Hyperbolic Problems: High Order Methods and Model Order Reduction

Davide Torlo

Supervisor: Rémi Abgrall

Universität Zürich
Institut für Mathematik

2nd June 2020, Zürich

Motivation: hyperbolic PDEs

Hyperbolic PDE

$$\partial_t u(x,t) + \nabla \cdot F(u(x,t)) = S(u(x,t)), \quad x \in \Omega,\, t \in \mathbb{R}^+$$

Models
- Linear Transport
- Euler's
- Shallow water
- Kinetic models

Features
- Discontinuities
- Waves

Properties
- Conservation
- Asymptoticity

Numerical Analysis
- Space discretization ($\mathcal{N}$)
- Time discretization
- Numerical solvers

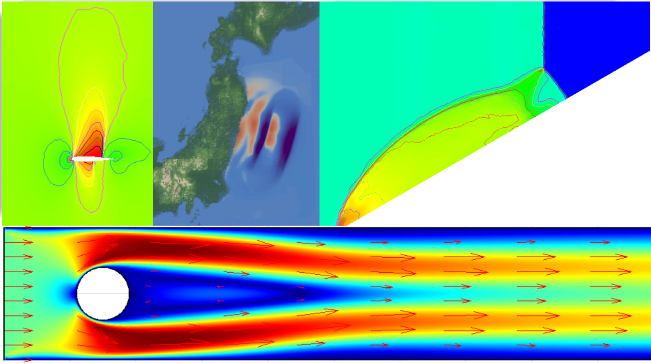
Motivation: hyperbolic PDEs

Hyperbolic PDE

$$\partial_t u(x,t) + \nabla \cdot F(u(x,t)) = S(u(x,t)), \quad x \in \Omega,\, t \in \mathbb{R}^+$$

Models

- Linear Transport
- Euler's
- Shallow water
- Kinetic models



Motivation: hyperbolic PDEs

Hyperbolic PDE

$$\partial_t u(x,t) + \nabla \cdot F(u(x,t)) = S(u(x,t)), \quad x \in \Omega,\, t \in \mathbb{R}^+$$

Models

- Linear Transport
- Euler's
- Shallow water
- Kinetic models

Features

- Discontinuities
- Waves

Properties

- Conservation
- Asymptoticity

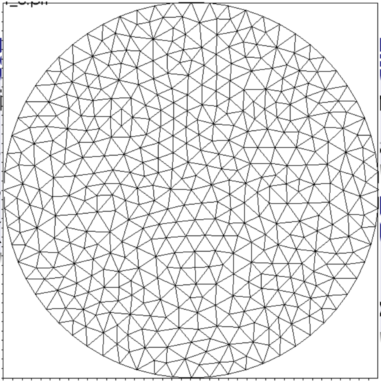
Numerical Analysis

- Space discretization ($\mathcal{N}$)
- Time discretization
- Numerical solvers

Motivation: hyperbolic PDEs

Hyperbolic PDE

$$\partial_t u(x,t) + \nabla \cdot F(u(x,t)) = S(u(x,t)), \quad x \in \Omega,\ t \in \mathbb{R}^+$$



Mo

- Li
- E
- S
- K

s

- ntinuities

es

- rvation
- ototicity

Numerical Analysis

- Space discretization ($\mathcal{N}$)
- Time discretization
- Numerical solvers

Motivation: hyperbolic PDEs

Hyperbolic PDE

$$\partial_t u(x,t) + \nabla \cdot F(u(x,t)) = S(u(x,t)), \quad x \in \Omega,\, t \in \mathbb{R}^+$$

Models

- Linear Transport
- Euler's
- Shallow water
- Kinetic models

Features

- Discontinuities
- Waves

Properties

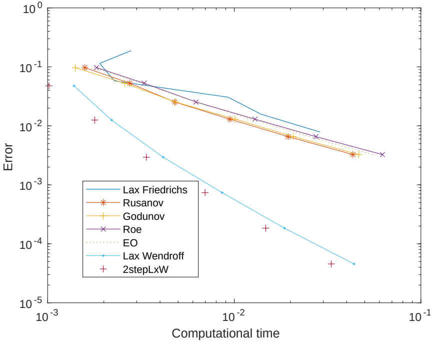
- Conservation
- Asymptoticity

Numerical Analysis

- Space discretization ($\mathcal{N}$)
- Time discretization
- Numerical solvers

Computational costs vs accuracy

- Numerical simulations $\Longrightarrow$ Computational times
- Scale up to hours, days, months



Possible solutions

- High order methods
- Model order reduction

PhD Thesis

1. Hyperbolic Problems
2. High order time integration for ODEs
   - Runge–Kutta
   - Deferred Correction
   - Modified Patankar Deferred Correction
3. High order numerical methods for PDEs
   - Finite Element
   - Finite Volume
   - Residual Distribution
   - IMEX RD DeC for kinetic models
4. Model order reduction
   - POD EIM Greedy for hyperbolic PDEs
   - Application to UQ
   - Arbitrary Lagrangian Eulerian formulation

PhD Thesis

1. Hyperbolic Problems
2. High order time integration for ODEs
   - Runge–Kutta
   - Deferred Correction
   - Modified Patankar Deferred Correction
3. High order numerical methods for PDEs
   - Finite Element
   - Finite Volume
   - Residual Distribution
   - IMEX RD DeC for kinetic models
4. Model order reduction
   - POD EIM Greedy for hyperbolic PDEs
   - Application to UQ
   - Arbitrary Lagrangian Eulerian formulation

PhD Thesis

1. Hyperbolic Problems
2. High order time integration for ODEs
   - Runge–Kutta
   - Deferred Correction
   - Modified Patankar Deferred Correction
3. High order numerical methods for PDEs
   - Finite Element
   - Finite Volume
   - Residual Distribution
   - IMEX RD DeC for kinetic models
4. Model order reduction
   - POD EIM Greedy for hyperbolic PDEs
   - Application to UQ
   - Arbitrary Lagrangian Eulerian formulation

Outline

1. High Order Methods
   - High Order in Time: Deferred Correction
   - High Order in Space: Residual Distribution
   - IMEX RD DeC for Kinetic Models
2. Model Order Reduction for Hyperbolic Problems
   - Advection Dominated Problems in MOR
   - ALE Formulation
3. Conclusions and Perspectives

High order methods: Why?

- Given a threshold $\texttt{tol}$
- We want $\|\texttt{error}\| \leq \texttt{tol}$
- Minimize number of cells $\mathcal{N}$
- Computational $\texttt{time}$ depends on $\mathcal{N}$

- Order p
- $\|\texttt{error}\| \approx \frac{1}{\mathcal{N}^p}$
- $\mathcal{N} \approx \texttt{tol}^{1/p}$
- $p \uparrow \Rightarrow \texttt{time} \downarrow$

High order methods: Why?

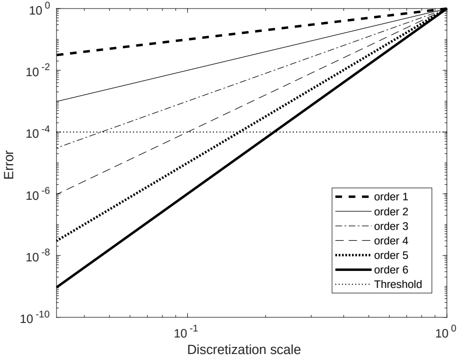
- Given a threshold $\texttt{tol}$
- We want $\|\texttt{error}\| \leq \texttt{tol}$
- Minimize number of cells $\mathcal{N}$
- Computational $\texttt{time}$ depends on $\mathcal{N}$

- Order p
- $\|\texttt{error}\| \approx \frac{1}{\mathcal{N}^p}$
- $\mathcal{N} \approx \texttt{tol}^{1/p}$
- $p \uparrow \Rightarrow \texttt{time} \downarrow$

High order methods: Why?

- Given a threshold $\texttt{tol}$
- We want $\|\texttt{error}\| \leq \texttt{tol}$
- Minimize number of cells $\mathcal{N}$
- Computational $\texttt{time}$ depends on $\mathcal{N}$

- Order p
- $\|\texttt{error}\| \approx \frac{1}{\mathcal{N}^p}$
- $\mathcal{N} \approx \texttt{tol}^{1/p}$
- $p \uparrow \Rightarrow \texttt{time} \downarrow$



High order in time: Deferred Correction (DeC)

Classical Runge Kutta (RK)

- One step method
- Internal stages

Explicit Runge Kutta

+ Simple to code
- Not easily generalizable to arbitrarily high order
- Stages $>$ order

Implicit Runge Kutta

+ Arbitrarily high order
- Require nonlinear solvers for nonlinear systems
- May not converge

Deferred Correction

- One step method
- Internal stages
- Can be seen as explicit RK

+ Explicit
+ Simple to code
+ Stages $=$ order
+ Arbitrarily high order
- Large memory storage

High order in time: Deferred Correction (DeC)

Classical Runge Kutta (RK)

- One step method
- Internal stages

Explicit Runge Kutta

\+ Simple to code

\- Not easily generalizable to arbitrarily high order

\- Stages $>$ order

Implicit Runge Kutta

\+ Arbitrarily high order

\- Require nonlinear solvers for nonlinear systems

\- May not converge

Deferred Correction

- One step method
- Internal stages
- Can be seen as explicit RK

\+ Explicit

\+ Simple to code

\+ Stages $=$ order

\+ Arbitrarily high order

\- Large memory storage

High order in time: Deferred Correction (DeC)

Classical Runge Kutta (RK)

- One step method
- Internal stages

Explicit Runge Kutta

\+ Simple to code

\- Not easily generalizable to arbitrarily high order

\- Stages $>$ order

Implicit Runge Kutta

\+ Arbitrarily high order

\- Require nonlinear solvers for nonlinear systems

\- May not converge

Deferred Correction

- One step method
- Internal stages
- Can be seen as explicit RK

\+ Explicit

\+ Simple to code

\+ Stages $=$ order

\+ Arbitrarily high order

\- Large memory storage

High order in time: Deferred Correction (DeC)

Classical Runge Kutta (RK)

- One step method
- Internal stages

Explicit Runge Kutta

\+ Simple to code

\- Not easily generalizable to arbitrarily high order

\- Stages $>$ order

Implicit Runge Kutta

\+ Arbitrarily high order

\- Require nonlinear solvers for nonlinear systems

\- May not converge

Deferred Correction

- One step method
- Internal stages
- Can be seen as explicit RK

\+ Explicit

\+ Simple to code

\+ Stages $=$ order

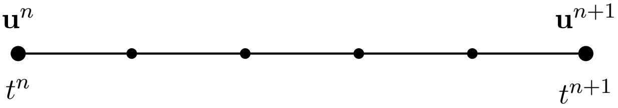
\+ Arbitrarily high order

\- Large memory storage

DeC

$$\frac{\partial \mathbf{u}(t)}{\partial t} + F(\mathbf{u}(t)) = 0 \qquad \text{(forget space for the moment)}$$

Discretization of each time step $[t^n, t^{n+1}]$.

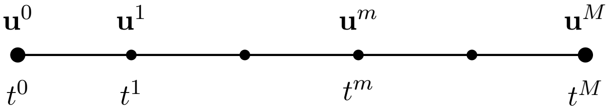


High order approximation of the equation in the Picard–Lindelöf form

$$\mathbf{u}^m = \mathbf{u}^0 - \int_{t^0}^{t^m} F(\mathbf{u}(t))\mathrm{d}t.$$

$$\frac{\partial \mathbf{u}(t)}{\partial t} + F(\mathbf{u}(t)) = 0 \qquad \text{(forget space for the moment)}$$

Discretization of each time step $[t^n, t^{n+1}]$.



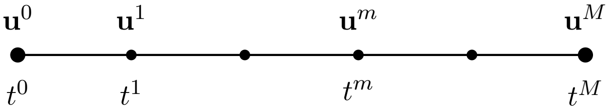
High order approximation of the equation in the Picard–Lindelöf form

$$\mathbf{u}^m = \mathbf{u}^0 - \int_{t^0}^{t^m} F(\mathbf{u}(t))\mathrm{d}t.$$

DeC

$$\frac{\partial \mathbf{u}(t)}{\partial t} + F(\mathbf{u}(t)) = 0 \qquad \text{(forget space for the moment)}$$

Discretization of each time step $[t^n, t^{n+1}]$.



High order approximation of the equation in the Picard–Lindelöf form

$$\mathbf{u}^m = \mathbf{u}^0 - \int_{t^0}^{t^m} F(\mathbf{u}(t)) \mathrm{d}t.$$

DeC: operators

$$\mathcal{L}^2 \begin{pmatrix} \mathbf{u}^0 \\ \dots \\ \mathbf{u}^M \end{pmatrix} := \begin{cases} \mathbf{u}^1 - \mathbf{u}^0 + \Delta t \sum_{r=0}^{M} \theta_r^1 F(\mathbf{u}^r) \\ \dots \\ \mathbf{u}^M - \mathbf{u}^0 + \Delta t \sum_{r=0}^{M} \theta_r^M F(\mathbf{u}^r) \end{cases}$$

$\mathcal{L}^2$

- implicit
- high order $= M+1$
- not easy to solve
- as implicit RK

DeC: operators

$$\mathcal{L}^2 \begin{pmatrix} \mathbf{u}^0 \\ \dots \\ \mathbf{u}^M \end{pmatrix} := \begin{cases} \mathbf{u}^1 - \mathbf{u}^0 + \Delta t \sum_{r=0}^{M} \theta_r^1 F(\mathbf{u}^r) \\ \dots \\ \mathbf{u}^M - \mathbf{u}^0 + \Delta t \sum_{r=0}^{M} \theta_r^M F(\mathbf{u}^r) \end{cases}$$

$$\mathcal{L}^1 \begin{pmatrix} \mathbf{u}^0 \\ \dots \\ \mathbf{u}^M \end{pmatrix} := \begin{cases} \mathbf{u}^1 - \mathbf{u}^0 + \Delta t \beta^1 F(\mathbf{u}^0) \\ \dots \\ \mathbf{u}^M - \mathbf{u}^0 + \Delta t \beta^M F(\mathbf{u}^0) \end{cases}$$

$\mathcal{L}^2$

- implicit
- high order $= M+1$
- not easy to solve
- as implicit RK

$\mathcal{L}^1$

- explicit
- low order $= 1$
- easy to solve
- as explicit Euler

DeC: algorithm

$$\mathcal{L}^1(\underline{\mathbf{u}})^m = \mathbf{u}^m - \mathbf{u}^0 + \Delta t \beta^m F(\mathbf{u}^0)$$

$$\mathcal{L}^2(\underline{\mathbf{u}})^m = \mathbf{u}^m - \mathbf{u}^0 + \Delta t \sum_{r=0}^{M} \theta_r^m F(\mathbf{u}^r)$$

DeC Iterations

$$\begin{cases} \underline{\mathbf{u}}^{(0)} = \mathbf{u}^n, & \underline{\mathbf{u}}^{(k)} := \left(\mathbf{u}^{(k),0}, \dots, \mathbf{u}^{(k),M}\right), \\ \mathcal{L}^1(\underline{\mathbf{u}}^{(k)}) = \mathcal{L}^1(\underline{\mathbf{u}}^{(k-1)}) - \mathcal{L}^2(\underline{\mathbf{u}}^{(k-1)}), & k = 1, \dots, p, \\ \mathbf{u}^{n+1} = \mathbf{u}^{(p),M} \end{cases}$$

DeC Theorem

- $\mathcal{L}^1$ coercive
- $\mathcal{L}^2 - \mathcal{L}^1$ Lipschitz

DeC is explicit method of order $\min\{p, M+1\}$

High Order Space Discretization: Residual Distribution

Classical solvers

Finite Element

+ Naturally high order
+ Compact stencil
- Inversion of mass matrix
- Tuning of stabilization terms

Finite Volume

+ Naturally conservative
+ No mass matrix
- More involved techniques for high order
- Choice of numerical flux

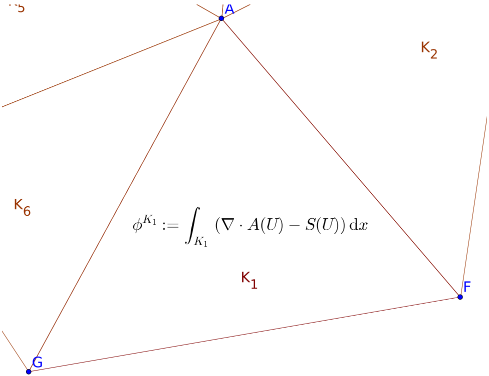
Residual Distribution

- FE based
+ High order
+ Compact stencil
+ Naturally conservative
+ No mass matrix
+ Easy to code
+ No need of Riemann solver
- Can recast some other FV, FE schemes
- Choice of residuals

$$\partial_t U + \nabla_x \cdot A(U) = S(U)$$

$$V_h = \{U \in L^2(\Omega_h, \mathbb{R}^D) \cap \mathcal{C}^0(\Omega_h),\, U|_K \in \mathbb{P}^p,\, \forall K \in \Omega_h\}.$$

Residual Distribution - Spatial Discretization



$$\phi^{K_1} := \int_{K_1} \left(\nabla \cdot A(U) - S(U)\right) \mathrm{d}x$$

Residual Distribution - Spatial Discretization

1. Define $\forall K \in \Omega_h$ a fluctuation term (total residual) $\phi^K = \int_K \left(\nabla \cdot A(U) - S(U)\right) \mathrm{d}x$
2. Define a nodal residual ϕ^K_σ $\forall \sigma \in K$:

$$\phi^K = \sum_{\sigma \in K} \phi^K_\sigma, \quad \forall K \in \Omega_h. \tag{1}$$

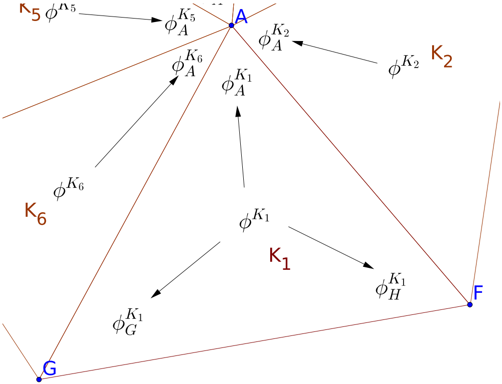
Choices in nodal residual

Basic algorithm (Galerkin), numerical fluxes (Rusanov), limiters (PSI), stabilization terms (SUPG).

3. The resulting scheme is

$$\int_K \partial_t U_\sigma + \sum_{K|\sigma \in K} \phi^K_\sigma = 0, \quad \forall \sigma \in D_h. \tag{2}$$

Residual Distribution - Spatial Discretization



Residual Distribution - Spatial Discretization

1. Define $\forall K \in \Omega_h$ a fluctuation term (total residual) $\phi^K = \int_K (\nabla \cdot A(U) - S(U))\, \mathrm{d}x$
2. Define a nodal residual ϕ^K_σ $\forall \sigma \in K$:

$$\phi^K = \sum_{\sigma \in K} \phi^K_\sigma, \quad \forall K \in \Omega_h. \tag{1}$$

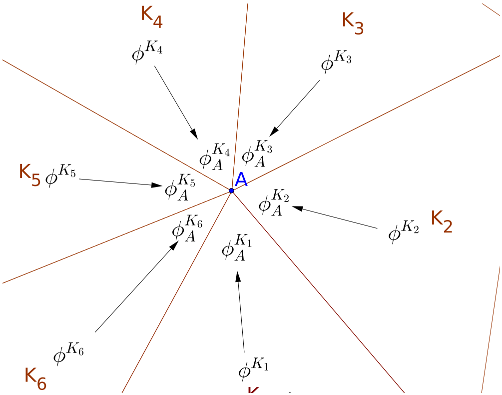
Choices in nodal residual

Basic algorithm (Galerkin), numerical fluxes (Rusanov), limiters (PSI), stabilization terms (SUPG).

3. The resulting scheme is

$$\int_K \partial_t U_\sigma + \sum_{K|\sigma \in K} \phi^K_\sigma = 0, \quad \forall \sigma \in D_h. \tag{2}$$

Residual Distribution - Spatial Discretization



Residual Distribution - Spatial Discretization

1. Define $\forall K \in \Omega_h$ a fluctuation term (total residual) $\phi^K = \int_K (\nabla \cdot A(U) - S(U))\, \mathrm{d}x$
2. Define a nodal residual $\phi_\sigma^K \; \forall \sigma \in K$:

$$\phi^K = \sum_{\sigma \in K} \phi_\sigma^K, \quad \forall K \in \Omega_h. \tag{1}$$

Choices in nodal residual

Basic algorithm (Galerkin), numerical fluxes (Rusanov), limiters (PSI), stabilization terms (SUPG).

3. The resulting scheme is

$$\int_K \partial_t U_\sigma + \sum_{K|\sigma \in K} \phi_\sigma^K = 0, \quad \forall \sigma \in D_h. \tag{2}$$

Residual Distribution - Spatial Discretization

1. Define $\forall K \in \Omega_h$ a fluctuation term (total residual) $\phi^K = \int_K (\nabla \cdot A(U) - S(U))\, \mathrm{d}x$
2. Define a nodal residual ϕ^K_σ $\forall \sigma \in K$:

$$\phi^K = \sum_{\sigma \in K} \phi^K_\sigma, \quad \forall K \in \Omega_h. \tag{1}$$

Choices in nodal residual

Basic algorithm (Galerkin), numerical fluxes (Rusanov), limiters (PSI), stabilization terms (SUPG).

3. The resulting scheme is

$$\int_K \partial_t U_\sigma + \sum_{K|\sigma \in K} \phi^K_\sigma = 0, \quad \forall \sigma \in D_h. \tag{2}$$

Method of Line $\Rightarrow$ Coupling with DeC

Kinetic and Relaxation Models

Kinetic and relaxation models $\Longleftrightarrow \varepsilon \ll 1$

Applications

- Boltzmann equations
- BGK models
- Multiphase flows
- Viscoelasticity problems

ε mean free path, relaxation parameter, reaction time

Asymptotic preserving (AP) property

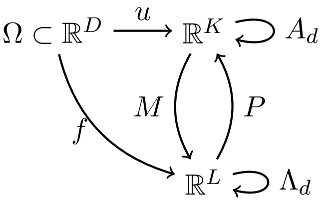
$$
\begin{array}{ccc}
\mathcal{F}^{\varepsilon}_{\Delta} & \xrightarrow{\varepsilon \to 0} & \mathcal{F}^{0}_{\Delta} \\
\Delta \to 0 \downarrow & & \downarrow \Delta \to 0 \\
\mathcal{F}^{\varepsilon} & \xrightarrow{\varepsilon \to 0} & \mathcal{F}^{0}
\end{array}
$$

Works of Jin, Bouchut, Natalini

A Kinetic Model

Kinetic relaxation models by D. Aregba-Driollet and R. Natalini[1].
Hyperbolic limit equation is

$$u_t + \sum_{d=1}^{D} \partial_{x_d} A_d(u) = 0, \quad u : \Omega \to \mathbb{R}^K.$$



Relaxation system

$$f_t^\varepsilon + \sum_{d=1}^{D} \Lambda_d \partial_{x_d} f^\varepsilon = \frac{1}{\varepsilon} \left(M(Pf^\varepsilon) - f^\varepsilon \right),$$

$Pf^\varepsilon \to u$ (AP property),

$P(M(u)) = u, \quad P\Lambda_d M(u) = A_d(u).$

Applications

- Boltzmann equations
- BGK models
- ≈ Multiphase flows

[1] D. Aregba-Driollet and R. Natalini. Discrete kinetic schemes for multidimensional systems of conservation laws. SIAM J. Numer. Anal., 37(6):1973–2004, 2000.

IMEX discretization - Kinetic Model

Stiff source term $\Rightarrow$ oscillations when $\varepsilon \ll \Delta t$

$\Delta t \approx \varepsilon$ not feasible

IMEX approach: IMplicit for source term, EXplicit for advection term

$$\frac{f^{n+1,\varepsilon} - f^{n,\varepsilon}}{\Delta t} + \sum_{d=1}^{D} \Lambda_d \partial_{x_d} f^{n,\varepsilon} = \frac{1}{\varepsilon}\left(M(Pf^{n+1,\varepsilon}) - f^{n+1,\varepsilon}\right) \tag{3}$$

$$f^{0,\varepsilon}(x) = f_0^{\varepsilon}(x)$$

How to treat non-linear implicit functions?

IMEX discretization - Kinetic Model

Stiff source term $\Rightarrow$ oscillations when $\varepsilon \ll \Delta t$
$\Delta t \approx \varepsilon$ not feasible
IMEX approach: IMplicit for source term, EXplicit for advection term

$$\frac{f^{n+1,\varepsilon} - f^{n,\varepsilon}}{\Delta t} + \sum_{d=1}^{D} \Lambda_d \partial_{x_d} f^{n,\varepsilon} = \frac{1}{\varepsilon}\left(M(Pf^{n+1,\varepsilon}) - f^{n+1,\varepsilon}\right) \tag{3}$$
$$f^{0,\varepsilon}(x) = f_0^{\varepsilon}(x)$$

How to treat non-linear implicit functions?
Recall: $PM(u) = u$ and $Pf^{\varepsilon} = u^{\varepsilon}$, so

$$\frac{u^{n+1,\varepsilon} - u^{n,\varepsilon}}{\Delta t} + \sum_{d=1}^{D} P\Lambda_d \partial_{x_d} f^{n,\varepsilon} = 0. \tag{4}$$

Find $u^{n+1,\varepsilon} = Pf^{n+1,\varepsilon}$ and substitute it in (3).
IMEX formulation = $\mathcal{L}^1$ (first order accurate).

IMEX discretization - Kinetic Model

Stiff source term ⇒ oscillations when $\varepsilon \ll \Delta t$
$\Delta t \approx \varepsilon$ not feasible
IMEX approach: IMplicit for source term, EXplicit for advection term

$$\frac{f^{n+1,\varepsilon} - \quad}{\Delta t} \quad \ldots \quad f^{n+1,\varepsilon}) \tag{3}$$

$$f^{0,\varepsilon}(x) = \ldots$$

DeC RD IMEX scheme

- Arbitrarily high order
  - Coercivity of $\mathcal{L}^1$
  - Lipschitz continuity of $\mathcal{L}^1 - \mathcal{L}^2$
- Asymptotic preserving ($Pf_h^\varepsilon \to u_h$)
- Computationally explicit

How to treat nor…
Recall: $PM(u) = \ldots$

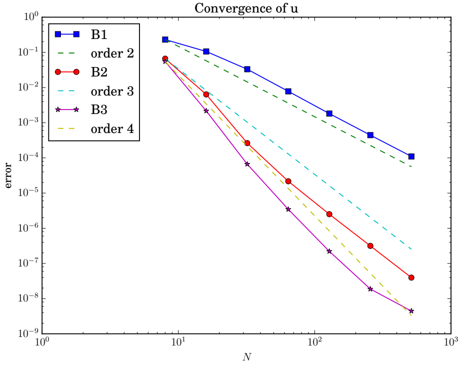
$$\frac{u^{n+1,\varepsilon} - u^{n,\varepsilon}}{\Delta t} + \sum_{d=1}^{D} P\Lambda_d \partial_{x_d} f^{n,\varepsilon} = 0. \tag{4}$$

Find $u^{n+1,\varepsilon} = Pf^{n+1,\varepsilon}$ and substitute it in (3).
IMEX formulation = $\mathcal{L}^1$ (first order accurate).

Numerical tests: Linear advection for convergence

$$u_t + u_x = 0, \quad x \in [0,1], \quad t \in [0,T],\, T = 0.12, \quad u_0(x) = e^{-80(x-0.4)^2},$$

outflow BC, $\varepsilon = 10^{-10}$.



(a) Scalar 1D convergence

Numerical tests: Euler's equation

Next simulations will be over Euler's equation

$$\begin{pmatrix} \rho \\ \rho v \\ E \end{pmatrix}_t + \begin{pmatrix} \rho v \\ \rho v^2 + p \\ (E+p)v \end{pmatrix}_x = 0, \qquad x \in [0,1],\, t \in [0,T] \tag{5}$$

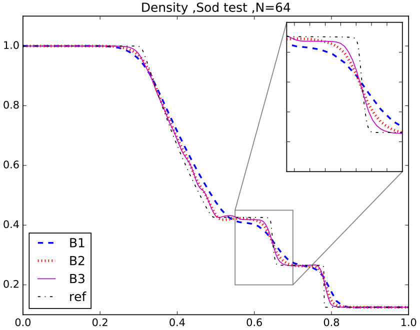
ρ is the density, v the speed, p the pressure and E the total energy.
The system is closed by the equation of state

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho v^2. \tag{6}$$

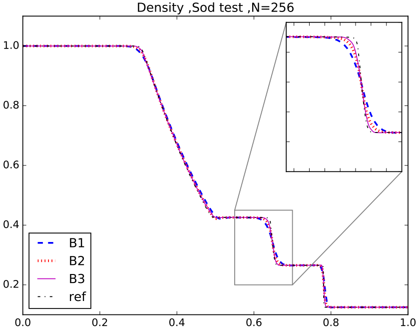
Numerical tests: Sod shock test

$\gamma = 1.4$, $T = 0.16$, outflow BC, $\varepsilon = 10^{-9}$, CFL = 0.2.

$$\rho_0 = \chi_{[0,0.5]}(x)+0.1\chi_{[0.5,1]}(x), \quad v_0 = 0, \quad p_0 = \chi_{[0,0.5]}(x)+0.125\chi_{[0.5,1]}(x).$$



(b) $N = 64$



(c) $N = 256$

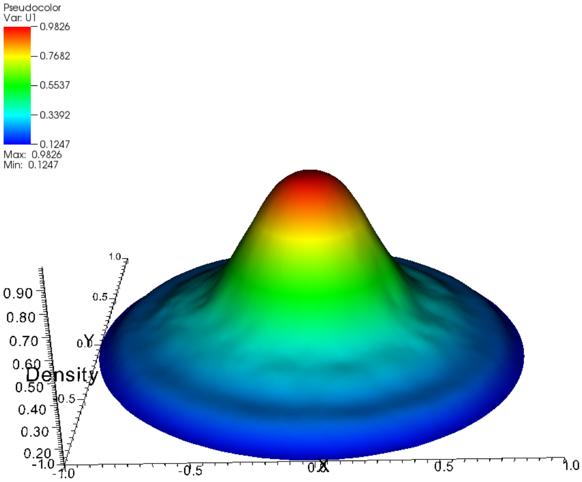
Numerical tests 2D: Sod shock test

Initial conditions are

$$\begin{pmatrix}\rho_0\\u_0\\v_0\\p_0\end{pmatrix}=\begin{pmatrix}1\\0\\0\\1\end{pmatrix}\text{ if } r<\frac{1}{2}, \qquad \begin{pmatrix}\rho_0\\u_0\\v_0\\p_0\end{pmatrix}=\begin{pmatrix}0.125\\0\\0\\0.1\end{pmatrix}\text{ if } r\geq\frac{1}{2}.$$

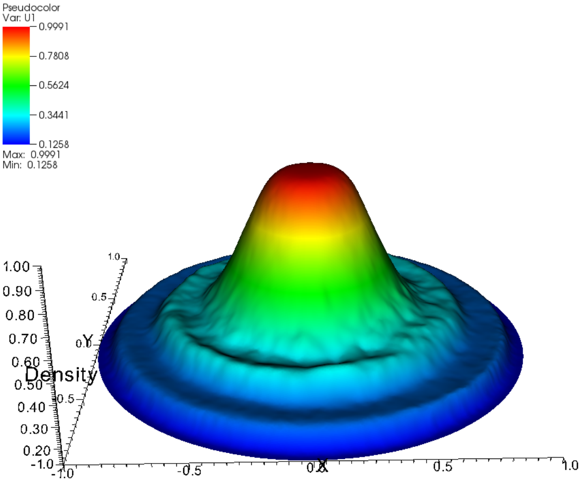
Here $r^2 = x^2 + y^2$, $\gamma = 1.4$, $\varepsilon = 10^{-9}$, $\lambda = 1.4$, CFL = 0.1, $T = 0.25$ and outflow boundary conditions.

Numerical tests 2D: Sod shock test



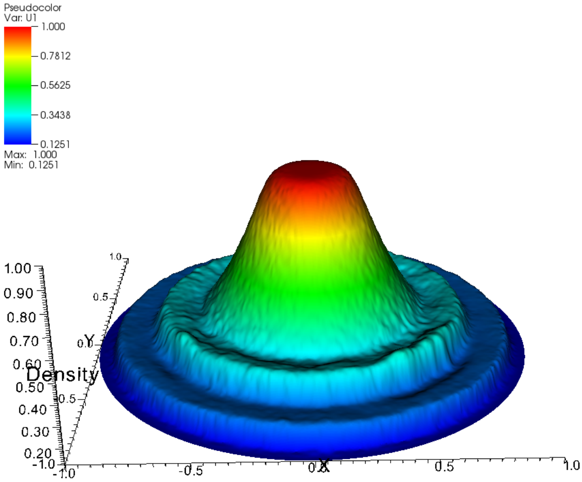
(d) $\mathbb{B}^1, N = 13548$

Numerical tests 2D: Sod shock test



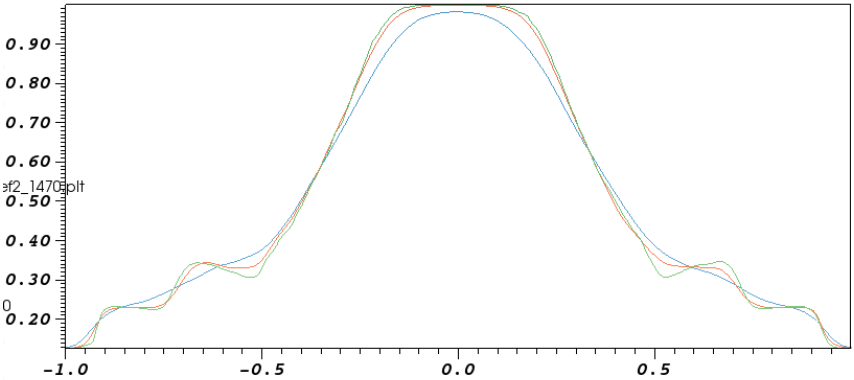
(e) $\mathbb{B}^2, N = 13548$

Numerical tests 2D: Sod shock test



(f) $\mathbb{B}^3, N = 13548$

Numerical tests 2D: Sod shock test



(g) Slices of $\mathbb{B}^1$ (blue), $\mathbb{B}^2$ (red) and $\mathbb{B}^3$ (green), $N = 13548$

Outline

Motivation: parametric hyperbolic problems

$$\begin{cases} \partial_t u(x,t,\boldsymbol{\mu}) + \nabla \cdot F(u,x,t,\boldsymbol{\mu}) = 0, \\ \mathbf{B}(u,\boldsymbol{\mu}) = g(t,\boldsymbol{\mu}) \\ u(x,t=0,\boldsymbol{\mu}) = u_0(x,\boldsymbol{\mu}) \end{cases}$$

- $\boldsymbol{\mu} \in \mathcal{P}$ influences boundaries, flux, initial conditions
- F nonlinear dependence on $\boldsymbol{\mu}$!

- Classical solvers FOM: FV, FEM, FD, RD. (Huge dimension $\mathcal{N}$)
- Many query task (UQ, optimization, etc.)

Offline phase

- Some snapshots of FOM (expensive)
- Find a RB space (dim N_{RB})
- Construct a ROM

Online phase

- Many fast evaluation of ROM (cheap)

MOR: Ingredients

- Solution manifold: $\mathcal{S} := \{u_{\mathcal{N}}(\cdot, t, \boldsymbol{\mu}) \in \mathbb{V}_{\mathcal{N}} : t \in \mathbb{R}^+, \boldsymbol{\mu} \in \mathcal{P}\}$
- Ansatz: $\mathcal{S} \approx \mathbb{V}_{N_{RB}} \subset \mathbb{V}_{\mathcal{N}}, \qquad N_{RB} \ll \mathcal{N}$
- Example: diffusion equation $u_t + \mu u_{xx} = 0$ with $u_0 = \sin(x\pi)$

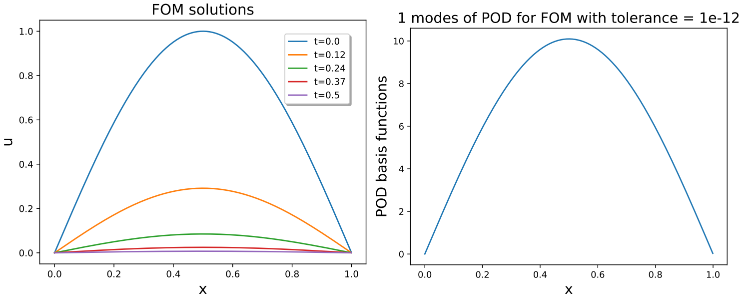


Figure: POD on a diffusion problem

MOR: Ingredients

Problem:

$$U^{n+1}(\boldsymbol{\mu}) = U^n(\boldsymbol{\mu}) - \mathcal{E}^n(U^n, \boldsymbol{\mu}), \quad U^n, U^{n+1} \in \mathbb{V}_{\mathcal{N}} \tag{7}$$

Objective:

$$\sum_{i=1}^{N_{RB}} \mathrm{u}_i^{n+1}(\boldsymbol{\mu})\psi_{RB}^i = \sum_{i=1}^{N_{RB}} \mathrm{u}_i^{n}(\boldsymbol{\mu})\psi_{RB}^i - \sum_{i=1}^{N_{RB}} \mathrm{E}^i(\mathrm{u}^n, \boldsymbol{\mu})\psi_{RB}^i, \quad \psi_{RB}^i \in \mathbb{V}_{\mathcal{N}}, \mathrm{u}^n, \mathrm{u}^{n+1} \in \mathbb{V}_{N_{RB}} \tag{8}$$

- POD $\Rightarrow$ RB space from the time evolution $U(\boldsymbol{\mu}^*, t)$, $t \in [0, T]$
- Greedy $\Rightarrow$ span the parameter space
- EIM $\Rightarrow$ Interpolates non–linear fluxes $\mathcal{E}$ in points τ_j function f_j
- Works of Haasdonk, Ohlberger, Maday, Farhat, Rozza, Patera, Willcox, etc.

MOR: Ingredients

Problem:

$$U^{n+1}(\boldsymbol{\mu}) = U^n(\boldsymbol{\mu}) - \mathcal{E}^n(U^n, \boldsymbol{\mu}), \quad U^n, U^{n+1} \in \mathbb{V}_{\mathcal{N}} \tag{7}$$

Objective:

$$\sum_{i=1}^{N_{RB}} \mathrm{u}_i^{n+1}(\boldsymbol{\mu})\psi_{RB}^i = \sum_{i=1}^{N_{RB}} \mathrm{u}_i^{n}(\boldsymbol{\mu})\psi_{RB}^i - \sum_{i=1}^{N_{RB}} \mathrm{E}^i(\mathrm{u}^n, \boldsymbol{\mu})\psi_{RB}^i, \qquad \psi_{RB}^i \in \mathbb{V}_{\mathcal{N}}, \mathrm{u}^n, \mathrm{u}^{n+1} \in \mathbb{V}_{N_{RB}} \tag{8}$$

- POD $\Rightarrow$ RB space from the time evolution $U(\boldsymbol{\mu}^*, t)$, $t \in [0, T]$
- Greedy $\Rightarrow$ span the parameter space
- EIM $\Rightarrow$ Interpolates non–linear fluxes $\mathcal{E}$ in points τ_j function f_j
- Works of Haasdonk, Ohlberger, Maday, Farhat, Rozza, Patera, Willcox, etc.

Online algorithm: PODEIM–Greedy

Solve the smaller system:

$$\sum_{i=1}^{N_{RB}} (\mathrm{u}_i^{n+1}(\boldsymbol{\mu}) - \mathrm{u}_i^{n}(\boldsymbol{\mu}))\psi_{RB}^i + \sum_{i=1}^{N_{RB}} \mathrm{E}^i(\mathrm{u}^n, \boldsymbol{\mu})\psi_{RB}^i = 0,$$

$$\sum_{i=1}^{N_{RB}} (\mathrm{u}_i^{n+1}(\boldsymbol{\mu}) - \mathrm{u}_i^{n}(\boldsymbol{\mu}))\psi_{RB}^i + \sum_{i=1}^{N_{RB}} \sum_{j=1}^{N_{EIM}} \tau_j(\mathcal{E}(\mathrm{u}_{RB}^n, \boldsymbol{\mu}))\Pi_{RB,i}(f_j)\psi_{RB}^i = 0$$

- $\Pi_{RB,i}(f_j)$ offline
- $\tau_j(\mathcal{E}(U^n, \boldsymbol{\mu}))$ online (evaluation of $\mathcal{E}$ in EIM point τ_j)
- MOR cost $\mathcal{O}(N_t N_{RB} N_{EIM})$ vs FOM cost $\mathcal{O}(N_t \mathcal{N})$
- Gain if $N_{RB}, N_{EIM} \ll \mathcal{N}$

Traveling shock, time evolution solution, little diffusion

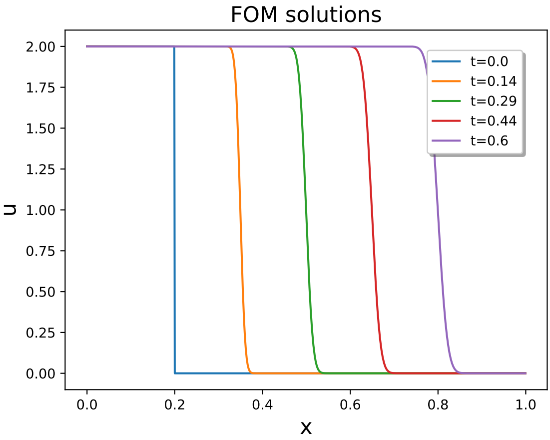


Figure: Solution of advection equation with shock IC

Traveling shock, POD, little diffusion

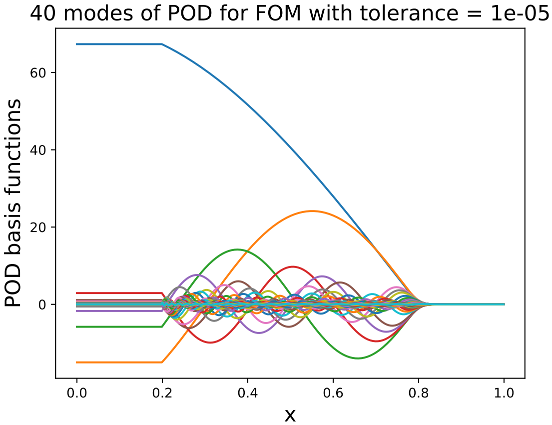


Figure: POD of time evolution of advection equation with shock IC

- Problem: one basis function for every shock position

Transfomation of the domain

Calibration map

$$\theta : \mathcal{P} \times [0, t_f] \to \Theta$$

- Smooth: $\theta(\cdot, \boldsymbol{\mu}) \in \mathcal{C}^1([0, t_f], \Theta)$ for all $\boldsymbol{\mu} \in \mathcal{P}$.

Geometry map

$$T : \Theta \times \mathcal{R} \to \Omega$$

- Bijection: $\exists T^{-1} : \Theta \times \Omega \to \mathcal{R}$ such that $T^{-1}(\theta, T(\theta, y)) = y$ for $y \in \mathcal{R}$ and $T(\theta, T^{-1}(\theta, x)) = x$ for $x \in \Omega$,
- Smooth: $T(\cdot, \cdot) \in \mathcal{C}^1(\Theta \times \mathcal{R}, \Omega)$, $T^{-1}(\cdot, \cdot) \in \mathcal{C}^1(\Theta \times \Omega, \mathcal{R})$.

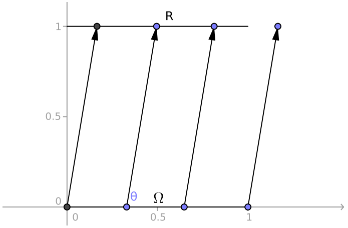
Goal

$$u_{\mathcal{N}}(T(\theta(t, \boldsymbol{\mu}), y), t, \boldsymbol{\mu}) \approx \bar{v}(y), \quad \forall \boldsymbol{\mu} \in \mathcal{P},\, t \in [0, t_f], y \in \mathcal{R}$$

Examples of Geometry Maps

Examples: θ is the point of maximum height or of steepest solution.

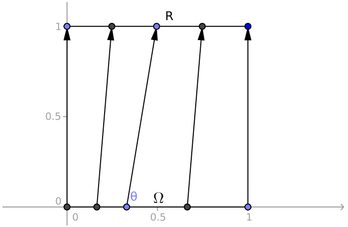
- Translation: $T(\theta, y) = y + \theta - 0.5$
  $T^{-1}(\theta, x) = x - \theta + 0.5$
- Dilatation: $T(\theta, y) = \frac{y\theta}{(2\theta-1)y+1-\theta}$
  $T^{-1}(\theta, x) = \frac{x(\theta-1)}{(2\theta-1)x-\theta}$
- High degree polynomials
- Gordon-Hall (see Cagniart, Crisovan, Maday, Abgrall)



Examples of Geometry Maps

Examples: θ is the point of maximum height or of steepest solution.

- Translation: $T(\theta, y) = y + \theta - 0.5$
  $T^{-1}(\theta, x) = x - \theta + 0.5$
- Dilatation: $T(\theta, y) = \frac{y\theta}{(2\theta-1)y+1-\theta}$
  $T^{-1}(\theta, x) = \frac{x(\theta-1)}{(2\theta-1)x-\theta}$
- High degree polynomials
- Gordon-Hall (see Cagniart, Crisovan, Maday, Abgrall)



Transformation examples

Dilatation (for other BCs): $T^{-1}(\theta,x) = x\frac{\theta-1}{(2\theta-1)x-\theta}$

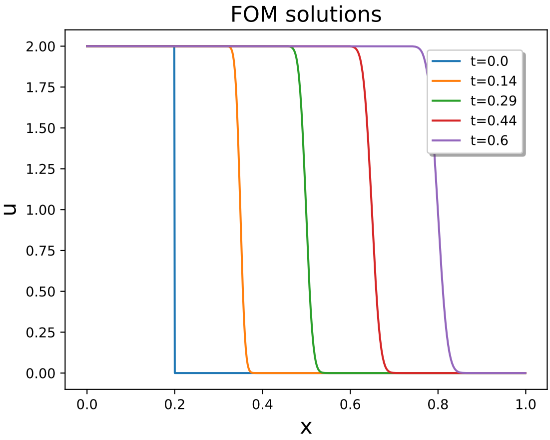


Figure: Original solutions for traveling discontinuity

Transformation examples

Dilatation (for other BCs): $T^{-1}(\theta, x) = x^{\frac{\theta-1}{(2\theta-1)x-\theta}}$

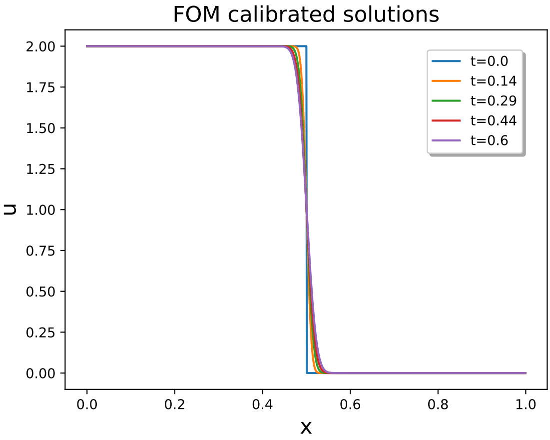


Figure: Calibrated solutions for traveling discontinuity

POD of calibrated solutions

Dilatation (for other BCs): $T^{-1}(\theta, x) = x^{\frac{\theta-1}{(2\theta-1)x-\theta}}$

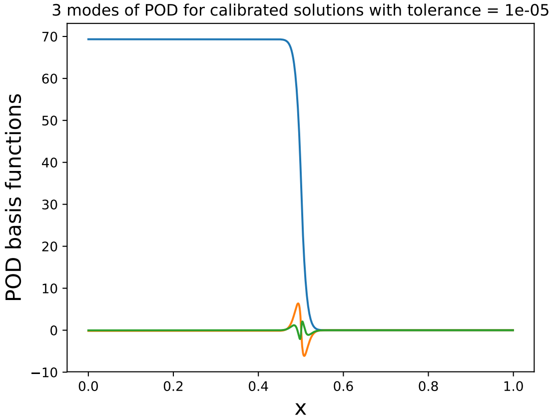


Figure: POD of calibrated solutions for traveling discontinuity

Arbitrary Lagrangian–Eulerian formulation

Influence on the online phase?

$$\frac{d}{dt}u(x,t,\boldsymbol{\mu}) + \frac{d}{dx}F(u(x,t,\boldsymbol{\mu}),\boldsymbol{\mu}) = 0$$

$$x := T(\theta(t,\boldsymbol{\mu}),y), \quad v(y,t,\boldsymbol{\mu}) := u(T(\theta(t,\boldsymbol{\mu}),y),t,\mu) = u(x,t,\boldsymbol{\mu})$$

$$\frac{\partial}{\partial t}v(y,\boldsymbol{\mu},t) + {\color{red}\frac{dy}{dx}}\frac{d}{dy}F(v,\mu) - {\color{red}\frac{dy}{dx}}{\color{green}\frac{dv}{dy}}{\color{blue}\frac{\partial T}{\partial t}} = 0$$

ALE formulation $\Longrightarrow$ EIM procedure on the reference domain $\mathcal{R}$.

- Jacobian $\frac{dy}{dx}$ low cost
- Flux $\frac{dv}{dy}$ low cost
- $\frac{\partial T}{\partial t}$??? $\Longrightarrow$ We must know $T(\theta(t,\boldsymbol{\mu}),y)$: easy parametrization in θ
- We must know θ

Learning of θ

- Quick evaluation of $\theta(t, \boldsymbol{\mu})$
  - Offline: $\boldsymbol{\mu}_i \in \mathcal{P}_{train} \Longrightarrow \theta(\boldsymbol{\mu}_i, t)$ with optimization or detection
  - Online: estimator $\hat{\theta}$ obtained with regression from $\theta(\boldsymbol{\mu}_i, t)$

Regression Maps

- Piecewise interpolation in $\boldsymbol{\mu}_i$ for every t^n
- Polynomial regression in $\boldsymbol{\mu}$ and t
- Neural network: multilayer perceptron

Modification to original algorithm

- Calibration map θ optimized on training samples $\theta(\boldsymbol{\mu}_k, t)$
- Regression on $\theta(\boldsymbol{\mu}_k, t)$ to have $\hat{\theta}$
- ALE formulation of the evolution operator $\mathcal{E}(\hat{\theta})$

Advection: traveling discontinuity

$$\begin{cases} u_t + \mu_0 u_x = 0,\ D = [0,1],\ T_{max} = 1.5, \text{ Dirichlet BC} \\ u_0(x, \boldsymbol{\mu}) = \begin{cases} \mu_1 & \text{if } x < 0.35 + 0.05\mu_2 \\ 0 & \text{else} \end{cases} \\ \mu_0 \sim \mathcal{U}([0,2]),\ \mu_1, \mu_2 \sim \mathcal{U}([-1,1]) \end{cases}$$

Without calibration

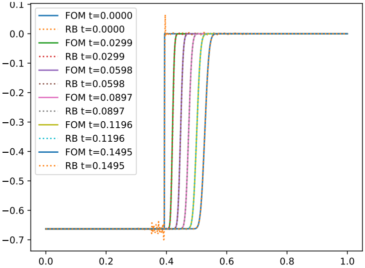
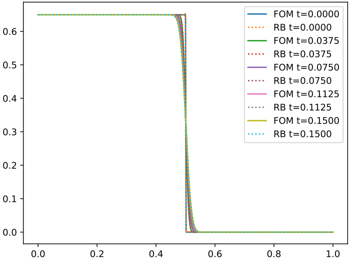
RB dim	64
EIM dim	124
FOM time	49 s
RB time	9 s
RB/FOM time	18%

With calibration: Poly2

RB dim	17
EIM dim	22
FOM time	125 s
RB time	6 s
RB/FOM time	5%

Advection: traveling discontinuity

$$\begin{cases} u_t + \mu_0 u_x = 0, \; D = [0,1], \; T_{max} = 1.5, \text{ Dirichlet BC} \\ u_0(x, \boldsymbol{\mu}) = \begin{cases} \mu_1 & \text{if } x < 0.35 + 0.05\mu_2 \\ 0 & \text{else} \end{cases} \\ \mu_0 \sim \mathcal{U}([0,2]), \; \mu_1, \mu_2 \sim \mathcal{U}([-1,1]) \end{cases}$$

Without calibration	With calibration: Poly2
	

Burgers: formation and motion of a shock

$$\begin{cases} u_t + \mu_0 (u^2/2)_x = 0,\ D = [0,\pi],\ T_{max} = 0.15, \text{ periodic BC} \\ u_0(x,\boldsymbol{\mu}) = |\sin(x+\mu_1)| + 0.1 \\ \mu_0 \sim \mathcal{U}([0,2]),\ \mu_1 \sim \mathcal{U}([0,\pi]) \end{cases}$$

Without calibration: FAIL!

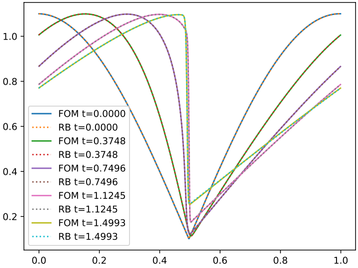
RB dim	failed
EIM dim	>600
FOM time	167 s
RB time	∞
RB/FOM time	∞

With calibration: Poly4

RB dim	19
EIM dim	41
FOM time	444 s
RB time	53 s
RB/FOM time	11%

Burgers: formation and motion of a shock

$$\begin{cases} u_t + \mu_0 (u^2/2)_x = 0,\ D = [0,\pi],\ T_{max} = 0.15, \text{ periodic BC} \\ u_0(x,\boldsymbol{\mu}) = |\sin(x+\mu_1)| + 0.1 \\ \mu_0 \sim \mathcal{U}([0,2]),\ \mu_1 \sim \mathcal{U}([0,\pi]) \end{cases}$$

Without calibration: FAIL!	With calibration: Poly4
	

Outline

Summary

High order space and time discretization

- Deferred Correction method
- Residual distribution
- Application on kinetic models
- IMEX scheme
- Asymptotic preserving

Model order reduction for hyperbolic problems

- POD EIM Greedy
- Troubles with advection
- ALE framework
- Calibration maps

Perspectives

High order space discretization

- Multiphase flows
- BGK models

Model order reduction for hyperbolic problems

- Systems
- Multidimensional geometries
- More complicated transformation maps
- Different regression maps

Thank you for the attention!

Residual Distribution

- High order
- FE based
- Compact stencil
- Explicit
- Can recast some other FV, FE, FD, DG schemes[2]

$$\partial_t U + \nabla \cdot F(U) = 0 \tag{9}$$

$$V_h = \{U \in L^2(\Omega_h, \mathbb{R}^D) \cap \mathcal{C}^0(\Omega_h),\, U|_K \in \mathbb{P}^k,\, \forall K \in \Omega_h\}. \tag{10}$$

$$U_h = \sum_{\sigma \in D_{\mathcal{N}}} U_\sigma \varphi_\sigma = \sum_{K \in \Omega_h} \sum_{\sigma \in K} U_\sigma \varphi_\sigma|_K \tag{11}$$

[2] **R. Abgrall. Computational Methods in Applied Mathematics; 2018**.

Residual Distribution

- High order
- FE based
- Compact stencil
- Explicit
- Can recast some other FV, FE, FD, DG schemes[2]

$$\partial_t U + \nabla \cdot F(U) = 0 \tag{9}$$

$$V_h = \{U \in L^2(\Omega_h, \mathbb{R}^D) \cap \mathcal{C}^0(\Omega_h),\, U|_K \in \mathbb{P}^k,\, \forall K \in \Omega_h\}. \tag{10}$$

$$U_h = \sum_{\sigma \in D_{\mathcal{N}}} U_\sigma \varphi_\sigma = \sum_{K \in \Omega_h} \sum_{\sigma \in K} U_\sigma \varphi_\sigma|_K \tag{11}$$

[2] **R. Abgrall. Computational Methods in Applied Mathematics; 2018**.

Residual Distribution - Spatial Discretization

1. Define $\forall K \in \Omega_h$ a fluctuation term (total residual) $\phi^K = \int_K \nabla \cdot F(U) dx$
2. Define a nodal residual $\phi_\sigma^K \; \forall \sigma \in K$:

$$\phi^K = \sum_{\sigma \in K} \phi_\sigma^K, \quad \forall K \in \Omega_h. \tag{12}$$

3. The resulting scheme is

$$U_\sigma^{n+1} - U_\sigma^n + \Delta t \sum_{K|\sigma \in K} \phi_\sigma^K = 0, \quad \forall \sigma \in D_{\mathcal{N}}. \tag{13}$$

Residual Distribution

- High order
- Easy to code
- FE based
- Compact stencil
- No need of Riemann solver
- No need of conservative variables
- Can recast some other FV, FE schemes

$$\partial_t U + \nabla \cdot A(U) = S(U) \tag{14}$$

$$V_h = \{U \in L^2(\Omega_h, \mathbb{R}^D) \cap \mathcal{C}^0(\Omega_h),\, U|_K \in \mathbb{P}^k,\, \forall K \in \Omega_h\}. \tag{15}$$

$$U_h = \sum_{\sigma \in D_\mathcal{N}} U_\sigma \varphi_\sigma = \sum_{K \in \Omega_h} \sum_{\sigma \in K} U_\sigma \varphi_\sigma|_K \tag{16}$$

Residual Distribution

- High order
- Easy to code
- FE based
- Compact stencil
- No need of Riemann solver
- No need of conservative variables
- Can recast some other FV, FE schemes

$$\partial_t U + \nabla \cdot A(U) = S(U) \tag{14}$$

$$V_h = \{U \in L^2(\Omega_h, \mathbb{R}^D) \cap \mathcal{C}^0(\Omega_h),\ U|_K \in \mathbb{P}^k,\ \forall K \in \Omega_h\}. \tag{15}$$

$$U_h = \sum_{\sigma \in D_{\mathcal{N}}} U_\sigma \varphi_\sigma = \sum_{K \in \Omega_h} \sum_{\sigma \in K} U_\sigma \varphi_\sigma|_K \tag{16}$$

Residual Distribution - Spatial Discretization

Focus on steady case.

1. Define $\forall K \in \Omega_h$ a fluctuation term (total residual) $\phi^K = \int_K \nabla \cdot A(U) - S(U) dx$
2. Define a nodal residual $\phi^K_\sigma \; \forall \sigma \in K$:

$$\phi^K = \sum_{\sigma \in K} \phi^K_\sigma, \quad \forall K \in \Omega_h. \tag{17}$$

Often done assigning $\phi^K_\sigma = \beta^K_\sigma \phi^K$, where must hold that

$$\sum_{\sigma \in K} \beta^K_\sigma = \mathrm{Id}. \tag{18}$$

3. The resulting scheme is

$$\sum_{K|\sigma \in K} \phi^K_\sigma = 0, \quad \forall \sigma \in D_{\mathcal{N}}. \tag{19}$$

This will be called residual distribution scheme.

Residual distribution - Choice of the scheme

How to split total residuals into nodal residuals ⇒ choice of the scheme.

$$\phi_\sigma^{K,LxF}(U_h) = \int_K \varphi_\sigma \left(\nabla \cdot A(U_h) - S(U_h)\right) dx + \alpha_K (U_\sigma - \overline{U}_h^K),$$

$$\overline{U}_h^K = \int_K U_h, \quad \alpha_K = \max_{e \text{ edge } \in K} \left(\rho_S \left(\nabla A(U_h) \cdot \mathbf{n}_e\right)\right),$$

$$\beta_\sigma^K(U_h) = \max\left(\frac{\Phi_\sigma^{K,LxF}}{\Phi^K}, 0\right) \left(\sum_{j \in K} \max\left(\frac{\Phi_j^{K,LxF}}{\Phi^K}, 0\right)\right)^{-1}, \tag{20}$$

$$\phi_\sigma^{*,K} = (1-\Theta)\beta_\sigma^K \phi_\sigma^K + \Theta \Phi_\sigma^{K,LxF}, \quad \Theta = \frac{|\Phi^K|}{\sum_{j\in K} |\Phi_j^{K,LxF}|},$$

$$\phi_\sigma^K = \beta_\sigma^K \phi_\sigma^{*,K} + \sum_{e | \text{edge of } K} \theta h_e^2 \int_e [\nabla U_h] \cdot [\nabla \varphi_\sigma] d\Gamma.$$

Error estimator

Additional hypothesis:

- $Id + \Delta t \mathcal{L}$ is Liptschitz continuous with constant $C > 0$,
- There are N'_{EIM} extra functions and functionals that capture the evolution of the solutions. (experimentally not so strict),
- Initial conditions are exactly represented in the reduced basis RB.

Total error estimator:

- EIM error, estimated by other N'_{EIM} basis functions f and functional τ iterating the EIM procedure after the stop, cost $\mathcal{O}(N'_{EIM})$,
- RB error given by the Lipschitz constant times residual of the small system,
- additionally one can add the projection error of the initial condition when not in RB.

Empirical interpolation method (EIM)

INPUT: $\mathcal{L}^n(U^n, \boldsymbol{\mu}, t^n)$, for $\boldsymbol{\mu} \in \mathcal{P}_h$, $n \leq N_t$

OUTPUT: $EIM = (\tau_k, f_k)_{k=1}^{N_{EIM}}$ where functions $f_k \in \mathbb{R}^{\mathcal{N}}$ and $\tau_k \in (\mathbb{R}^{\mathcal{N}})'$ (Examples of τ_k are point evaluations)

- Greedy iterative procedure
- At each step chooses the worst approximated function via an error estimator $\mathcal{L}^{worst} = \arg\max_{\mathcal{L}} ||\mathcal{L} - \sum_{k=1}^{N_{EIM}} \tau_k(\mathcal{L}) f_k||$
- Maximise the functional τ on the function $\mathcal{L}^{worst}$ $\tau^{chosen} = \arg\max_{\tau} |\tau(\mathcal{L}^{worst})|$
- $EIM = EIM \cup (\tau^{chosen}, \mathcal{L}^{worst})$
- Stop when error is smaller than a tolerance

Proper orthogonal decomposition (POD)

INPUT: Collection of functions $\{f_j\}_{j=1}^N$

OUTPUT: Reduced basis spaces
$RB = \underset{U|dim(U)=N_{POD}}{\arg\min} \sum_{j=1}^N ||f_j - \mathcal{P}_U(f_j)||_2$

- Based on SVD
- Prescribed tolerance to stop the algorithm
- Global optimizer of the problem

Greedy algorithm

INPUT: Collection of functions $\{f_j\}_{j=1}^N$

OUTPUT: Reduced basis space RB

- There is an error estimator (normally cheap) $\varepsilon_{RB}(f) \sim ||f - \mathcal{P}_{RB}(f)||$
- Iteratively choose the worst represented function $f^{worst} = \arg\max_f \varepsilon_{RB}(f)$
- Add f^{worst} to the RB space
- Stop up to a certain tolerance

DeC: Iterative process

K iterations where the iteration index is the superscript (k), with $k = 0, \dots, K$

1. Define $\mathbf{u}^{(0),m} = \mathbf{u}^n = \mathbf{u}(t^n)$ for $m = 0, \dots, M$
2. Define $\mathbf{u}^{(k),0} = \mathbf{u}(t^n)$ for $k = 0, \dots, K$
3. Find $\underline{\mathbf{u}}^{(k)}$ as $\mathcal{L}^1(\underline{\mathbf{u}}^{(k)}) = \mathcal{L}^1(\underline{\mathbf{u}}^{(k-1)}) - \mathcal{L}^2(\underline{\mathbf{u}}^{(k-1)})$
4. $\mathbf{u}^{n+1} = \mathbf{u}^{(K),M}$.

Theorem (Convergence DeC)

- *If $\mathcal{L}^1$ coercive with constant C_1*
- *If $\mathcal{L}^1 - \mathcal{L}^2$ Lipschitz with constant $C_2 \Delta t$*

Then $\|\underline{\mathbf{u}}^{(k)} - \underline{\mathbf{u}}^*\| \leq C \Delta t^k$

Hence, choosing $K = M + 1$, then $\|\mathbf{u}^{(K),M} - \mathbf{u}^{ex}(t^{n+1})\| \leq C \Delta t^K$

DeC – Proof

Proof.

Let $\underline{\mathbf{u}}^*$ be the solution of $\mathcal{L}^2(\underline{\mathbf{u}}^*)=0$. We know that $\mathcal{L}^1(\underline{\mathbf{u}}^*)=\mathcal{L}^1(\underline{\mathbf{u}}^*)-\mathcal{L}^2(\underline{\mathbf{u}}^*)$ and $\mathcal{L}^1(\underline{\mathbf{u}}^{(k+1)})=\left(\mathcal{L}^1(\underline{\mathbf{u}}^{(k)})-\mathcal{L}^2(\underline{\mathbf{u}}^{(k)})\right)$, so that

$$\begin{aligned} C_1||\underline{\mathbf{u}}^{(k+1)}-\underline{\mathbf{u}}^*|| \leq & ||\mathcal{L}^1(\underline{\mathbf{u}}^{(k+1)})-\mathcal{L}^1(\underline{\mathbf{u}}^*)|| = \\ = & ||\mathcal{L}^1(\underline{\mathbf{u}}^{(k)})-\mathcal{L}^2(\underline{\mathbf{u}}^{(k)})-(\mathcal{L}^1(\underline{\mathbf{u}}^*)-\mathcal{L}^2(\underline{\mathbf{u}}^*))|| \leq \\ \leq & C_2\Delta t||\underline{\mathbf{u}}^{(k)}-\underline{\mathbf{u}}^*||. \end{aligned}$$

$$||\underline{\mathbf{u}}^{(k+1)}-\underline{\mathbf{u}}^*|| \leq \left(\frac{C_2}{C_1}\Delta t\right)||\underline{\mathbf{u}}^{(k)}-\underline{\mathbf{u}}^*|| \leq \left(\frac{C_2}{C_1}\Delta t\right)^{k+1}||\underline{\mathbf{u}}^{(0)}-\underline{\mathbf{u}}^*||.$$

After K iteration we have an error at most of $\eta^K \cdot ||\underline{\mathbf{u}}^{(0)}-\underline{\mathbf{u}}^*||$. □