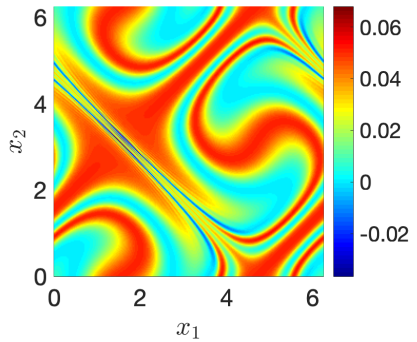
$$\|\boldsymbol{\Phi}_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}_0) - \hat{\boldsymbol{\Phi}}_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}_0, \boldsymbol{e})\| \leq M \Delta t^{p+1}$$

(M constant, rank is time dependent.)

[1] A. Rodgers, A. Dektor, and D. Venturi, "Adaptive integration of nonlinear evolution equations on tensor manifolds"

Error for Fixed Rank Step-Truncation

Fixed rank local error accumulates, leading to loss of long time accuracy.

	Rank 16 AB2	Rank 64 AB2	Reference AB2
$t = 0$			
$t = 5.0$			

1. Background and Motivation
2. Tensor Methods for Function Approximation
   - Two-Dimensional Functions
   - Higher Dimensional Functions
   - Data Parallelization
3. Tensor Methods for Time-Dependent PDEs
   - Convergence of Global Error
   - Explicit Step-Truncation Methods
   - Implicit Step-Truncation Methods
   - Absolute Stability
4. Numerical Examples
   - Fokker-Planck Equation
   - Nonlinear Schrödinger
   - Burgers-Hopf FDE

Zero Stability and Convergence

We say that $\Phi_{\Delta t}(\boldsymbol{G}, \boldsymbol{f})$ is zero stable with order p when

$$\|\Phi_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}) - \Phi_{\Delta t}(\boldsymbol{G}, \boldsymbol{g})\| \leq (1 + C\Delta t)\|\boldsymbol{f} - \boldsymbol{g}\| + E\Delta t^{p+1}. \tag{11}$$

When $\Phi_{\Delta t}(\boldsymbol{G}, \boldsymbol{f})$ and $\boldsymbol{G}$ are linear, this corresponds with the more traditional definition.

Theorem (Convergence for Step-Truncation Methods)

Suppose that $\Phi_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}, \boldsymbol{e})$ is any zero stable and consistent order p step-truncation method. Then the iteration

$$\boldsymbol{f}_{k+1} = \Phi_{\Delta t}(\boldsymbol{G}, \boldsymbol{f}_k, \boldsymbol{e})$$

is convergent. I.e. the global error at final time $T = N\Delta t$ is

$$\|\boldsymbol{f}(T) - \boldsymbol{f}_N\| = O(\Delta t^p)$$

1. Background and Motivation
2. Tensor Methods for Function Approximation
   - Two-Dimensional Functions
   - Higher Dimensional Functions
   - Data Parallelization
3. Tensor Methods for Time-Dependent PDEs
   - Convergence of Global Error
   - Explicit Step-Truncation Methods
   - Implicit Step-Truncation Methods
   - Absolute Stability
4. Numerical Examples
   - Fokker-Planck Equation
   - Nonlinear Schrödinger
   - Burgers-Hopf FDE

Explicit Methods: Rank-Adaptive Euler

The explicit Euler method:

$$\boldsymbol{f}_{k+1} = \boldsymbol{f}_k + \Delta t \boldsymbol{G}(\boldsymbol{f}_k) \tag{12}$$

A step-truncation Euler method:

$$\boldsymbol{f}_{k+1} = \mathfrak{T}_{\boldsymbol{r}}(\boldsymbol{f}_k + \Delta t \mathfrak{T}_{\boldsymbol{s}}(\boldsymbol{G}(\boldsymbol{f}_k))). \tag{13}$$

Convergence when:

$$\begin{aligned} &||\boldsymbol{G}(\boldsymbol{f}_k) - \mathfrak{T}_{\boldsymbol{s}}(\boldsymbol{G}(\boldsymbol{f}_k))|| \leq M_1 \Delta t, \\ &||\boldsymbol{f}_k + \Delta t \mathfrak{T}_{\boldsymbol{s}}(\boldsymbol{G}(\boldsymbol{f}_k)) - \mathfrak{T}_{\boldsymbol{r}}(\boldsymbol{f}_k + \Delta t \mathfrak{T}_{\boldsymbol{s}}(\boldsymbol{G}(\boldsymbol{f}_k)))|| \leq M_2 \Delta t^2, \end{aligned} \tag{14}$$

Explicit Methods: Explicit Midpoint

The explicit midpoint method:

$$\boldsymbol{f}_{k+1} = \boldsymbol{f}_k + \Delta t \boldsymbol{G}\left(\boldsymbol{f}_k + \frac{\Delta t}{2}\boldsymbol{G}(\boldsymbol{f}_k)\right) \tag{15}$$

A step-truncation midpoint method:

$$\boldsymbol{f}_{k+1} = \mathcal{T}_{\boldsymbol{\alpha}}\left(\boldsymbol{f}_k + \Delta t \mathcal{T}_{\boldsymbol{\beta}}\left(\boldsymbol{G}\left(\boldsymbol{f}_k + \frac{\Delta t}{2}\mathcal{T}_{\boldsymbol{\gamma}}(\boldsymbol{G}(\boldsymbol{f}_k))\right)\right)\right). \tag{16}$$

Convergence when:

$$\varepsilon_{\boldsymbol{\alpha}} \leq A\Delta t^3, \qquad \varepsilon_{\boldsymbol{\beta}} \leq B\Delta t^2, \qquad \varepsilon_{\boldsymbol{\gamma}} \leq G\Delta t, \tag{17}$$

where $\varepsilon_{\boldsymbol{\xi}} = \|\boldsymbol{g} - \mathcal{T}_{\boldsymbol{\xi}}(\boldsymbol{g})\|$, $\boldsymbol{\xi} = \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ are the rank truncation errors.

Explicit Methods: Rank-Adaptive AB2

Consider the two-step Adams-Bashforth method:

$$\boldsymbol{f}_{k+1} = \boldsymbol{f}_k + \Delta t \frac{3}{2} \boldsymbol{G}(\boldsymbol{f}_k) - \Delta t \frac{1}{2} \boldsymbol{G}(\boldsymbol{f}_{k-1}) \tag{18}$$

A step-truncation version of the AB2 method takes the form:

$$\boldsymbol{f}_{k+1} = \mathcal{T}_{\boldsymbol{\alpha}} \left(\boldsymbol{f}_k + \mathcal{T}_{\boldsymbol{\beta}} \left(\Delta t \frac{3}{2} \mathcal{T}_{\boldsymbol{\gamma}(1)} \left(\boldsymbol{G}(\boldsymbol{f}_k) \right) - \Delta t \frac{1}{2} \mathcal{T}_{\boldsymbol{\gamma}(2)} \left(\boldsymbol{G}(\boldsymbol{f}_{k-1}) \right) \right) \right) \tag{19}$$

Convergence when:

$$\varepsilon_{\boldsymbol{\alpha}} \leq A \Delta t^3, \qquad \varepsilon_{\boldsymbol{\beta}} \leq B \Delta t^2, \qquad \varepsilon_{\boldsymbol{\gamma}(j)} \leq G_j \Delta t^2, \; j = 1, 2. \tag{20}$$

again $\varepsilon_{\boldsymbol{\xi}} = \| \boldsymbol{g} - \mathcal{T}_{\boldsymbol{\xi}}(\boldsymbol{g}) \|$, $\boldsymbol{\xi} = \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}(j)$ are the rank truncation errors.

1. Background and Motivation
2. Tensor Methods for Function Approximation
   - Two-Dimensional Functions
   - Higher Dimensional Functions
   - Data Parallelization
3. Tensor Methods for Time-Dependent PDEs
   - Convergence of Global Error
   - Explicit Step-Truncation Methods
   - Implicit Step-Truncation Methods
   - Absolute Stability
4. Numerical Examples
   - Fokker-Planck Equation
   - Nonlinear Schrödinger
   - Burgers-Hopf FDE

Implicit Step-Truncation Methods

Implicit methods are defined by solving an algebraic equation

$$\boldsymbol{H}_k(\boldsymbol{f}_{k+1}) = \boldsymbol{0} \tag{21}$$

where $\boldsymbol{H}_k$ is defined in terms of previous time steps.

E.g., the implicit Euler method

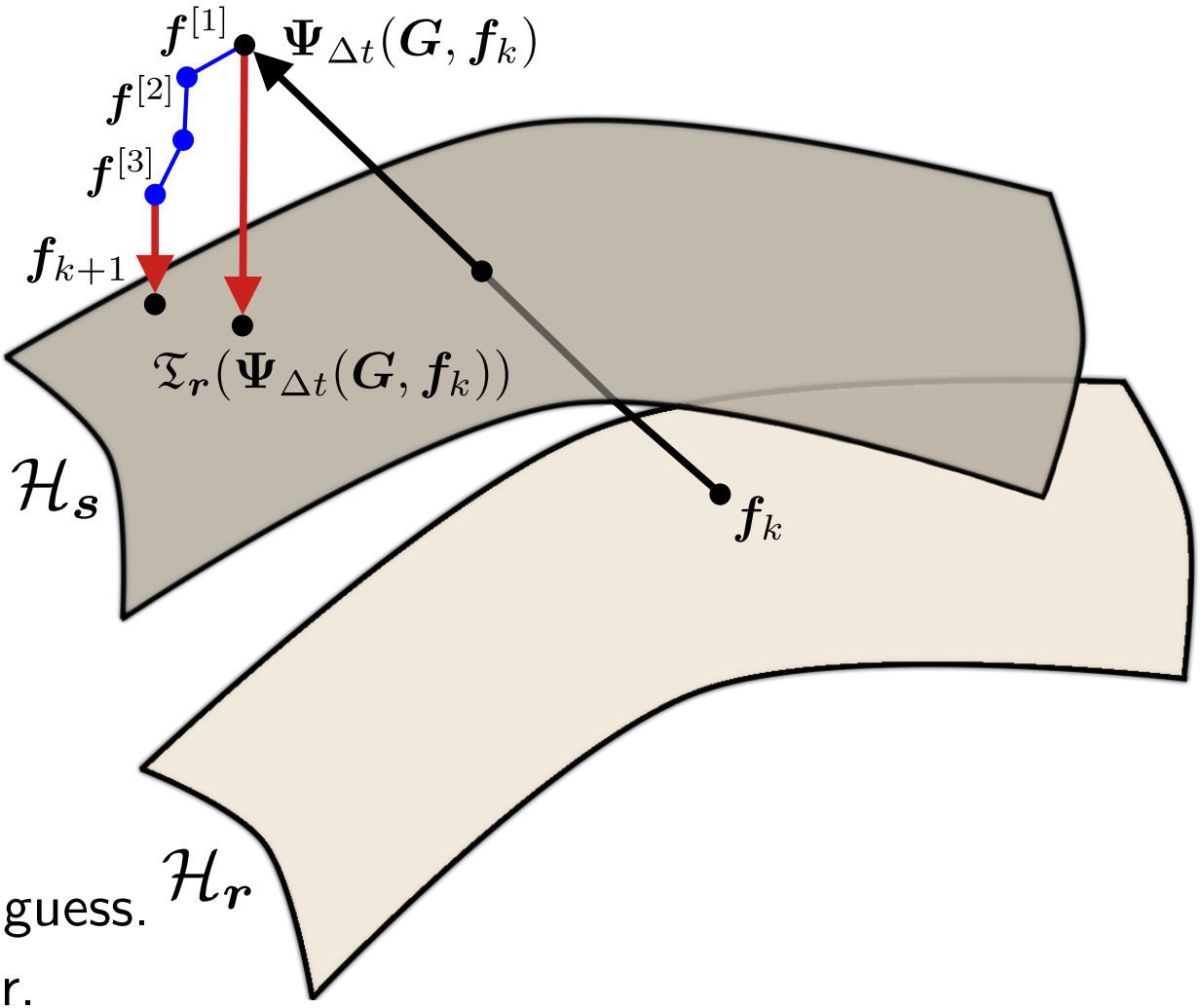
$$\boldsymbol{H}_k(\boldsymbol{f}_{k+1}) = \boldsymbol{f}_{k+1} - \boldsymbol{f}_k - \Delta t \boldsymbol{G}(\boldsymbol{f}_{k+1}) = \boldsymbol{0}$$

The implicit function theorem guarantees the existence of an approximate flow $\boldsymbol{f}_{k+1} = \boldsymbol{\Theta}_{\Delta t}(\boldsymbol{f}_k)$ solving (21).

Then an implicit step-truncation method is

$$\boldsymbol{f}_{k+1} = \mathfrak{T}_{\boldsymbol{r}}(\boldsymbol{\Theta}_{\Delta t}(\boldsymbol{f}_k))$$

Geometric Interpretation



$\Psi_{\Delta t}$: Explicit initial guess.
$f_k^{[j]}$: Iterative Solver.
f_{k+1} : Truncated implicit step.
$\mathcal{H}_s$: Tensor manifold of rank s.

Implicit Methods: Linear Solvers

Typically, the implicit increment map $\boldsymbol{\Theta}_{\Delta t}(\boldsymbol{f}_k)$ is computed by a linear solver or by Newton's iteration.

Full matrix representation impractical $\Rightarrow$ Iterative Linear Solver

Lemma (Accuracy of HT/TT-GMRES)

Let $\boldsymbol{J}\boldsymbol{f} = \boldsymbol{b}$ be a linear system where $\boldsymbol{f}$, $\boldsymbol{b}$ are tensors in HT or TT format, and $\boldsymbol{J}$ is a bounded linear operator on $\boldsymbol{f}$. Then for any $\varepsilon > 0$ there is an iterate j of HT/TT-GMRES method so that

$$\left\| \boldsymbol{J}\boldsymbol{f}^{[j]} - \boldsymbol{b} \right\| \leq m \|\boldsymbol{J}\| \|\boldsymbol{J}^{-1}\| \|\boldsymbol{b}\| \varepsilon, \tag{22}$$

where m is the number of Krylov vectors before restart. [1]

[1] S.V. Dolgov, "TT-GMRES: solution to a linear system in the structured tensor format.", *Russian Journal of Numerical Analysis and Mathematical Modelling*, **28(2)** (2013) p. 149

Implicit Methods: Inexact Newton

Theorem (Inexact Newton method)

Let $\boldsymbol{H} : \mathbb{R}^N \to \mathbb{R}^N$ *be smooth near a zero* $\boldsymbol{f}^*$. *Let the Jacobian of* $\boldsymbol{H}$, *i.e.,* $\boldsymbol{J_H}(\boldsymbol{f})$, *be invertible at* $\boldsymbol{f}^*$. *Given* $\boldsymbol{f}^{[0]} \in \mathbb{R}^N$, *consider the sequence*

$$\boldsymbol{f}^{[j+1]} = \boldsymbol{f}^{[j]} + \boldsymbol{s}^{[j]}, \qquad j = 0, 1, \ldots \tag{23}$$

where each $\boldsymbol{s}^{[j]}$ *solves the Newton iteration with the relative error formula*

$$\left\| \boldsymbol{J_H}\left(\boldsymbol{f}^{[j]}\right) \boldsymbol{s}^{[j]} + \boldsymbol{H}\left(\boldsymbol{f}^{[j]}\right) \right\| \leq \left\| \boldsymbol{H}\left(\boldsymbol{f}^{[j]}\right) \right\| \eta^{[j]}. \tag{24}$$

If $\eta^{[j]} < \tau < 1$ *for all* j, *then* $\boldsymbol{f}^{[j]} \to \boldsymbol{f}^*$ *linearly. If* $\eta^{[j]} \to 0$, *then* $\boldsymbol{f}^{[j]} \to \boldsymbol{f}^*$ *converges superlinearly.*

---

[1] Dembo, R.S., Eisenstat, S.C. and Steihaug, T., "Inexact Newton methods" *SIAM J. Num. Anal*, **19(2)** (1982), p.400

Implicit Methods: HT/TT Newton

Combining the inexact Newton Iteration with a truncation step and HT/TT-GMRES produces a nonlinear equation solver for the the HT/TT formats.

Specifically

$$\boldsymbol{f}^{[j+1]} = \mathcal{T}_{\boldsymbol{r}}\left(\boldsymbol{f}^{[j]} + \boldsymbol{s}^{[j]}\right)$$

where we solve the iteration with the stricter tolerance

$$\left\| \boldsymbol{J}_{\boldsymbol{H}}\left(\boldsymbol{f}^{[j]}\right)\boldsymbol{s}^{[j]} + \boldsymbol{H}\left(\boldsymbol{f}^{[j]}\right)\right\| \leq \frac{1}{2}\left\|\boldsymbol{H}\left(\boldsymbol{f}^{[j]}\right)\right\|\eta^{[j]},$$

where the relative error $\eta^{[j]}$ can be any value satisfying $0 \leq \eta^{[j]} < \tau < 1$. Then the Newton iteration converges linearly as long as we choose $\boldsymbol{r}$ to satisfy

$$\left\|\mathcal{T}_{\boldsymbol{r}}\left(\boldsymbol{f}^{[j]} + \boldsymbol{s}^{[j]}\right) - \boldsymbol{f}^{[j]} - \boldsymbol{s}^{[j]}\right\| \leq \frac{\left\|\boldsymbol{H}\left(\boldsymbol{f}^{[j]}\right)\right\|}{2\left\|\boldsymbol{J}_{\boldsymbol{H}}\left(\boldsymbol{f}^{[j]}\right)\right\|}\eta^{[j]}.$$

Implicit Methods: Implicit Euler

The root finding problem

$$\boldsymbol{H}_k(\boldsymbol{f}^*) = \boldsymbol{f}_{k+1} - \boldsymbol{f}_k - \Delta t \boldsymbol{G}(\boldsymbol{f}^*) = \boldsymbol{0}$$

Set the stopping tolerance as

$$\left\| \boldsymbol{H}_k \left(\boldsymbol{f}_{k+1}^{[j]} \right) \right\| = O(\Delta t^2).$$

Then set

$$\boldsymbol{f}_{k+1} = \mathcal{T}_{\boldsymbol{r}} \left(\boldsymbol{f}_{k+1}^{[j]} \right).$$

Convergence order 1 when rank truncation error is $O(\Delta t^2)$.

Implicit Methods: Implicit Midpoint

The root finding problem

$$\boldsymbol{H}_k\left(\boldsymbol{f}_{k+1}\right) = \boldsymbol{f}_{k+1} - \boldsymbol{f}_k - \Delta t \boldsymbol{G}\left(\frac{1}{2}(\boldsymbol{f}_k + \boldsymbol{f}_{k+1})\right) = \boldsymbol{0}$$

Set the stopping tolerance as

$$\left\| \boldsymbol{H}_k\left(\boldsymbol{f}_{k+1}^{[j]}\right) \right\| = O(\Delta t^3).$$

Then set

$$\boldsymbol{f}_{k+1} = \mathfrak{T}_{\boldsymbol{r}}\left(\boldsymbol{f}_{k+1}^{[j]}\right).$$

Convergence order 2 when rank truncation error is $O(\Delta t^3)$.

1. Background and Motivation
2. Tensor Methods for Function Approximation
   - Two-Dimensional Functions
   - Higher Dimensional Functions
   - Data Parallelization
3. Tensor Methods for Time-Dependent PDEs
   - Convergence of Global Error
   - Explicit Step-Truncation Methods
   - Implicit Step-Truncation Methods
   - Absolute Stability
4. Numerical Examples
   - Fokker-Planck Equation
   - Nonlinear Schrödinger
   - Burgers-Hopf FDE

Absolute Stability for Step-Truncation

In analysis of absolute stability, we are concerned if a linear update

$$\boldsymbol{f}_{k+1} = \boldsymbol{L}_{\Delta t}\boldsymbol{f}_k \tag{25}$$

produces a stable iteration solving a linear continuous time equation $d\boldsymbol{f}/dt = \boldsymbol{G}(\boldsymbol{f})$.

This is the case for explicit step-truncation methods:

Theorem (Explicit ST Stability)

Suppose

$$\boldsymbol{f}_{k+1} = \boldsymbol{L}_{\Delta t}\boldsymbol{f}_k \tag{26}$$

is lax stable. (I.e. $\|\boldsymbol{L}_{\Delta t}(\boldsymbol{f})\|_2 \leq (1 + K\Delta t)\|\boldsymbol{f}\|_2$.) Then

$$\boldsymbol{f}_{k+1} = \mathfrak{T}_{\boldsymbol{r}}(\boldsymbol{L}_{\Delta t}\boldsymbol{f}_k) \tag{27}$$

is stable with same stability coefficient.

Absolute Stability for Step-Truncation

Implicit ST methods require more precise analysis.

Theorem (Implicit ST Stability)

Let

$$\boldsymbol{A}\boldsymbol{f}_{k+1} = \boldsymbol{W}\boldsymbol{f}_k.$$

define an implicit linear update with contractive $\boldsymbol{A}^{-1}\boldsymbol{W}$*. If the HT/TT-GMRES is tolerance is selected so that*

$$\left\| \boldsymbol{A}\hat{\boldsymbol{f}}_k - \boldsymbol{W}\hat{\boldsymbol{f}}_{k-1} \right\| \leq m\|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|\eta,$$

then $\left\|\hat{\boldsymbol{f}}_k\right\| = O(\eta)$ *as* $k \to \infty$.

Unconditional stability is preserved on the order of accuracy of the approximate matrix inversion.

1. Background and Motivation
2. Tensor Methods for Function Approximation
   - Two-Dimensional Functions
   - Higher Dimensional Functions
   - Data Parallelization
3. Tensor Methods for Time-Dependent PDEs
   - Convergence of Global Error
   - Explicit Step-Truncation Methods
   - Implicit Step-Truncation Methods
   - Absolute Stability
4. Numerical Examples
   - Fokker-Planck Equation
   - Nonlinear Schrödinger
   - Burgers-Hopf FDE

1. Background and Motivation
2. Tensor Methods for Function Approximation
   - Two-Dimensional Functions
   - Higher Dimensional Functions
   - Data Parallelization
3. Tensor Methods for Time-Dependent PDEs
   - Convergence of Global Error
   - Explicit Step-Truncation Methods
   - Implicit Step-Truncation Methods
   - Absolute Stability
4. Numerical Examples
   - Fokker-Planck Equation
   - Nonlinear Schrödinger
   - Burgers-Hopf FDE

Numerical Example: Rank-Adaptive Fokker-Planck

Consider the Fokker-Planck Equation

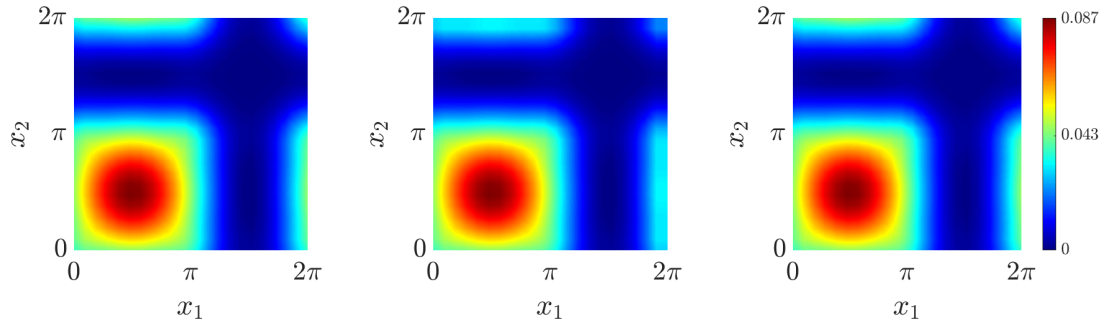
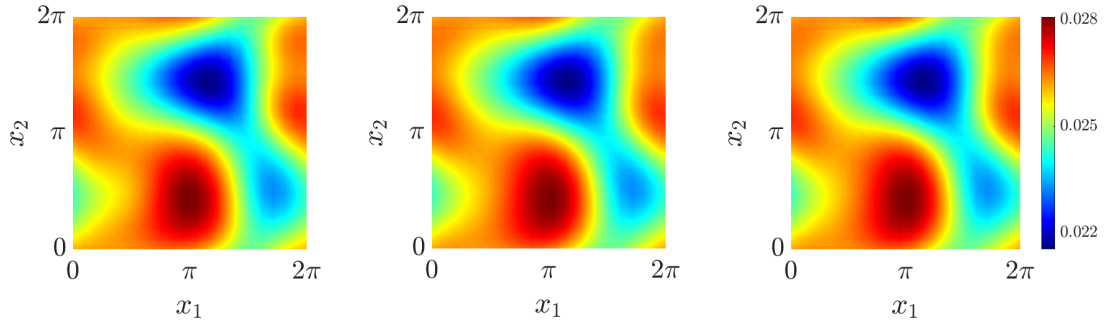
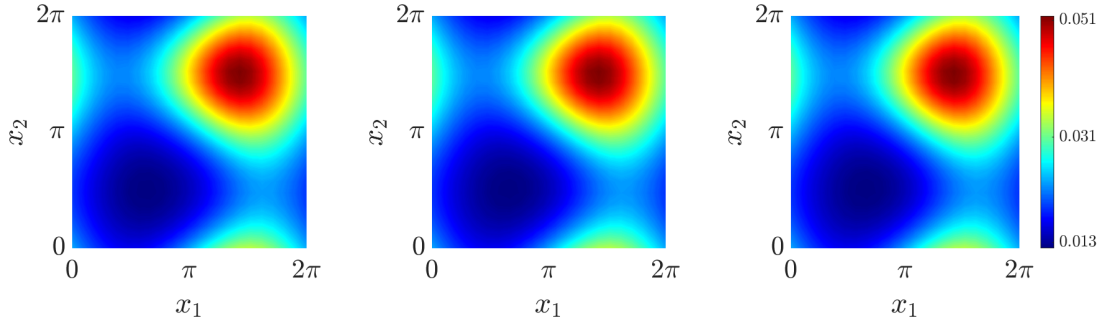
$$\frac{\partial f(\boldsymbol{x},t)}{\partial t} = -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}\left(\mu_i(\boldsymbol{x}) f(\boldsymbol{x},t)\right) + \frac{\sigma^2}{2}\sum_{i=1}^{d} \frac{\partial^2 f(\boldsymbol{x},t)}{\partial x_i^2}, \qquad f(\boldsymbol{x},0) = f_0(\boldsymbol{x}), \tag{28}$$

discretized on a d-dimensional flat torus.
In our numerical demonstrations, we set

$$\mu_i(\boldsymbol{x}) = (\gamma(x_{i+1}) - \gamma(x_{i-2}))\xi(x_{i-1}) - \phi(x_i), \qquad i = 1, \ldots, d \tag{29}$$

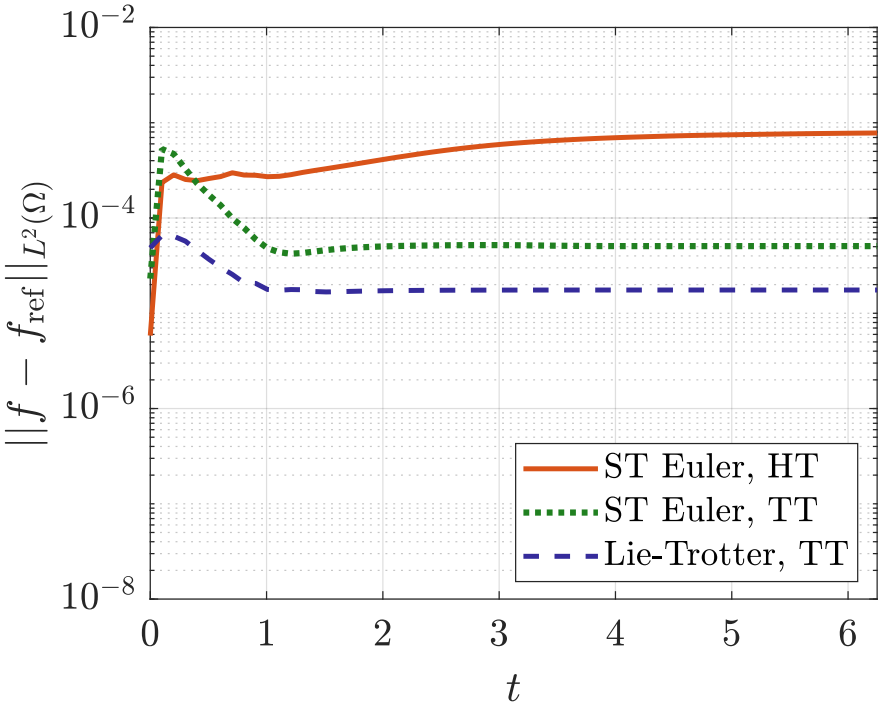
We run the experiment with various drift functions,
$\sigma = 2$ for explicit nonstiff, $\sigma = 5$ for implicit stiff.

4d Fokker-Planck: (Marginal PDF)

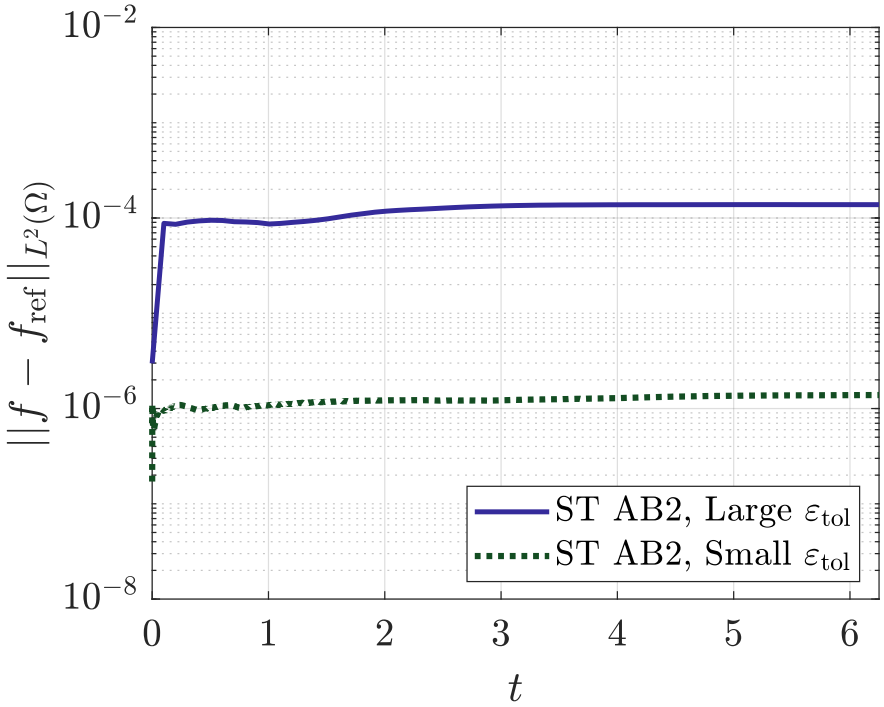
	Adaptive Euler	Adaptive AB2	Reference Rk4
$t = 0$			
$t = 0.4$			
Steady State			

4d Fokker-Planck: (Explicit Error vs Time)

Order 1

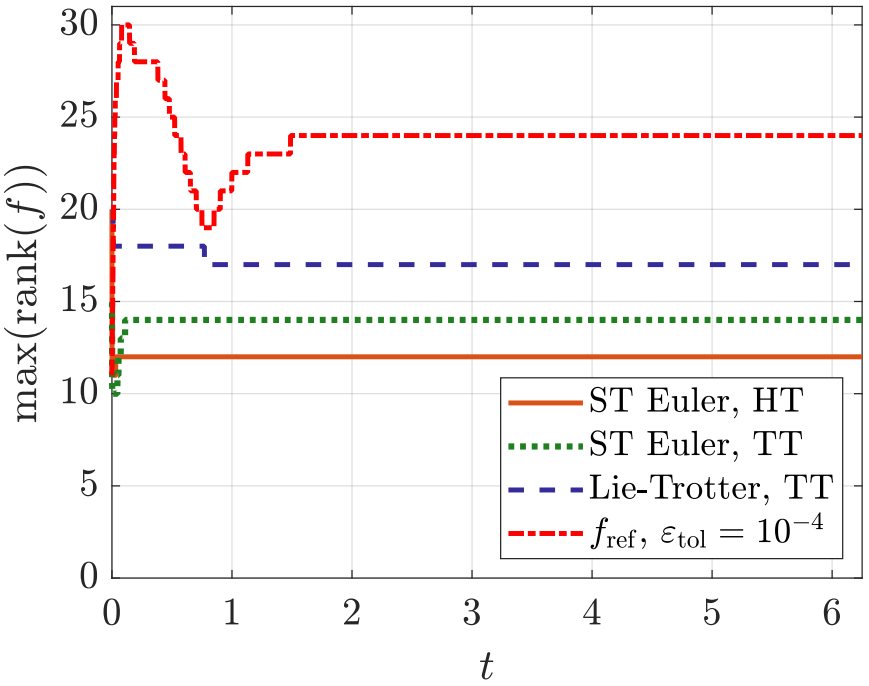


Order 2

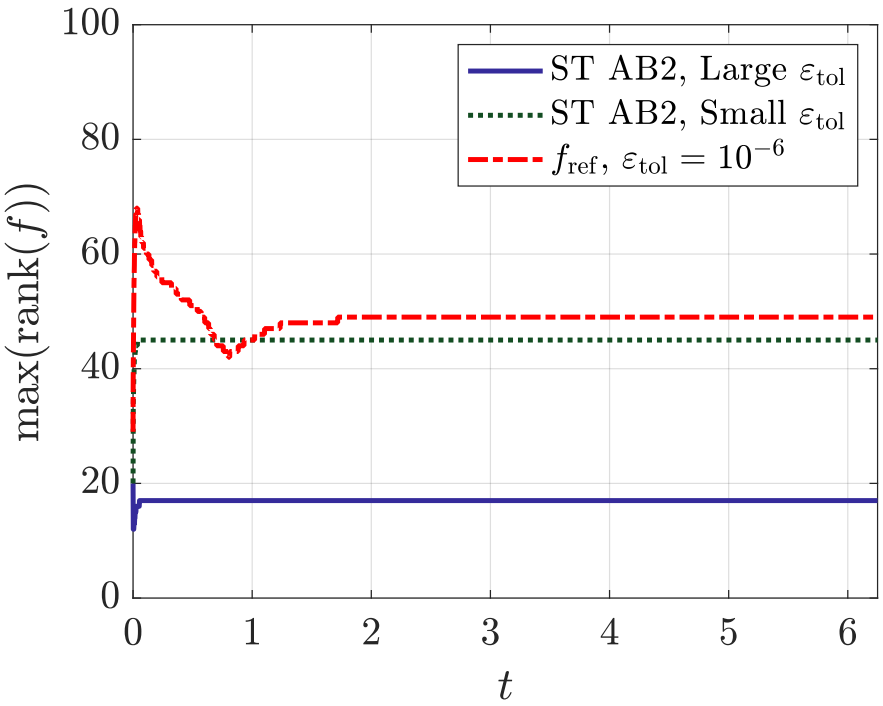


4d Fokker-Planck: (Explicit Rank vs Time)

Order 1

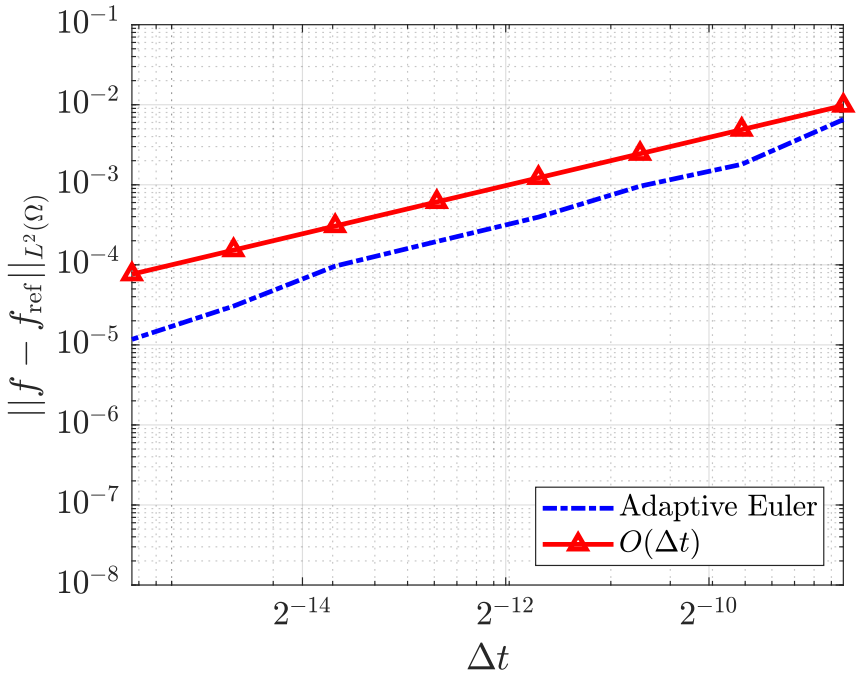


Order 2

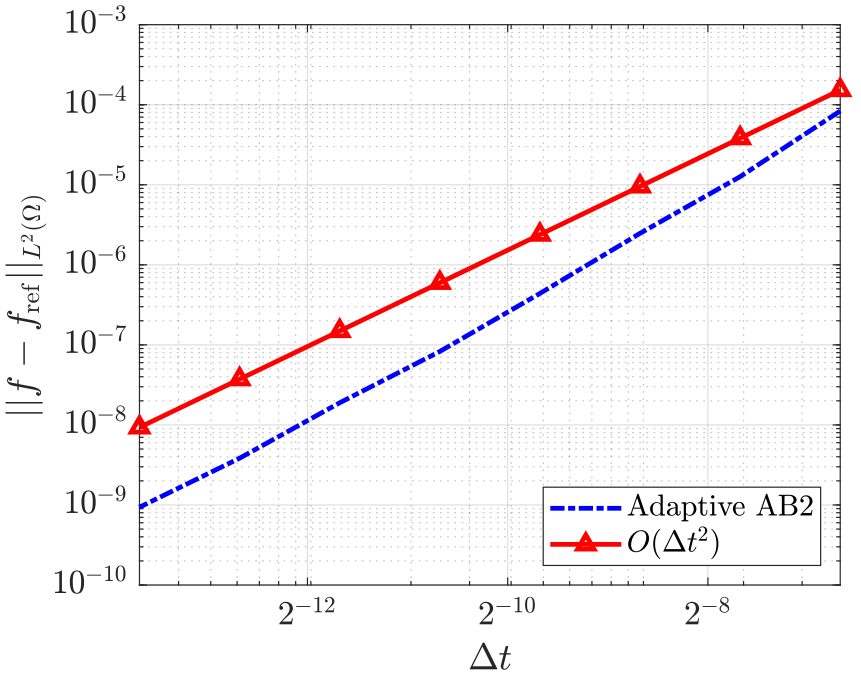


4d Fokker-Planck: (Convergence Order)

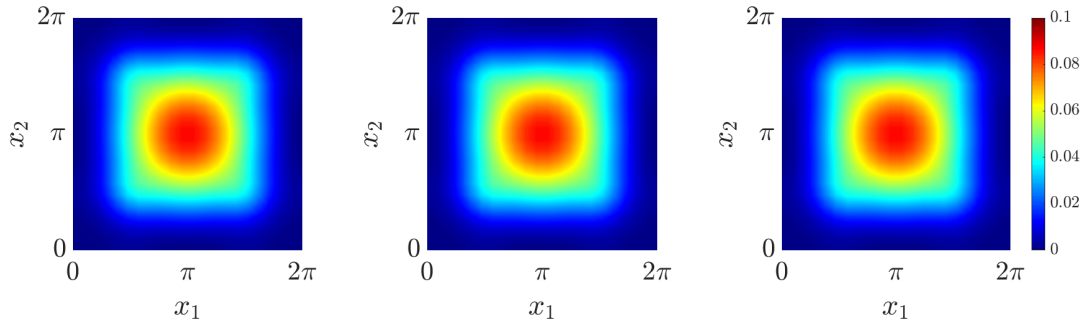
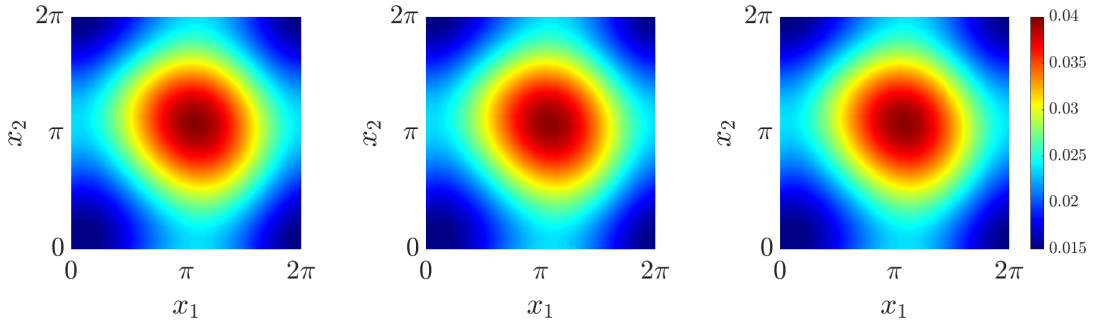
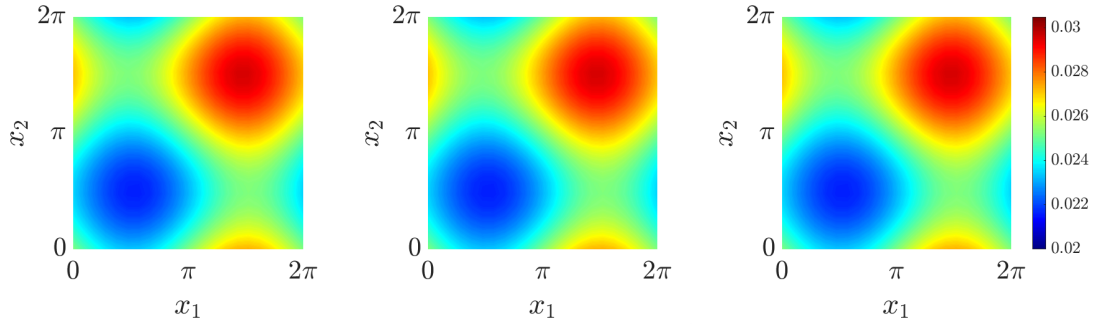
Adaptive Euler



Adaptive AB2

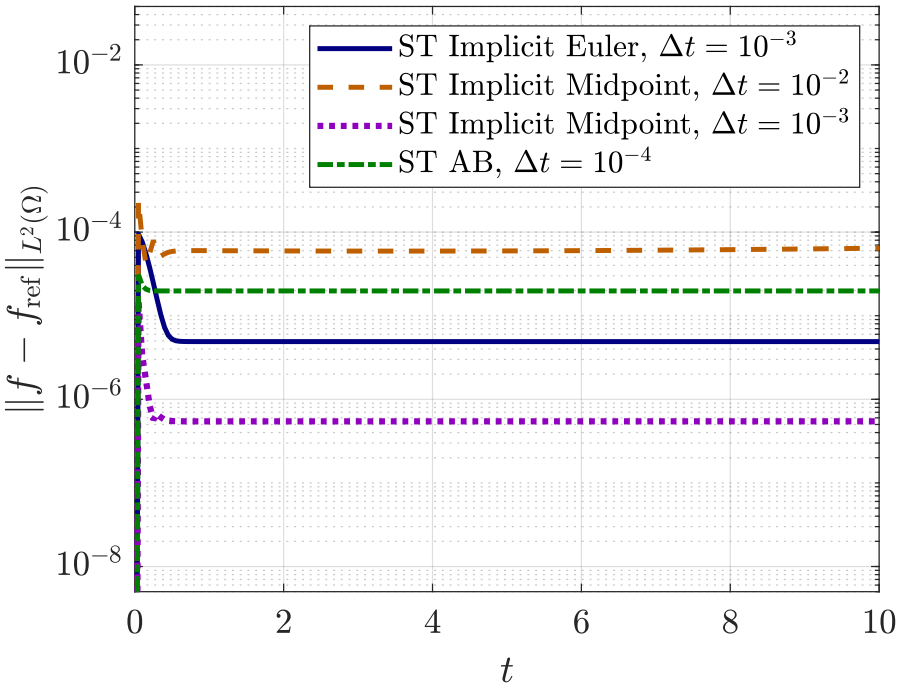


4d Fokker-Planck: (Marginal PDF)

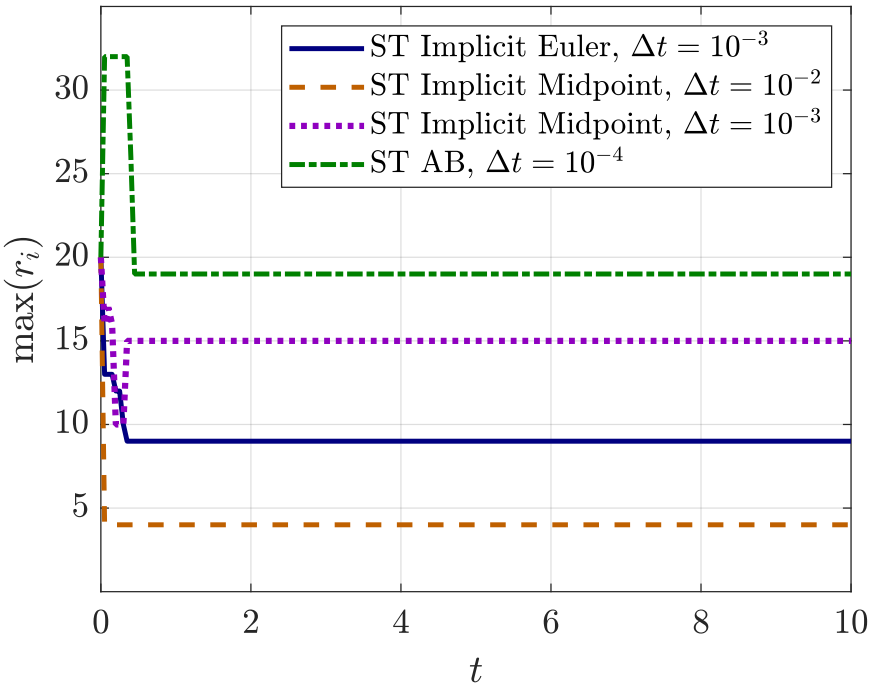
	Implicit Euler	Implicit Midpoint	Reference Rk4
$t = 0$			
$t = 0.1$			
Steady State			

4d Fokker-Planck: (Implicit Transients)

Transient Error

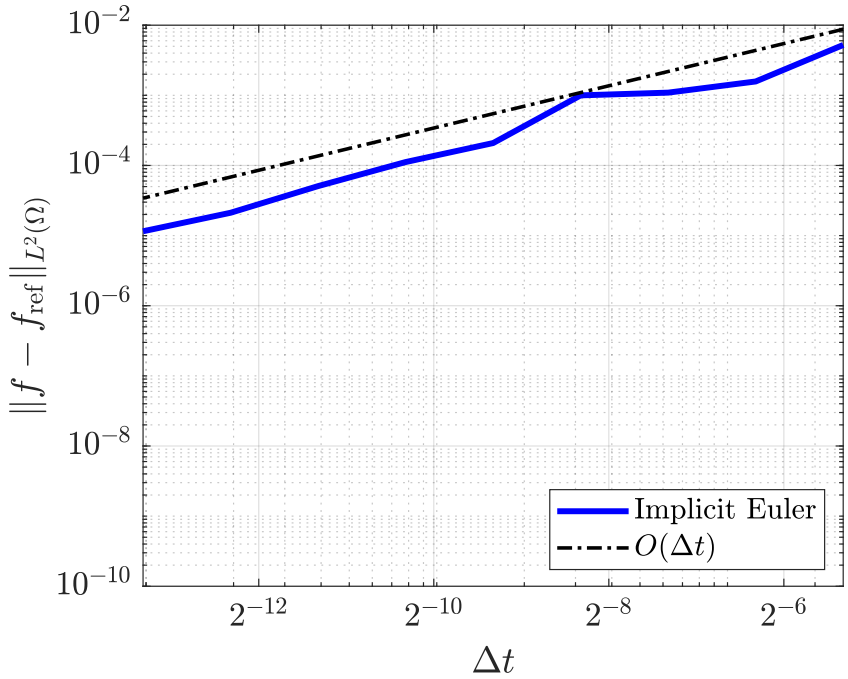


Transient Rank

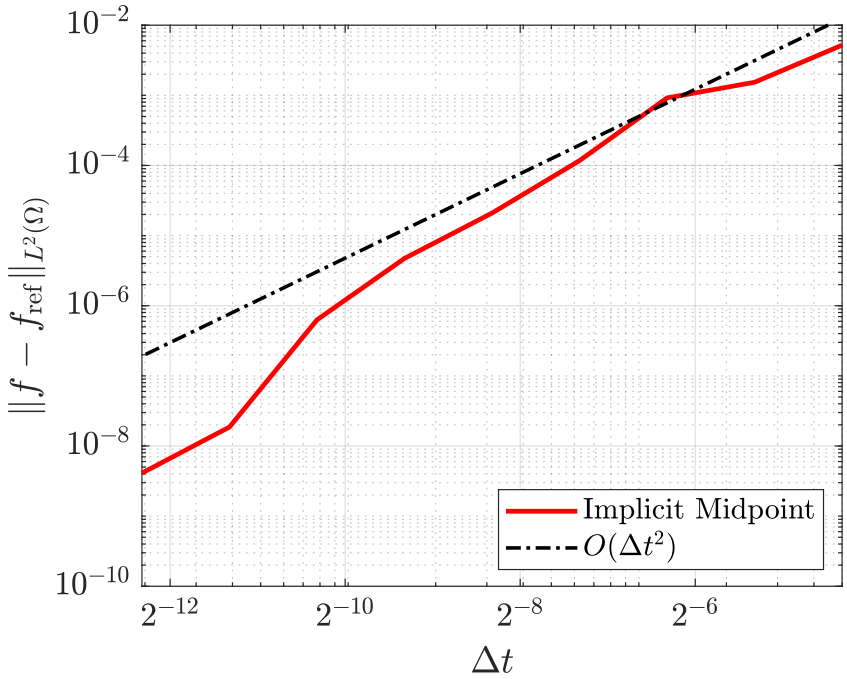


4d Fokker-Planck: (Convergence Order)

Implicit Euler



Implicit Midpoint

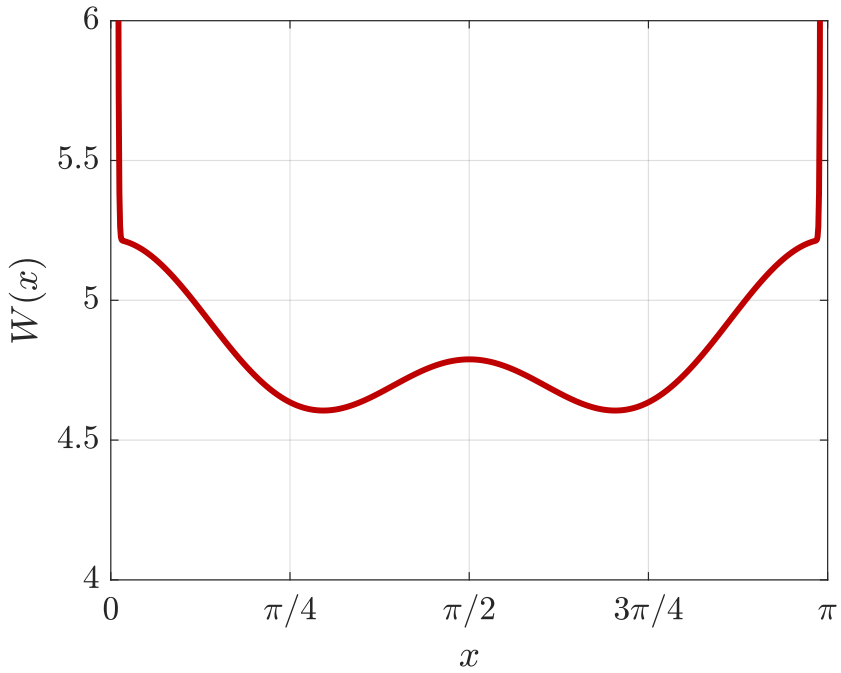


1. Background and Motivation
2. Tensor Methods for Function Approximation
   - Two-Dimensional Functions
   - Higher Dimensional Functions
   - Data Parallelization
3. Tensor Methods for Time-Dependent PDEs
   - Convergence of Global Error
   - Explicit Step-Truncation Methods
   - Implicit Step-Truncation Methods
   - Absolute Stability
4. Numerical Examples
   - Fokker-Planck Equation
   - Nonlinear Schrödinger
   - Burgers-Hopf FDE

6d Nonlinear Schrödinger: (Marginal PDF)

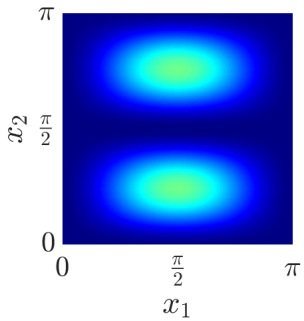
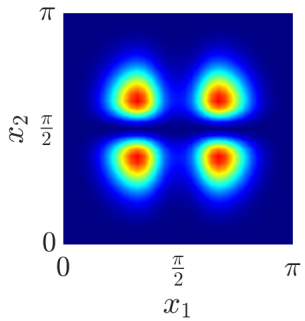
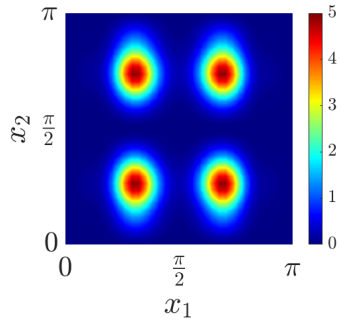
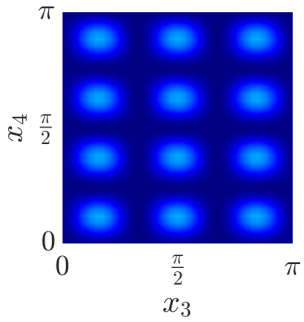
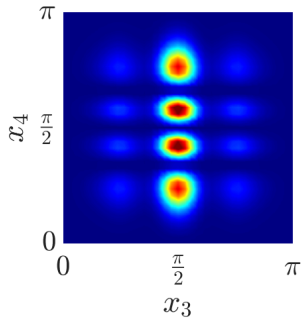
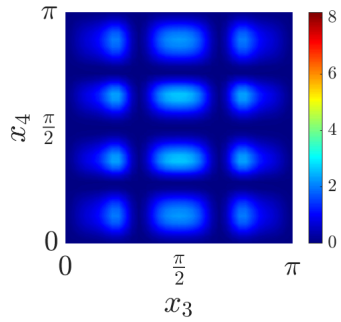
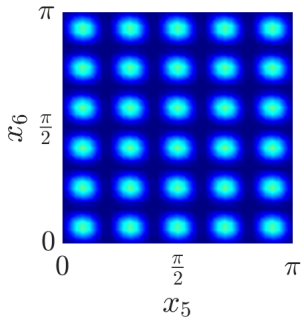
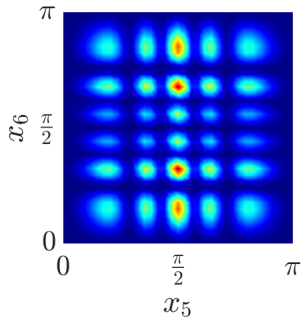
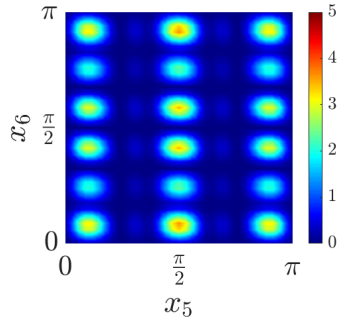
$$\frac{\partial \phi(\boldsymbol{x}, t)}{\partial t} = \frac{i}{2}\Delta\phi(\boldsymbol{x}, t) - iV(\boldsymbol{x})\phi(\boldsymbol{x}, t) - i\varepsilon|\phi(\boldsymbol{x}, t)|^2\phi(\boldsymbol{x}, t). \tag{30}$$

Nonlinear interaction term has $\varepsilon = 10^{-4}$. Potential $V(\boldsymbol{x})$ defined by trapping all particles in a double well



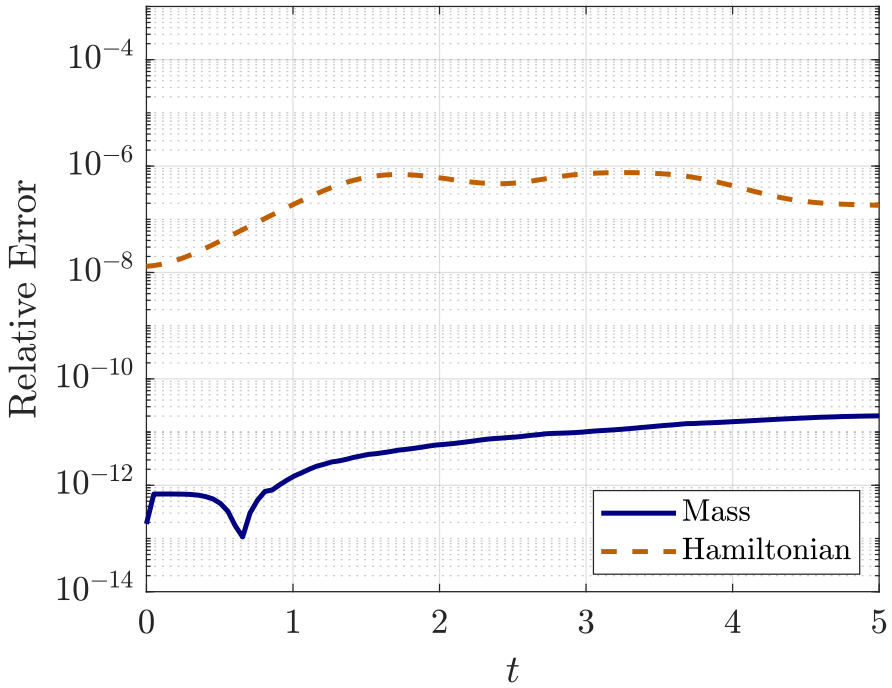
Solutions found using implicit midpoint.

6d Nonlinear Schrödinger

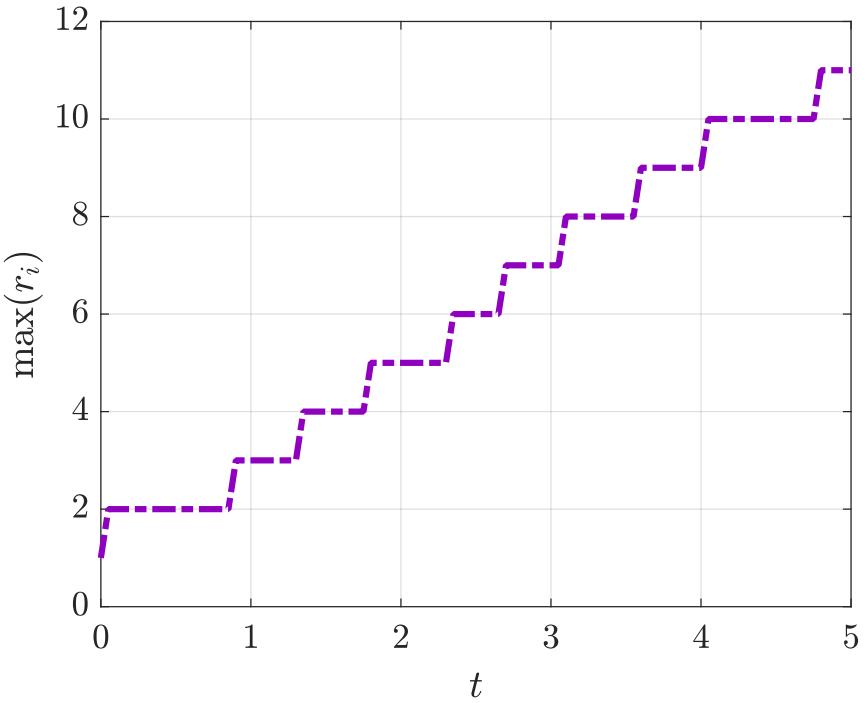
	$t = 0$	$t = 2.5$	$t = 5$
$p(x_1, x_2, t)$			
$p(x_3, x_4, t)$			
$p(x_5, x_6, t)$			

6d Nonlinear Schrödinger: Physical Quantities

Transient Error



Transient Rank



1. Background and Motivation
2. Tensor Methods for Function Approximation
   - Two-Dimensional Functions
   - Higher Dimensional Functions
   - Data Parallelization
3. Tensor Methods for Time-Dependent PDEs
   - Convergence of Global Error
   - Explicit Step-Truncation Methods
   - Implicit Step-Truncation Methods
   - Absolute Stability
4. Numerical Examples
   - Fokker-Planck Equation
   - Nonlinear Schrödinger
   - Burgers-Hopf FDE

Burgers-Hopf FDE Approximation

Nonlinear PDE (stochastic IVP)

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} = \nu\dfrac{\partial^2 u}{\partial x^2} + \alpha\sin(\pi u) \\ u(0,x) = u_0(x) \qquad \text{(random process)} \end{cases}$$

Burgers-Hopf FDE (linear IVP)

$$\begin{cases} \dfrac{\partial \Phi\left([\theta],t\right)}{\partial t} = \displaystyle\int_0^{2\pi}\left[\frac{i}{2}\frac{\partial}{\partial x}\frac{\delta^2\Phi([\theta],t)}{\delta\theta(x)^2} + \nu\frac{\partial^2}{\partial x^2}\frac{\delta\Phi([\theta],t)}{\delta\theta(x)} + \sum_{k=1}^{\infty}\frac{(-\pi)^k}{(2k+1)!i^{2k}}\frac{\delta^{2k+1}\Phi([\theta],t)}{\delta\theta(x)^{2k+1}}\right]\theta(x)dx \\ \Phi([\theta],0) = \Phi_0([\theta]) \end{cases}$$

CDF equation (FDE approximation)

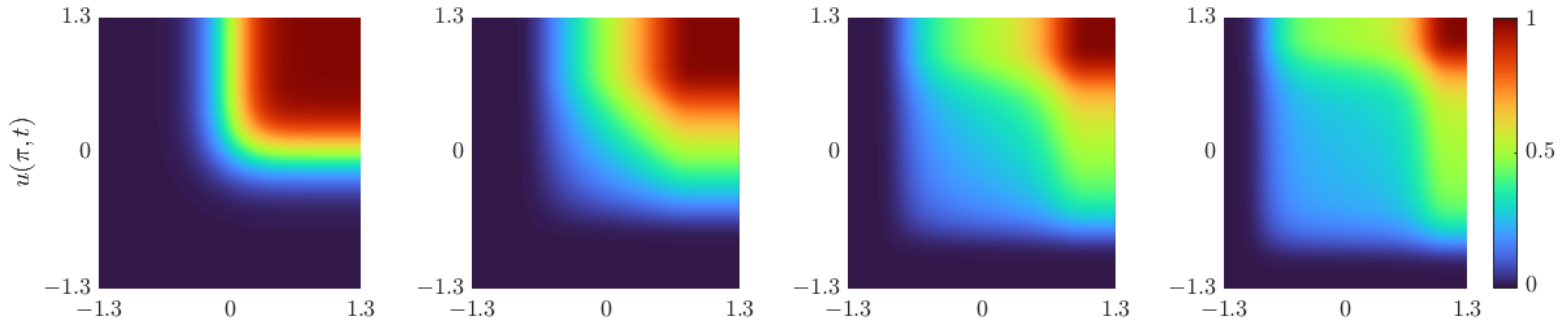
$$\begin{cases} \dfrac{\partial F}{\partial t} = \displaystyle\sum_{i,j=1}^{N}\left(D_{ij}^{(1)}u_i + \nu D_{ij}^{(2)}\right)u_j\frac{\partial F}{\partial u_i} - \sum_{\substack{i,j=1\\ i\neq j}}^{N}\left(D_{ij}^{(1)}u_i + \nu D_{ij}^{(2)}\right)\int_{-\infty}^{u_j}\frac{\partial F}{\partial u_i}du_j - \alpha\sum_{j=1}^{N}\sin(\pi u_j)\frac{\partial F}{\partial u_j} \\ F(0,\boldsymbol{u}) = F_0(\boldsymbol{u}) \end{cases}$$

We solve in $N = 20$ variables with the explicit midpoint method.

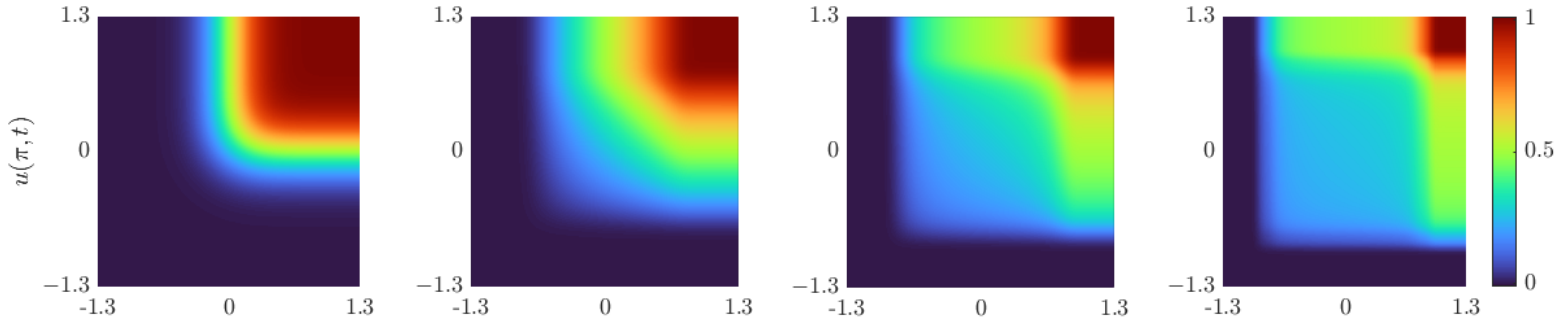
Burgers-Hopf FDE Approximation

$t = 0.0$ $t = 0.04$ $t = 0.08$ $t = 0.12$

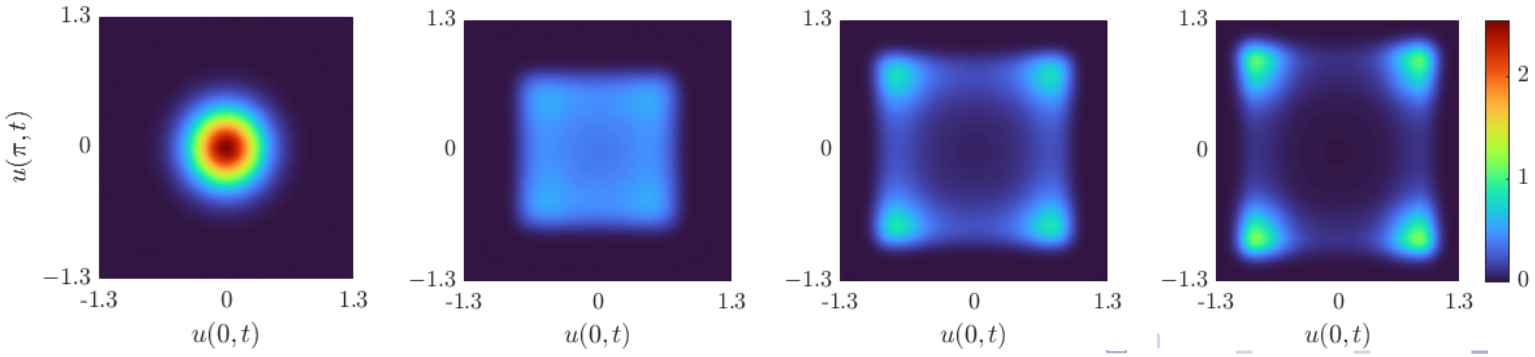
MC CDF



ST CDF

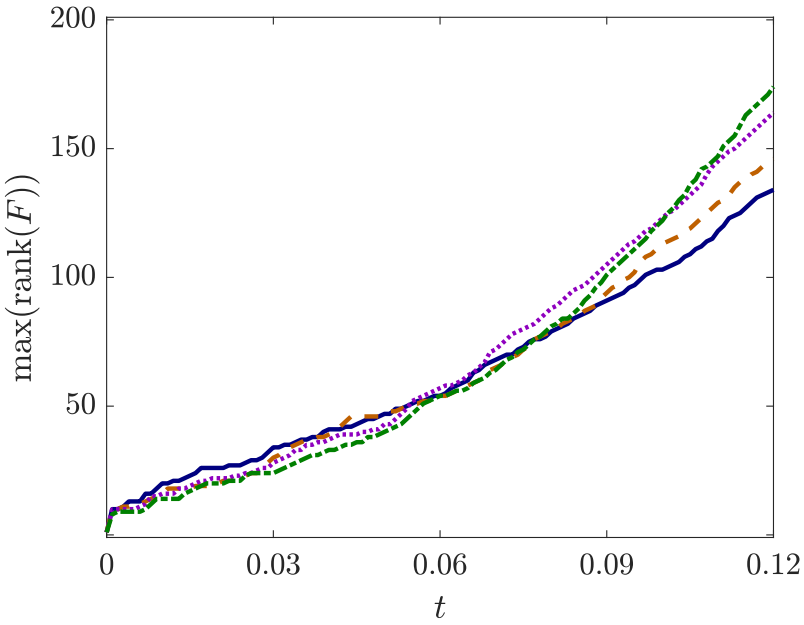


MC PDF

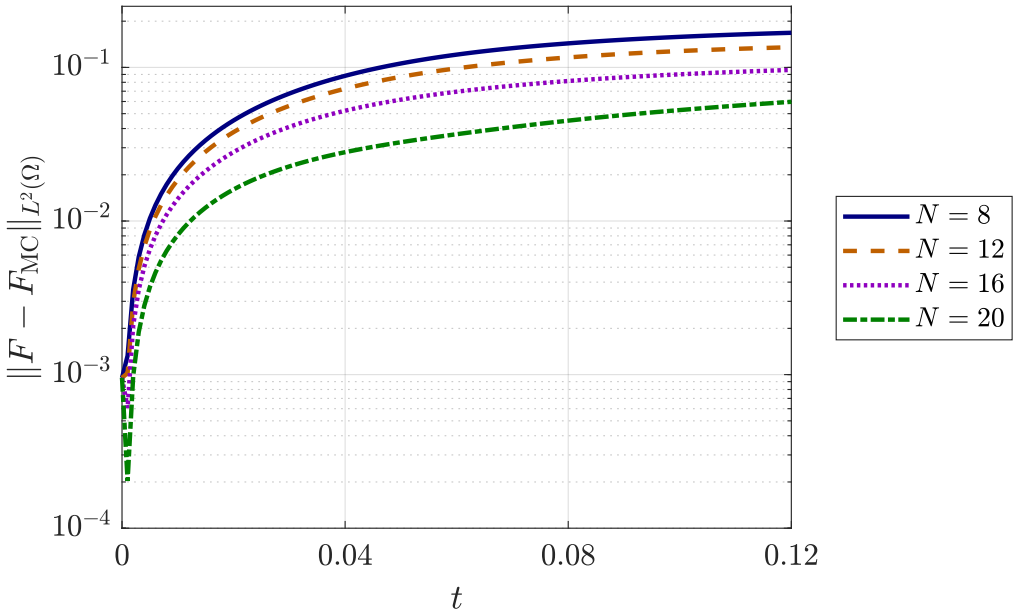


Burgers-Hopf FDE Approximation

Transient Rank



Transient Error



Summary and Conclusion

- We developed new algorithms called Step-Truncation methods for temporal integration of nonlinear evolution equations on tensor manifolds.
- We constructed a rigorous mathematical framework for explicit and implicit problems, proving that these methods are convergent.
- We also developed a rank-adaptive variant for Newton's method for the solutions to the implicit methods' recurrence relations.
- These methods are straightforward to implement as they rely only on arithmetic operations between tensors, which can be performed by efficient and scalable parallel algorithms.
- Numerical applications were presented and discussed for linear diffusion problems with spatially dependent coefficients, a nonlinear Schrödinger equation, and a discretized functional differential equation.

References and Acknowledgements

This research was supported by the U.S. Army Research Office (ARO) grant W911NF-18-1-0309, U.S. Air Force Office of Scientific Research (AFOSR) grant FA9550-20-1-0174, and partial funding for Bram's PhD tenure was provided by the Transformational Tools and Technologies (T^3) Project through the NASA Pathways internship program.

---

[1] A. Rodgers and D. Venturi, "Stability analysis of hierarchical tensor methods for time-dependent PDEs", *J. Comput. Phys.*, **409** (2020) p. 109341

[2] A. Dektor, A. Rodgers, D. Venturi, "Rank-adaptive tensor methods for high-dimensional nonlinear PDEs", *J. Sci. Comput.*, **88(2)** (2021), p.36.

[3] A. Rodgers, A. Dektor, and D. Venturi, "Adaptive integration of nonlinear evolution equations on tensor manifolds", *J. Sci. Comput.*, **92(2)** (2022), p.39.

[4] A. Rodgers, and D. Venturi, "Implicit step-truncation integration of nonlinear PDEs on low-rank tensor manifolds", *J. Sci. Comput.*, **97** (2023)

[5] A. Rodgers, "Paratt: An MPI Parallel TT Library for solutions to high dimensional PDEs." *Github*, `https://github.com/akrodger/paratt` (2023)