O(N) unconditionally stable methods through kernel based Successive Convolution

Andrew Christlieb

Michigan State University
Department of Computational Mathematics, Science and Engineering

ICERM workshop
Holistic Design of Time-Dependent PDE Discretizations

Jan 14, 2022
William Sands (Student)

Hyoseon Yang (Former Post Doc at Kyung Hee University)

Yan Jiang (Former Post Doc now at USTC)

Wei Gou (Former Post Doc now at Texas Tech)

Thank you to: AFOSR, ONR, NSF and DoE
Rothe’s Method


▶ Analysis and Numerics Summary Papers

▶ Transverse Method of Lines (Show that treating all time at once lades to constant high order methods in time)
  ▶ Annamaria Mazzia and Francesca Mazzia: 1997, “High-order transverse schemes for the numerical solution of PDEs”, *Journal of computational and applied mathematics*
Rothe’s Method

- Parallel in Time (All of time, but distributed computing)

- Method of Lines Transpose (Use of Greens functions to address BVP)
  - Matthew Causley, Andrew J Christlieb, Benjamin Ong, and Lee Van Groningen: 2014 “Method of lines transpose: An implicit solution to the wave equation”, *Mathematics of Computation*
Rothe’s Method

- Successive Convolution (Expanding the spatial operator in continues well behaved convergent expansions)
What are we talking about?

Two big themes:

- First - New spatial discretization that makes explicit time stepping methods be unconditionally stable. 
  Method designed for multi-core computing, 
  Avoids iteration and all to one communication of implicit solves.

- Second - The method has been developed to solve PDEs with several boundary conditions.

We will go back and forth between the two as we put the story together.
Non-ideal MHD ($\nabla \cdot \mathbf{B} = 0$, $\mathbf{B} = \nabla \times \mathbf{A}$, and $\mathbf{J} = \nabla \times \mathbf{B}$)

\[
\partial_t \mathbf{A} + (\nabla \times \mathbf{A}) \times \mathbf{u} = - \left( \eta \mathbf{J} + \frac{1}{ne} \mathbf{J} \times \nabla \times \mathbf{A} + \frac{m_e}{ne^2} \partial_t \mathbf{J}, \right)
\]

\[
\partial_t \mathbf{J} = \partial_t \nabla \times \nabla \times \mathbf{A}
\]

*Hamilton-Jacobi equations - Constrained Transport for MHD*
Consider 1D Hamilton-Jacobi equations

\[ \phi_t + H(\phi_x) = 0. \]

We construct the following semi-discrete scheme

\[ \frac{d}{dt} \phi_i(t) + \hat{H}(\phi^-_{x,i}, \phi^+_{x,i}) = 0, \]

where \( \phi^-_{x,i} \) and \( \phi^+_{x,i} \) are the approximations to \( \phi_x \) at \( x_i \) obtained by left-biased and right-biased methods, respectively.

**Applications** in diverse fields:

- Optimal control, seismic waves, crystal growth, robotic navigation, image processing, and calculus of variations.
- Burgers equation, magnetic scalar/vector potential equations for MHD, Navier-Stokes with Maxwell equation can also be casted as H-J equations.

**Reference:**

What to look for in this talk:

Consider $\partial_t y = c \partial_x f(y)$.

- Explicit time stepping, for example, forward Euler
  $$\frac{y^{n+1} - y^n}{\Delta t} = c \partial_x f(y)$$
  $$\frac{y_{j}^{n+1} - y_{j}^{n}}{\Delta t} = c \frac{f(y_{j}^{n}) - f(y_{j-1}^{n})}{\Delta x}$$

- CFL $\Delta t < \frac{\Delta x}{c}$, finite propagation $\rightarrow$ stable.

(a) Stable

(b) Unstable
Kernel based approximation

- Terms like $\partial_x$ and $\partial_x(n(x)\partial_x)$ are made up of linear derivatives “$\partial_x$”.

- **Big idea:** $\partial_x \sim$ global instead of local gives ($\Delta t > \frac{\Delta x}{c} \rightarrow$ stable).

- Approximate $\partial_x$ via a fast $O(N)$ convolution integral with a kernel that gives an $O(\Delta t^k)$ approximation.

(c) Stable for all $\Delta t$

- For linear PDEs, provably unconditionally stable even when using explicit time stepping.
Kernel Based Expansion of \( \partial_x \),
\[
\sim \partial_x(\cdot) \text{ with } \sum_{i=0}^{p} (\int_{a}^{x}(\cdot))^{i} \, dy \text{ in } \mathcal{O}(N).
\]
Prototypical linear PDEs:

- Linear advection equation: \((\partial_t - c \partial_x)u = 0\)
- Diffusion equation: \((\partial_t - \nu \partial_{xx})u = 0\)
- Wave equation: \((\partial_{tt} - c^2 \partial_{xx})u = 0\)

Semi-discretize the equations in time (could use a BDF method to get high order):

- Linear advection equation: \((\mathcal{I} - c \Delta t \partial_x)u^{n+1} = u^n\)
- Diffusion equation: \((\mathcal{I} - \nu \Delta t \partial_{xx})u^{n+1} = u^n\)
- Wave equation: \((\mathcal{I} - c^2 \Delta t^2 \partial_{xx})u^{n+1} = 2u^n - u^{n-1}\)

with an identity operator \(\mathcal{I}\).

**Observation:** the operators \(\mathcal{I} \pm \frac{1}{\alpha} \partial_x\) arise in each of examples.

\[
(\mathcal{I} - \frac{1}{\alpha^2} \partial_{xx}) = (\mathcal{I} - \frac{1}{\alpha} \partial_x)(\mathcal{I} + \frac{1}{\alpha} \partial_x)
\]

We introduce the operators for simplicity:

\[
\mathcal{L}_L := \mathcal{I} - \frac{1}{\alpha} \partial_x \quad \text{and} \quad \mathcal{L}_R := \mathcal{I} + \frac{1}{\alpha} \partial_x.
\]
Consider the univariate Helmholtz operators

\[ \mathcal{L}_L[u] = (\mathcal{I} - \frac{1}{\alpha} \partial_x) u(x) \quad \text{and} \quad \mathcal{L}_R[u] = (\mathcal{I} + \frac{1}{\alpha} \partial_x) u(x). \]

We define convolution with the Green's function by the integral operators

\[ I_L[u](x) := \alpha \int_x^b e^{-\alpha(\tau-x)} u(\tau) d\tau \quad \text{and} \quad I_R[u](x) := \alpha \int_a^x e^{-\alpha(x-\tau)} u(\tau) d\tau \]

so that the Helmholtz operators are inverted as

\[ \mathcal{L}_L^{-1} = Be^{-\alpha(b-x)} + I_L[u](x) \quad \text{and} \quad \mathcal{L}_R^{-1} = Ae^{-\alpha(x-a)} + I_R[u](x), \]

with boundary terms \( B \) and \( A \) in the Homogeneous solutions.
Kernel expansion of $\partial^+_x$

\[
\mathcal{L}_L = (\mathcal{I} - \frac{1}{\alpha} \partial_x)(\cdot) + B.C., \quad \mathcal{L}_L^{-1} = B^{n+1} e^{-\alpha(b-x)} + \alpha \int_x^b e^{-\alpha(\tau-x)}(\cdot) d\tau.
\]

Rewriting the first equation:

\[
-\frac{1}{\alpha} \partial_x = (\mathcal{L}_L - \mathcal{I}) + B.C.
\]

Defining a new operator $\mathcal{D}_L = \mathcal{I} - \mathcal{L}_L^{-1}$, then this gives

\[
\mathcal{L}_L = (\mathcal{I} - \mathcal{D}_L)^{-1}
\]

and we expand $(\mathcal{I} - \mathcal{D}_L)^{-1}$ in a power series

\[
-\frac{1}{\alpha} \partial^+_x = (\mathcal{L}_L - \mathcal{I}) = \mathcal{L}_L(\mathcal{I} - \mathcal{L}_L^{-1}) = (\mathcal{I} - \mathcal{D}_L)^{-1} \mathcal{D}_L = \sum_{p=1}^{\infty} \mathcal{D}_L^p.
\]
Kernel based approximation

A right traveling wave gives left facing operator ($x$ moving from $b$ to $a$) to give a right facing derivative:

\[
\mathcal{L}_L = (1 - \frac{1}{\alpha} \partial_x)(\cdot) + B.C. , \quad \mathcal{L}_L^{-1} = B^{n+1} e^{-\alpha (b-x)} + \alpha \int_x^b e^{-\alpha (\tau-x)}(\cdot) d\tau
\]

\[
\partial^+ x = -\alpha \sum_{p=1}^{\infty} \mathcal{D}^p_L
\]

and a left traveling wave gives right facing operator ($x$ moving from $a$ to $b$) to give a left facing derivative:

\[
\mathcal{L}_R = (1 + \frac{1}{\alpha} \partial_x)(\cdot) + B.C. , \quad \mathcal{L}_R^{-1} = A^{n+1} e^{-\alpha (x-a)} + \alpha \int_a^x e^{-\alpha (x-\tau)}(\cdot) d\tau
\]

\[
\partial^- x = \alpha \sum_{p=1}^{\infty} \mathcal{D}^p_R
\]

where

\[
\mathcal{D}_L = \mathcal{I} - \mathcal{L}_L^{-1} \quad \text{and} \quad \mathcal{D}_R = \mathcal{I} - \mathcal{L}_R^{-1}.
\]
Kernel Based Methods: 
Description and Analysis
Multiple ways to establish consistency, easiest way is to observe the relations between our series and the resolvent expansion [Abadias et. al. (2017)].

**Theorem (Consistency)**

If $\phi \in C^{k+1}[a, b]$, then

$$
\| \partial_x \phi(x) + \alpha \sum_{p=1}^{k} D^p_L[\phi](x) \|_\infty = O(\Delta t^k), \quad \| \partial_x \phi(x) - \alpha \sum_{p=1}^{k} D^p_R[\phi](x) \|_\infty = O(\Delta t^k).
$$

**Reference:**

Kernel based scheme 1 - The whole picture

\((\ast)\) \(\partial_t y - c \partial_x y = 0 + B.C.\)

**Single-step and Multi-stage method**

Apply SSP-RK3 for the equation \((\ast)\):

\[
\begin{align*}
y^{(1)} &= y^n - \Delta tc \partial_x y^n, \\
y^{(2)} &= \frac{3}{4} y^n + \frac{1}{4} \left( y^{(1)} - \Delta tc \partial_x y^{(1)} \right), \\
y^{n+1} &= \frac{1}{3} y^n + \frac{2}{3} \left( y^{(2)} - \Delta tc \partial_x y^{(2)} \right).
\end{align*}
\]

Then we use the approximation

\[
\partial_x^- = \alpha \sum_{p=1}^{3} D_R^p + O(\Delta t^3).
\]

**Reference:**

\[(\ast) \ \partial_t y - c \partial_x y = 0 \ + \ B.C.\]

**Multi-step and Single-stage method**

Integrating the equation \(\ast\):

\[
\int_{t}^{t+\Delta t} \partial_\tau y(\tau, x) \, d\tau - c \partial_x \int_{t}^{t+\Delta t} y(\tau, x) \, d\tau = 0
\]

gives the update

\[
y^{n+1} = y^n - c \partial_x P[y^n](x)
\]

with

\[
P[y^n](x) := \int_{t^n}^{t^{n+1}} y(\tau, x) \, d\tau \approx \Delta t \left( \frac{5}{12} y^{n-2}(x) - \frac{4}{3} y^{n-1}(x) + \frac{23}{12} y^n(x) \right).
\]

Then we use the approximation

\[
\partial_x^- = \alpha \sum_{p=1}^{3} D^p_R + O(\Delta t^3).
\]

**Reference:**

Stability

Consider Hamilton-Jacobi equations on $[a, b]$ with periodic boundary conditions. We take the parameter $\alpha = \beta / (c \Delta t)$ with $c = \max |H'(\phi)|$. Here we address the operators $D_L$ and $D_R$ with periodic boundary treatments

$$D_L^p[\phi](a) = D_L^p[\phi](b) \quad \text{and} \quad D_R^p[\phi](a) = D_R^p[\phi](b);$$

**Theorem (Stability: Von Neumann analysis)**

Suppose the suggested method employs the $k$-th order SSP RK scheme or $k$-th order multistep strategy derived above for $k = 1, 2, 3$. There exists constant $\beta_{k, \text{max}} > 0$ such that the scheme is A-stable provided $0 < \beta \leq \beta_{k, \text{max}}$.

```
<table>
<thead>
<tr>
<th>k</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_{k, \text{max}}$</td>
<td>2</td>
<td>1</td>
<td>1.243</td>
</tr>
</tbody>
</table>
```

**Reference:**

Kernel Based Methods

with Successive Convolution and WENO
Recap: For distributed spatial domain \( \{x_i\} \), we approximate

\[
\phi^+_x(x_i) \approx -\alpha \sum_{p=1}^{k} D^p_L[\phi](x_i),
\]

\[
\phi^-_x(x_i) \approx \alpha \sum_{p=1}^{k} D^p_R[\phi](x_i),
\]

where \( D^p_L = D[\mathcal{L}^{p-1}_L] \) and \( D^p_R = D[\mathcal{L}^{p-1}_R] \) for \( p > 1 \) and

\[
\mathcal{D}_L = \mathcal{I} - \mathcal{L}^{-1}_L, \quad \mathcal{D}_R = \mathcal{I} - \mathcal{L}^{-1}_R,
\]

with

\[
\mathcal{L}^{-1}_L[v](x_i) = \alpha \int_{x_i}^{b} e^{-\alpha(y-x_i)} v(y) dy + Be^{-\alpha(b-x)},
\]

\[
\mathcal{L}^{-1}_R[v](x_i) = \alpha \int_{a}^{x_i} e^{-\alpha(x_i-y)} v(y) dy + Ae^{-\alpha(x-a)}.
\]
Successive convolution for \( I_L \) and \( I_R \)

Then

\[
I_L[v](x_i) := \alpha \int_{x_i}^{b} e^{-\alpha(y-x_i)} v(y) \, dy \quad \text{and} \quad I_R[v](x_i) := \alpha \int_{a}^{x_i} e^{-\alpha(x_i-y)} v(y) \, dy
\]

can be calculated by

\[
I_L[v](x_i) = e^{-\alpha \Delta x_i} I_L[v](x_{i+1}) + J_L[v](x_i), \quad i = 0, \ldots, N - 1,
\]
\[
I_R[v](x_i) = e^{-\alpha \Delta x_i} I_R[v](x_{i-1}) + J_R[v](x_i), \quad i = 1, \ldots, N,
\]

where \( I_L[v](x_N) = 0 \) and \( I_R[v](x_0) = 0 \), and

\[
J_L[v](x_i) = \alpha \int_{x_i}^{x_{i+1}} v(y) e^{-\alpha(y-x_i)} \, dy \quad \text{and} \quad J_R[v](x_i) = \alpha \int_{x_{i-1}}^{x_i} v(y) e^{-\alpha(x_i-y)} \, dy.
\]

Therefore, once \( J_L[v](x_i) \) and \( J_R[v](x_i) \) are computed for all \( i \), we then can obtain \( I_L[v](x_i) \) and \( I_R[v](x_i) \) via the recursive relation above.

Evaluate \( J_L[v](x_i) \) and \( J_R[v](x_i) \) with WENO methodology.
Consider the approximation for $J_R[v](x_i)$ on the 6-point stencil $\{x_{i-3}, \ldots, x_{i+2}\}$:

$$J_i := \alpha \int_{x_{i-1}}^{x_i} e^{-\alpha(x_i-s)} p(s) \, ds,$$

and

$$J_{i,r} := \alpha \int_{x_{i-1}}^{x_i} e^{-\alpha(x_i-s)} p_r(s) \, ds,$$

where $p$ and $p_r$ are interpolants to $v$ on $S(i)$ and $S_r(i)$ that satisfy

$$J_i = J_R[v](x_i) + \alpha \mathcal{O}(\Delta x^6) \quad \text{and} \quad J_{i,r} = J_R[v](x_i) + \alpha \mathcal{O}(\Delta x^4)$$

if $v$ is smooth. Then we can find linear weights $d_r$ s.t.

$$J_i = \sum_{r=0}^{2} d_r J_{i,r}.$$
From
\[ \sum_{r=0}^{2} d_r J_{i,r} \approx J_R[v](x_i), \]
we construct the final approximation to \( J_R[v](x_i) \) as
\[ \sum_{r=0}^{2} \omega_r J_{i,r} \]
where the nonlinear weight \( \omega_r \) for each of local solutions \( J_{i,r} \) is defined by
\[ \omega_r = \frac{\alpha_r}{\sum_k \alpha_k} , \quad \alpha_r = \frac{d_r}{\epsilon + \beta_r} \]
so that \( \omega_r \approx d_r \) and
\[ \sum_{r=0}^{2} \omega_r J_{i,r} = J_R[v](x_i) + \alpha O(\Delta x^6). \]

Here \( \beta_r \) is derived to measure the smoothness of the function on each of substencils \( S_r(i) \), \( r = 0, 1, 2 \) and \( \epsilon > 0 \).
Numerical Results (Part 1)
Linear equations

Example

\[
\phi_t + (\phi_x + \phi_y + 1) = 0, \quad -2 \leq x, y \leq 2,
\]

\[
\phi(x, y, 0) = -\cos(\pi(x + y)/2).
\]

<table>
<thead>
<tr>
<th>CFL</th>
<th>(N_x \times N_y)</th>
<th>(k = 1. \beta = 1.)</th>
<th>(k = 2. \beta = 0.5.)</th>
<th>(k = 3. \beta = 0.6.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>error</td>
<td>order</td>
<td>error</td>
</tr>
<tr>
<td>0.5</td>
<td>20 \times 20</td>
<td>1.28e-02</td>
<td>–</td>
<td>1.73e-01</td>
</tr>
<tr>
<td></td>
<td>40 \times 40</td>
<td>3.22e-03</td>
<td>1.987</td>
<td>4.48e-02</td>
</tr>
<tr>
<td></td>
<td>80 \times 80</td>
<td>8.07e-04</td>
<td>1.998</td>
<td>1.13e-02</td>
</tr>
<tr>
<td></td>
<td>160 \times 160</td>
<td>2.02e-04</td>
<td>2.000</td>
<td>2.82e-03</td>
</tr>
<tr>
<td></td>
<td>320 \times 320</td>
<td>5.05e-05</td>
<td>2.000</td>
<td>7.06e-04</td>
</tr>
<tr>
<td>2</td>
<td>20 \times 20</td>
<td>1.94e-01</td>
<td>–</td>
<td>9.92e-01</td>
</tr>
<tr>
<td></td>
<td>40 \times 40</td>
<td>5.09e-02</td>
<td>1.931</td>
<td>5.66e-01</td>
</tr>
<tr>
<td></td>
<td>80 \times 80</td>
<td>1.29e-02</td>
<td>1.984</td>
<td>1.73e-01</td>
</tr>
<tr>
<td></td>
<td>160 \times 160</td>
<td>3.23e-03</td>
<td>1.996</td>
<td>4.48e-02</td>
</tr>
<tr>
<td></td>
<td>320 \times 320</td>
<td>8.07e-04</td>
<td>1.999</td>
<td>1.13e-02</td>
</tr>
</tbody>
</table>

Table: \(L_\infty\)-errors and orders of accuracy at \(T = 2\).
Nonlinear equations in uniform grids

Example

\[ \phi_t + \frac{1}{2}(\phi_x + \phi_y + 1)^2 = 0, \quad -2 \leq x, y \leq 2, \]

\[ \phi^0(x, y) = -\cos(\frac{\pi(x + y)}{2}). \]

<table>
<thead>
<tr>
<th>CFL</th>
<th>( N_x \times N_y )</th>
<th>( k = 1. \beta = 1. )</th>
<th>( k = 2. \beta = 0.5. )</th>
<th>( k = 3. \beta = 0.6. )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \text{error} )</td>
<td>( \text{order} )</td>
<td>( \text{error} )</td>
<td>( \text{order} )</td>
</tr>
<tr>
<td>0.5</td>
<td>20 ( \times ) 20</td>
<td>5.48e-02</td>
<td>–</td>
<td>1.68e-02</td>
</tr>
<tr>
<td></td>
<td>40 ( \times ) 40</td>
<td>2.98e-02</td>
<td>0.877</td>
<td>4.72e-03</td>
</tr>
<tr>
<td></td>
<td>80 ( \times ) 80</td>
<td>1.63e-02</td>
<td>0.874</td>
<td>1.28e-03</td>
</tr>
<tr>
<td></td>
<td>160 ( \times ) 160</td>
<td>8.37e-03</td>
<td>0.961</td>
<td>3.30e-04</td>
</tr>
<tr>
<td></td>
<td>320 ( \times ) 320</td>
<td>4.27e-03</td>
<td>0.970</td>
<td>8.46e-05</td>
</tr>
<tr>
<td>2</td>
<td>20 ( \times ) 20</td>
<td>1.66e-01</td>
<td>–</td>
<td>2.41e-01</td>
</tr>
<tr>
<td></td>
<td>40 ( \times ) 40</td>
<td>9.94e-02</td>
<td>0.739</td>
<td>5.96e-02</td>
</tr>
<tr>
<td></td>
<td>80 ( \times ) 80</td>
<td>5.64e-02</td>
<td>0.817</td>
<td>1.70e-02</td>
</tr>
<tr>
<td></td>
<td>160 ( \times ) 160</td>
<td>3.05e-02</td>
<td>0.886</td>
<td>4.74e-03</td>
</tr>
<tr>
<td></td>
<td>320 ( \times ) 320</td>
<td>1.63e-02</td>
<td>0.906</td>
<td>1.28e-03</td>
</tr>
</tbody>
</table>

Table: \( L_\infty \)-errors and orders of accuracy at \( T = 0.5/\pi^2 \).
Example

\[ \phi_t + \frac{1}{2}(\phi_x + \phi_y + 1)^2 = 0, \quad -2 \leq x, y \leq 2, \]
\[ \phi(x, y, 0) = -\cos(\pi(x + y)/2). \]

<table>
<thead>
<tr>
<th>CFL</th>
<th>(N_x \times N_y)</th>
<th>(k = 1. \beta = 1.)</th>
<th>(k = 2. \beta = 0.5.)</th>
<th>(k = 3. \beta = 0.6.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>error</td>
<td>order</td>
<td>error</td>
</tr>
<tr>
<td>0.5</td>
<td>40 (\times) 40</td>
<td>1.09e-01</td>
<td>–</td>
<td>6.43e-02</td>
</tr>
<tr>
<td></td>
<td>80 (\times) 80</td>
<td>6.14e-02</td>
<td>0.835</td>
<td>1.66e-02</td>
</tr>
<tr>
<td></td>
<td>160 (\times) 160</td>
<td>3.30e-02</td>
<td>0.895</td>
<td>4.47e-03</td>
</tr>
<tr>
<td></td>
<td>320 (\times) 320</td>
<td>1.71e-02</td>
<td>0.945</td>
<td>1.19e-03</td>
</tr>
<tr>
<td></td>
<td>640 (\times) 640</td>
<td>8.72e-03</td>
<td>0.975</td>
<td>3.05e-04</td>
</tr>
<tr>
<td>2</td>
<td>40 (\times) 40</td>
<td>2.16e-01</td>
<td>–</td>
<td>3.21e-01</td>
</tr>
<tr>
<td></td>
<td>80 (\times) 80</td>
<td>1.72e-01</td>
<td>0.327</td>
<td>1.98e-01</td>
</tr>
<tr>
<td></td>
<td>160 (\times) 160</td>
<td>1.10e-01</td>
<td>0.653</td>
<td>6.45e-02</td>
</tr>
<tr>
<td></td>
<td>320 (\times) 320</td>
<td>6.15e-02</td>
<td>0.835</td>
<td>1.66e-02</td>
</tr>
<tr>
<td></td>
<td>640 (\times) 640</td>
<td>3.30e-02</td>
<td>0.897</td>
<td>4.46e-03</td>
</tr>
</tbody>
</table>

Table: \(L_\infty\)-errors and orders of accuracy at \(T = 0.5/\pi^2\).
Riemann problem in nonuniform grids: CFL=7 for small cells

Example

\[ \phi_t + \sin(\phi_x + \phi_y) = 0, \quad -1 \leq x, y \leq 1 \]

\[ \phi(x, y, 0) = \pi(|y| - |x|). \]

Construct a nonuniform mesh consisting of 60 × 60 grid points using a geometric series, selecting the ratio between the smallest cell size and the biggest cell size to be 1 : 7.

(a) Generated mesh

(b) Numerical solutions’ surfaces and contour at \( T = 1 \).
Example 1 (H-J form: Burgers equation)

\[ \phi_t + \frac{1}{2}(\phi_x + \phi_y + 1)^2 = 0, \quad -2 \leq x, y \leq 2, \]
\[ \phi^0(x, y) = -\cos(\pi(x + y)/2). \]

<table>
<thead>
<tr>
<th>(N_x \times N_y)</th>
<th>Runge-Kutta 3</th>
<th>Multistep-3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>error</td>
<td>order</td>
</tr>
<tr>
<td>20 \times 20</td>
<td>5.63e-02</td>
<td>–</td>
</tr>
<tr>
<td>40 \times 40</td>
<td>8.39e-03</td>
<td>2.74</td>
</tr>
<tr>
<td>80 \times 80</td>
<td>9.15e-04</td>
<td>3.19</td>
</tr>
<tr>
<td>160 \times 160</td>
<td>6.26e-05</td>
<td>3.87</td>
</tr>
</tbody>
</table>

Example 2 (Advection-Diffusion equation)

\[ u_t + u_x + u_y = 0.01(u_{xx} + u_{yy}), \quad -2 \leq x, y \leq 2, \]
\[ u(x, y, 0) = \sin(\pi(x + y)/2). \]

<table>
<thead>
<tr>
<th>(N_x \times N_y)</th>
<th>Runge-Kutta 3</th>
<th>Multistep-3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>error</td>
<td>order</td>
</tr>
<tr>
<td>20 \times 20</td>
<td>5.81e-01</td>
<td>–</td>
</tr>
<tr>
<td>40 \times 40</td>
<td>1.45e-01</td>
<td>2.00</td>
</tr>
<tr>
<td>80 \times 80</td>
<td>2.07e-02</td>
<td>2.81</td>
</tr>
<tr>
<td>160 \times 160</td>
<td>2.69e-03</td>
<td>2.94</td>
</tr>
</tbody>
</table>
Example 1 (H-J form: Burgers equation)

\[ \phi_t + \frac{1}{2}(\phi_x + \phi_y + 1)^2 = 0, \quad -2 \leq x, y \leq 2, \]
\[ \phi^0(x, y) = -\cos(\pi(x + y)/2). \]

<table>
<thead>
<tr>
<th>( N_x \times N_y )</th>
<th>Runge-Kutta 3</th>
<th>Multistep-3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>error</td>
<td>order</td>
</tr>
<tr>
<td>20 \times 20</td>
<td>6.88e-02</td>
<td>–</td>
</tr>
<tr>
<td>40 \times 40</td>
<td>5.92e-02</td>
<td>0.22</td>
</tr>
<tr>
<td>80 \times 80</td>
<td>8.34e-03</td>
<td>2.83</td>
</tr>
<tr>
<td>160 \times 160</td>
<td>9.21e-04</td>
<td>3.18</td>
</tr>
</tbody>
</table>

Example 2 (Advection-Diffusion equation)

\[ u_t + u_x + u_y = 0.01(u_{xx} + u_{yy}), \quad -2 \leq x, y \leq 2, \]
\[ u(x, y, 0) = \sin(\pi(x + y)/2). \]

<table>
<thead>
<tr>
<th>( N_x \times N_y )</th>
<th>Runge-Kutta 3</th>
<th>Multistep-3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>error</td>
<td>order</td>
</tr>
<tr>
<td>20 \times 20</td>
<td>8.03e-01</td>
<td>–</td>
</tr>
<tr>
<td>40 \times 40</td>
<td>5.84e-01</td>
<td>0.46</td>
</tr>
<tr>
<td>80 \times 80</td>
<td>1.45e-01</td>
<td>2.00</td>
</tr>
<tr>
<td>160 \times 160</td>
<td>2.07e-02</td>
<td>2.81</td>
</tr>
</tbody>
</table>
Generalized Kernel Based Methods
Kernel based approximation with arbitrary boundary conditions

From the linear wave equation

$$\partial_t y - c \partial_x y = 0 \quad + B.C., \quad x \in [a, b],$$

the numerical solution is updated from

$$y^{n+1}(x) = y^{n+1}(b) e^{-\alpha(b-x)} + \alpha \int_x^b e^{-\alpha(\tau-x)} y^n(\tau) d\tau$$

with $\alpha = 1/(c\Delta t)$. Therefore the operators are defined by

$$\mathcal{L}_L = (I - \frac{1}{\alpha} \partial_x)(\cdot) + B.C., \quad \mathcal{L}_L^{-1}[(\cdot)^n] = B e^{-\alpha(b-x)} + \alpha \int_x^b e^{-\alpha(\tau-x)} (\cdot)^n d\tau$$

with $B := B[(\cdot)^n] = (\cdot)^{n+1}(b)$.

Now, the boundary term can be approximated by Taylor expansion:

$$B[y^n] := y^{n+1}(b) = y^n(b) + \Delta t y^n_t(b) + O(\Delta t^2)$$

$$= y^n(b) + c\Delta t y^n_x(b) + O(\Delta t^2)$$

$$= y^n(b) + \frac{1}{\alpha} y^n_x(b) + O(\Delta t^2).$$

Reference:

Lemma

Suppose $\phi \in C^{k+1}[a, b]$ and we set the operator $\mathcal{D}_L$ with general boundary treatments. Then we can obtain that

$$
\mathcal{D}_L[\phi](x) = -\sum_{p=1}^{k} \frac{1}{\alpha^p} \partial_x^p \phi(x) + \sum_{p=2}^{k} \frac{1}{\alpha^p} \partial_x^p \phi(b) e^{-\alpha(b-x)} - \frac{1}{\alpha^{k+1}} I_L[\partial_x^{k+1} \phi](x).
$$

Consider $k = 2$ case:

$$
\mathcal{D}_L[\phi](x) = -\frac{1}{\alpha} \phi'(x) - \frac{1}{\alpha^2} \phi''(x) + \frac{1}{\alpha^2} \phi''(b) e^{-\alpha(b-x)} - \frac{1}{\alpha^3} I_L[\phi'''](x).
$$

Instead of adding successively defined term $\mathcal{D}_L^p = \mathcal{D}_L[\mathcal{D}_L^{p-1}]$, $p \geq 2$, we define

$$
\tilde{\mathcal{D}}_L[\phi](x) := \mathcal{D}_L[\phi](x) - \frac{1}{\alpha^2} \phi''(b) e^{-\alpha(b-x)}
$$

and apply the $\mathcal{D}_L$ operator:

$$
\mathcal{D}_L[\tilde{\mathcal{D}}_L[\phi]](x) = \frac{1}{\alpha^2} \phi''(x) + \frac{1}{\alpha^3} I_L[\phi'''](x) + \frac{1}{\alpha^3} I_L^2[\phi'''](x).
$$

Then we obtain higher order approximation to $\phi'(x)$ with the modified partial sum:

$$
\mathcal{D}_L[\phi](x) + \mathcal{D}_L[\tilde{\mathcal{D}}_L[\phi]](x) - \frac{1}{\alpha^2} \phi''(b) e^{-\alpha(b-x)} = -\frac{1}{\alpha} \phi'(x) + O\left(\frac{1}{\alpha^3}\right).
$$
Kernel based approximation with arbitrary boundary conditions

Using the general boundary treatments, we can modify the partial sums for the first derivative operators:

\[ \phi_x^+(x) \approx \mathcal{P}_k^L[\phi](x) = -\alpha \sum_{p=1}^{k} \tilde{D}_L^p[\phi](x) + \frac{1}{\alpha} \partial^2_x \phi(b) e^{-\alpha(b-x)} \]

where

\[
\begin{cases}
\tilde{D}_L^1 = D_L, \\
\tilde{D}_L^p = D_L[\tilde{D}_L^{p-1}] - \left(\frac{1}{\alpha}\right)^p \partial^p_x \phi(b) e^{-\alpha(b-x)}, & p \geq 2,
\end{cases}
\]

It can be easily derived for the case \( \phi_x^-(x) \approx \mathcal{P}_k^R[\phi](x) \) by a similar process.

**Theorem**

Suppose \( \phi \in C^{k+1}[a,b], \ k = 1, 2, 3 \). Then, the modified partial sums satisfy

\[ \|\partial_x \phi(x) - \mathcal{P}_k^*[\phi](x)\|_\infty \leq C \left(\frac{1}{\alpha}\right)^k \|\partial^{k+1}_x \phi(x)\|_\infty \]

where \( * \) indicates \( L \) and \( R \) operators.
Numerical Results (Part 2)
Linear equations

Example

\[ \phi_t + (\phi_x) = 0, \quad -1 \leq x \leq 1, \]
\[ \phi(x, 0) = -\cos(\pi x). \]

Table: (Periodic boundary conditions) \( L_\infty \)-errors and orders of accuracy at \( T = 1 \).

<table>
<thead>
<tr>
<th>CFL</th>
<th>( N )</th>
<th>( k = 1. )</th>
<th>( k = 2. )</th>
<th>( k = 3. )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>error</td>
<td>order</td>
<td>error</td>
</tr>
<tr>
<td>0.5</td>
<td>40</td>
<td>3.07e-01</td>
<td>–</td>
<td>7.92e-02</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>1.69e-01</td>
<td>0.864</td>
<td>2.07e-02</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>8.84e-02</td>
<td>0.933</td>
<td>5.27e-03</td>
</tr>
<tr>
<td></td>
<td>320</td>
<td>4.52e-02</td>
<td>0.967</td>
<td>1.33e-03</td>
</tr>
<tr>
<td></td>
<td>640</td>
<td>2.29e-02</td>
<td>0.983</td>
<td>3.33e-04</td>
</tr>
</tbody>
</table>

Table: (Dirichlet boundary conditions) \( L_\infty \)-errors and orders of accuracy at \( T = 1 \).

<table>
<thead>
<tr>
<th>CFL</th>
<th>( N )</th>
<th>( k = 1. )</th>
<th>( k = 2. )</th>
<th>( k = 3. )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>error</td>
<td>order</td>
<td>error</td>
</tr>
<tr>
<td>0.5</td>
<td>40</td>
<td>3.06e-01</td>
<td>–</td>
<td>7.93e-02</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>1.68e-01</td>
<td>0.862</td>
<td>2.06e-02</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>8.83e-02</td>
<td>0.931</td>
<td>5.23e-03</td>
</tr>
<tr>
<td></td>
<td>320</td>
<td>4.52e-02</td>
<td>0.966</td>
<td>1.31e-03</td>
</tr>
<tr>
<td></td>
<td>640</td>
<td>2.29e-02</td>
<td>0.983</td>
<td>3.29e-04</td>
</tr>
</tbody>
</table>
Nonlinear equations

Example

\[ \phi_t + \frac{1}{2} (\phi_x + 1)^2 = 0, \quad -1 \leq x \leq 1, \]
\[ \phi(x, 0) = -\cos(\pi x). \]

Table: (Periodic boundary) \( L_{\infty} \)-errors and orders of accuracy at \( T = 0.3/\pi^2 \).

<table>
<thead>
<tr>
<th>CFL</th>
<th>( N )</th>
<th>( k = 1 )</th>
<th>( k = 2 )</th>
<th>( k = 3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>error</td>
<td>order</td>
<td>error</td>
</tr>
<tr>
<td>0.5</td>
<td>40</td>
<td>3.52e-02</td>
<td>–</td>
<td>2.81e-03</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>6.59e-03</td>
<td>2.419</td>
<td>1.15e-03</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>3.30e-03</td>
<td>0.997</td>
<td>4.00e-04</td>
</tr>
<tr>
<td></td>
<td>320</td>
<td>1.66e-03</td>
<td>0.996</td>
<td>1.26e-04</td>
</tr>
<tr>
<td></td>
<td>640</td>
<td>8.30e-04</td>
<td>0.995</td>
<td>3.43e-05</td>
</tr>
</tbody>
</table>

Table: (Dirichlet boundary) \( L_{\infty} \)-errors and orders of accuracy at \( T = 0.3/\pi^2 \).

<table>
<thead>
<tr>
<th>CFL</th>
<th>( N )</th>
<th>( k = 1 )</th>
<th>( k = 2 )</th>
<th>( k = 3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>error</td>
<td>order</td>
<td>error</td>
</tr>
<tr>
<td>0.5</td>
<td>40</td>
<td>3.47e-02</td>
<td>–</td>
<td>2.81e-03</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>6.59e-03</td>
<td>2.395</td>
<td>1.01e-03</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>3.30e-03</td>
<td>0.997</td>
<td>3.76e-04</td>
</tr>
<tr>
<td></td>
<td>320</td>
<td>1.66e-03</td>
<td>0.996</td>
<td>1.11e-04</td>
</tr>
<tr>
<td></td>
<td>640</td>
<td>8.30e-04</td>
<td>0.995</td>
<td>3.02e-05</td>
</tr>
</tbody>
</table>
A propagating problem with Dirichlet boundaries in nonuniform grids

Example

\[ \phi_t - \sqrt{\phi_x^2 + \phi_y^2} + 1 = 0, \quad x^2 + y^2 \leq 1 \]

\[ \phi(x, y, 0) = \sin \left( \frac{\pi}{2} (x^2 + y^2) \right) \]

with the Dirichlet boundary \( \phi(x, y, t) = 1 + t \) for all \( x^2 + y^2 = 1 \).

(a) Domain discretization

(b) Propagating solutions at \( T = 0, 0.6 \) and 1.2

CFL = 0.5

CFL = 2
Magnetohydrodynamics
Examples
Consider the MHD equations

\[
\begin{bmatrix}
\partial_t \\
\rho u \\
\varepsilon \\
B
\end{bmatrix}
\begin{bmatrix}
\rho \\
\rho u \\
\varepsilon \\
B
\end{bmatrix}
+ \nabla \cdot \begin{bmatrix}
\rho u \\
\rho u \otimes u + p_{\text{tot}} \mathbb{I} - B \otimes B \\
\mathbf{u}(\varepsilon + p_{\text{tot}}) - B(\mathbf{u} \cdot B) \\
\mathbf{u} \otimes \mathbf{B} - \mathbf{B} \otimes \mathbf{u}
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
0 \\
\nabla \times \left( \frac{1}{\eta} \mathbf{J} + \frac{1}{ne} \mathbf{J} \times \mathbf{B} \right)
\end{bmatrix}
\]

\[\nabla \cdot \mathbf{B} = 0,\]

with the equation of state as

\[\varepsilon = \frac{p}{\gamma - 1} + \frac{\rho \| \mathbf{u} \|^2}{2} + \frac{\| \mathbf{B} \|^2}{2}.\]

**Methods to overcome \( \nabla \cdot \mathbf{B} = 0 \):**
The projection method, 8-wave scheme, hyperbolic divergence cleaning method, the constrained transport method.

**Reference:**
Observe: B is a field, therefore there exists an A, magnetic vector potential, such that \( \nabla \times A = B \).

Goal: Express B in MHD in terms of A.

Start with Electron inertial equation, solve for E:

\[
E = -u \times B + \eta J + \frac{1}{ne} J \times B + \frac{m_e}{ne^2} \partial_t J.
\]

In Maxwell, replace E in \( \partial_t B = -\nabla \times E \), with the above,

\[
\partial_t B = \nabla \times \left( u \times B - \left( \eta J + \frac{1}{ne} J \times B + \frac{m_e}{ne^2} \partial_t J \right) \right).
\]

Noting that \( \nabla \times A = B \) and using the Weyl gauge,

\[
\partial_t A + (\nabla \times A) \times u = - \left( \eta J + \frac{1}{ne} J \times \nabla \times A + \frac{m_e}{ne^2} \partial_t J \right).
\]

Reference:
Example 1: 2D smooth vortex problem

The initial conditions are
\[
(\rho, u^1, u^3, u^3, p, B^1, B^2, B^3) = (1, 1, 1, 0, 1, 0, 0, 0)
\]
with perturbations on \(u^1, u^2, B^1, B^2\) and \(p\) as:
\[
(\delta u^1, \delta u^2) = \frac{\mu}{2\pi} e^{0.5(1-r^2)} (-y, x),
\]
\[
(\delta B^1, \delta B^2) = \frac{\kappa}{2\pi} e^{0.5(1-r^2)} (-y, x),
\]
\[
\delta p = \frac{\mu^y (1 - r^2) - \kappa^2}{8\pi^2} e^{(1-r^2)},
\]
and the initial magnetic potential is
\[
A^3(0, x, y) = \frac{\mu}{2\pi} e^{0.5(1-r^2)},
\]
where \(r^2 = x^2 + y^2\). The vortex strength is taken as \(\mu = 5.389489439\) and \(\kappa = \sqrt{2}\mu\) such that the lowest pressure is around \(5.3 \times 10^{-12}\) which happens in the center of the vortex. The domain is \([-10, 10] \times [-10, 10]\) and periodic boundary condition is used on all four boundaries.
Example 1: 2D smooth vortex in resistive MHD

- Solve 2D smooth vortex problem with several resistive terms $\eta$.
- Contour plots of $\|\mathbf{B}\|$ at time $t = 20$ with $\Delta x = \Delta y = 0.2$:

(i) FD scheme

(ii) Kernel based scheme

(a) $\eta = 0$  
(b) $\eta = 0.01$  
(c) $\eta = 0.1$  
(d) $\eta = 0.5$  
(e) $\eta = 1$
The initial conditions are

\[(\rho, u^1, u^2, u^3, B^1, B^2, B^3) = (1, 0, 0, 0, 50/\sqrt{2\pi}, 50/\sqrt{2\pi}, 0)\]

with a spherical pressure pulse

\[p = \begin{cases} 
1000 & r \leq 0.1 \\
0.1 & \text{otherwise.}
\end{cases}\]

where \(r = \sqrt{x^2 + y^2 + z^2}\). The initial condition for magnetic potential:

\[A(0, x, y, z) = (0, 0, 50/\sqrt{2\pi}y - 50/\sqrt{2\pi}x).\]

We use domain \([-0.5, 0.5] \times [-0.5, 0.5] \times [-0.5, 0.5]\) and outflow boundary conditions are applied everywhere.
Example 2: 3D Blast wave

(i) Density contour plots at $t = 0.01$ with $150 \times 150 \times 150$ grid point

(ii) Pressure contour plots at $t = 0.01$ with $150 \times 150 \times 150$ grid point

(a) FD scheme  
(b) Kernel based scheme
THANK YOU!!
Appendix
Parallel Algorithm
Domain Decomposition

\[ \mathcal{D}_L = I - \mathcal{L}_L^{-1}, \quad \mathcal{D}_R = I - \mathcal{L}_R^{-1}, \]

with

\[ \mathcal{L}_L^{-1}[v](x_i) = \alpha \int_{x_i}^{b} e^{-\alpha(y-x_i)} v(y) \, dy + B e^{-\alpha(b-x)} , \]

\[ \mathcal{L}_R^{-1}[v](x_i) = \alpha \int_{a}^{x_i} e^{-\alpha(x_i-y)} v(y) \, dy + A e^{-\alpha(x-a)} . \]
Non-blocking All Reduce Parallel Code Diagram

Reference:
Weak scaling results: up to 9 nodes

![Graphs showing weak scaling results for Advection, Diffusion, and Hamilton-Jacobi problems with different DOF/node values.](image)

- DOF/node = 1681^2
- DOF/node = 3361^2
- DOF/node = 6721^2
- DOF/node = 13441^2
- DOF/node = 26881^2
Weak scaling results: up to 49 nodes

- **Advection**
  - DOF/node = 3361
- **Diffusion**
  - DOF/node = 13441
- **Hamilton-Jacobi**
  - DOF/node = 26881

![Graphs showing weak scaling results for Advection, Diffusion, and Hamilton-Jacobi](image)
Strong scaling results: up to 9 nodes

![Graph showing Strong scaling results](image)

- Advection
- Diffusion
- Hamilton-Jacobi

- DOF/node/s: ×10^9
- Efficiency

DOF:
- DOF = 1681^2
- DOF = 3361^2
- DOF = 6721^2
- DOF = 13441^2
- DOF = 26881^2

Nodes:
- 1
- 4
- 9

Efficiency:
- 0.0
- 0.2
- 0.4
- 0.6
- 0.8
- 1.0

Advection
- DOF = 1681^2
- DOF = 3361^2
- DOF = 6721^2
- DOF = 13441^2
- DOF = 26881^2

Diffusion
- DOF = 1681^2
- DOF = 3361^2
- DOF = 6721^2
- DOF = 13441^2
- DOF = 26881^2

Hamilton-Jacobi
- DOF = 1681^2
- DOF = 3361^2
- DOF = 6721^2
- DOF = 13441^2
- DOF = 26881^2
The Hartmann flow model for MHD is taken to be incompressible ($\nabla \cdot \mathbf{v} = 0$), and $\rho = 1, p = 1$. The momentum and magnetic field equations are given as

$$
\rho \left[ \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} \right] = -\nabla p + \mathbf{f} + N(\mathbf{j} \times \mathbf{B}) + \frac{1}{R} \nabla^2 \mathbf{v},
$$

$$
\frac{\partial \mathbf{B}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{B} = (\mathbf{B} \cdot \nabla)\mathbf{v} + \frac{1}{Rm} \nabla^2 \mathbf{B}.
$$

With the assumptions of linearity in above equations, we have that

$$
\mathbf{v} = v_x(y)\hat{x} \quad \text{and} \quad \mathbf{B} = \hat{y} + \frac{Rm}{Ha} B_x(y)\hat{x}.
$$

The geometry of the problem consists of two infinite parallel plates and there is flow in the gap between the plates.
Example 3: Hartmann problem with PEC boundary conditions

Figure: Induced magnetic field for the Hartmann problem at $Ha = 0, 2, 5, 10$.

(a) [Muller]

(b) Kernel based scheme

Reference: