L'CADAME

High Performance Computing &
Machine Learning in Engineering

H.R. Anbarlooei, R. Arefidamghani

19/02/2025

Laboratório de Computação de Alto Desempenho e
Aprendizado de Máquina em Engenharia (L'CADAME)

UFRJ

Course

Overview

- Initial Value Problem (IVP)
- Boundary Value Problem (BVP)
- Finite Element Method
- Finite Volume Methos
- Eulerian, Lagrangian and ALE Frameworks
- Modeling Mixtures & Material Behavior
- Spectral Methods
- Direct Numerical Simulation
- Machine Learning

Initial Value Problem

Forward Euler

- Problem definition:

$$\frac{d}{dt}y(t) = f(y,t) \qquad\qquad \dot{y}(t) = \frac{y+t}{y-t}$$

- Taylor expansion:

$$y(t+\Delta t) = y(t) + \Delta t \left.\frac{dy}{dt}\right|_t + O(\Delta t^2)$$

- Define:

$$t \to t_n \qquad t+\Delta t \to t_{n+1}$$
$$y(t) \to y_n \qquad y(t+\Delta t) \to y_{n+1}$$

$$y(t+\Delta t) \approx y_{n+1} = y_n + \Delta t\, f(y_n, t_n)$$

Forward Euler Method

Initial Value Problem

Backward Euler

- Problem definition:

$$\frac{d}{dt}y(t) = f(y,t) \qquad\qquad \dot{y}(t) = \frac{y+t}{y-t}$$

- Taylor expansion:

$$y(t) = y(t+\Delta t) - \Delta t\,\left.\frac{dy}{dt}\right|_{t+\Delta t} + O(\Delta t^2)$$

- Approximation:

$$y(t+\Delta t) \approx y_n = y_{n+1} + \Delta t\, f(y_{n+1}, t_{n+1})$$

Backward Euler Method

Initial Value Problem

Example

- Toy problem:

$$\dot{y}(t) = \frac{y+t}{y-t} \qquad\qquad y(0) = 1$$

- Forward Euler:

$$y_{n+1} = y_n + \Delta t\, f(y_n, t_n)$$

- Solution ($\Delta t = 0.1$):

$$y_0 = 1,$$

$$y(0.1) \approx y_1 = y_0 + \Delta t \times \frac{y_0 + t_0}{y_0 - t_0} = 1 + 0.1 \times \frac{1+0}{1-0} = 1.1$$

$$y(0.2) \approx y_2 = y_1 + \Delta t \times \frac{y_1 + t_1}{y_1 - t_1} = 1.1 + 0.1 \times \frac{1.1+0.1}{1.1-0.1} = 1.22$$

Initial Value Problem

Example

- Toy problem:

$$\dot{y}(t) = \frac{y+t}{y-t} \qquad\qquad y(0) = 1$$

- Backward Euler:

$$y_{n+1} = y_n + \Delta t\, f(y_{n+1}, t_{n+1})$$

- Solution ($\Delta t = 0.1$):

$$y_0 = 1,$$

$$y(0.1) \approx y_1 = y_0 + \Delta t \times \frac{y_1 + t_1}{y_1 - t_1} = 1 + 0.1 \times \frac{y_1 + 0}{y_1 - 0}$$

$$y(0.2) \approx y_2 = y_1 + \Delta t \times \frac{y_2 + t_2}{y_2 - t_2}$$

Initial Value Problem

Error and Order

- Forward Euler (Explicit method):

$$y_{n+1} = y_n + \Delta t\, f(y_n, t_n)$$

- Backward Euler (Implicit Method):

$$y_{n+1} = y_n + \Delta t\, f(y_{n+1}, t_{n+1})$$

- Error:

$$y(t_{n+1}) \neq y_{n+1} \rightarrow e_n = |y(t_n) - y_n|$$

- Order of Method (order *p*):

$$e = O(\Delta t^p)$$

Forward and Backward Euler methods are first order.

- Discrete Solution (finite number of points)

Initial Value Problem

Higher Order Methods

- Forward & Backward Euler:

$$y_{n+1} = y_n + \Delta t\, f(y_n, t_n)$$

$$y_{n+1} = y_n + \Delta t\, f(y_{n+1}, t_{n+1})$$

- Crank–Nicolson Method:

$$y_{n+1} = y_n + \frac{1}{2}\Delta t\big(f(y_n, t_n) + f(y_{n+1}, t_{n+1})\big)$$

- Runge–Kutta Method (RK4):

$$y_{n+1} = y_n + \frac{1}{6}\Delta t(f_1 + 2f_2 + 2f_3 + f_4)$$

$$f_2 = f\left(t + \frac{\Delta t}{2}, y_n + \frac{\Delta t}{2} f_1\right) \qquad f_3 = f\left(t + \frac{\Delta t}{2}, y_n + \frac{\Delta t}{2} f_2\right)$$

Initial Value Problem

Numerical Stability

- Forward Euler:

$$y_{n+1} = y_n + \Delta t\, f(y_n, t_n)$$

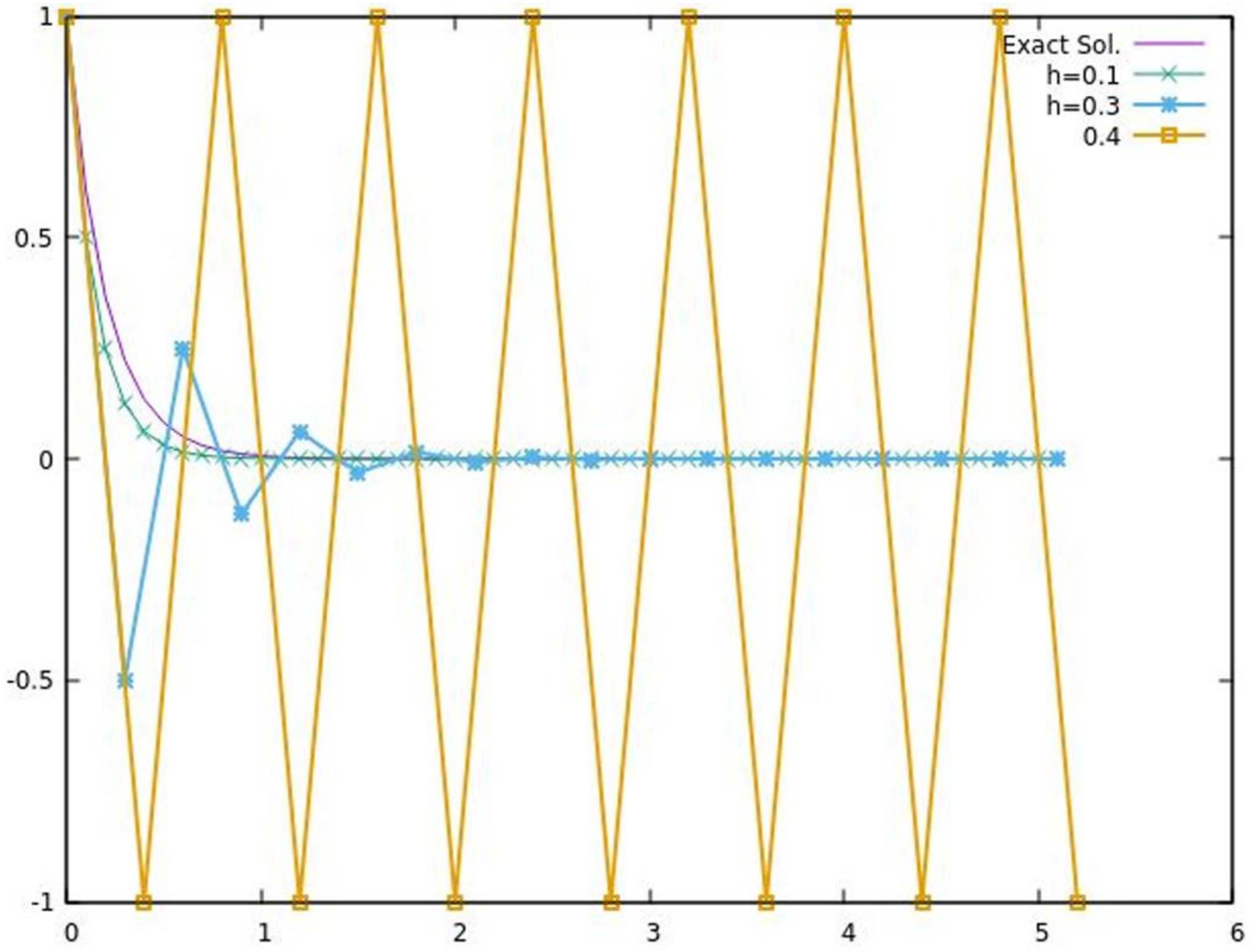
- Toy problem:

$$\frac{dy}{dt} = -5y$$
$$y(0) = 1$$

- Exact Solution:

$$y(t) = e^{-5t}$$

- Region of Stability.



Boundary Value Problem

Introduction

- Model Problem:

$$\frac{d^2u}{dx^2} = f\left(x, u, \frac{du}{dx}\right)$$

- Boundary condition:

$$u(a) = u_a \quad ; \quad u(b) = u_b$$ Dirichlet

$$\left.\frac{du}{dx}\right|_a = \dot{u}_a \quad ; \left.\frac{du}{dx}\right|_b = \dot{u}_b$$ Von Neumann

$$\left.\frac{du}{dx}\right|_a = \alpha_1 u(a) + \beta_1 \quad ; \quad \left.\frac{du}{dx}\right|_b = \alpha_2 u(b) + \beta_2$$ Robin

Boundary Value Problem

Solution Method

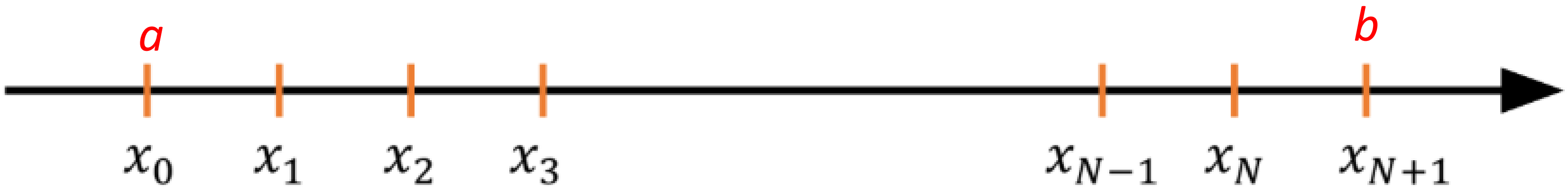
- Model Problem:

$$\frac{d^2u}{dx^2} = f\left(x, u, \frac{du}{dx}\right) \qquad u(a) = u_a \quad ; \quad u(b) = u_b$$

- Approximation (Central Difference):

$$\frac{d^2u}{dx^2} \approx \frac{u(x+\Delta x) - 2u(x) + u(x-\Delta x)}{\Delta x^2}$$

$$u_{i+1} - 2u_i + u_{i-1} = \Delta x^2\, f\left(x_i, u_i, \left.\frac{du}{dx}\right|_i\right)$$

- Computational Grid
- 

$$x_0 = a \to u(a) = u_0 = u_a$$

$$x_{N+1} = b \to u(b) = u_{N+1} = u_b$$

$$\Delta x = \frac{b-a}{N+1}$$

Boundary Value Problem

Solution Method

- Model Problem:

$$\frac{d^2u}{dx^2} = f\left(x, u, \frac{du}{dx}\right) \qquad u(a) = u_a \quad ; \quad u(b) = u_b$$

- Approximation (Central Difference):

$$\frac{d^2u}{dx^2} \approx \frac{u(x+\Delta x) - 2u(x) + u(x-\Delta x)}{\Delta x^2}$$

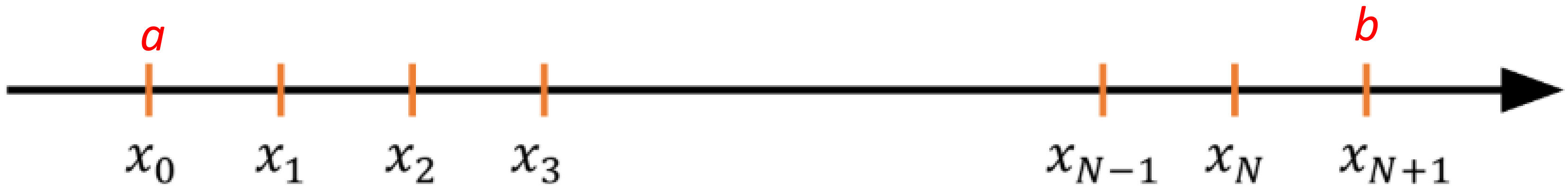
$$u_{i+1} - 2u_i + u_{i-1} = \Delta x^2 \, f\left(x_i, u_i, \left.\frac{du}{dx}\right|_i\right)$$

$$x_0 = a \rightarrow u(a) = u_0 = u_a$$

$$x_{N+1} = b \rightarrow u(b) = u_{N+1} = u_b$$

$$\Delta x = \frac{b-a}{N+1}$$

- Computational Grid

- 

Boundary Value Problem

Solution Method

$$u_{i+1} - 2u_i + u_{i-1} = \Delta x^2 \, f\left(x_i, u_i, \left.\frac{du}{dx}\right|_i\right)$$

- Solution:

$$u_2 - 2u_1 + u_0 = \Delta x^2 f_1$$

$$u_3 - 2u_2 + u_1 = \Delta x^2 f_2$$

$$\vdots$$

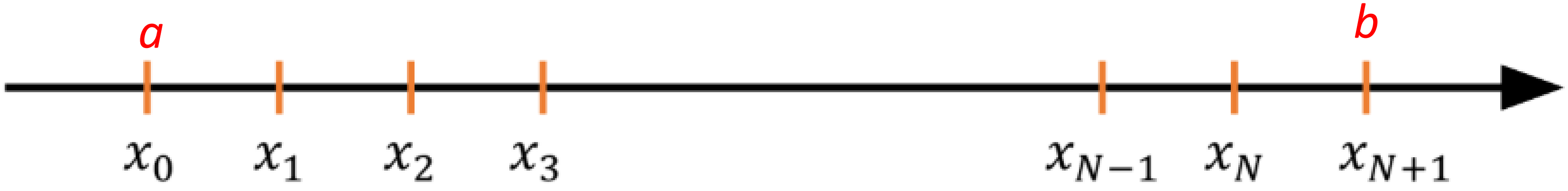
$$u_{N+1} - 2u_N + u_{N-1} = \Delta x^2 f_N$$

$$AU = B$$

$$U = [u_1, u_2, \dots, u_N]^T$$

$$A = \begin{bmatrix} -2 & 1 & 0 & \dots & 0 & 0 \\ 1 & -2 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \dots & \dots & \dots & \dots & 1 & -2 \end{bmatrix}$$

$$B = [L_1 - u_0, L_2, \dots, L_{N-1}, L_N - u_{N+1}]^T$$



Boundary Value Problem

2D Problems

$$\frac{d^2u}{dx^2} + \frac{d^2u}{dy^2} = f(x,y)$$

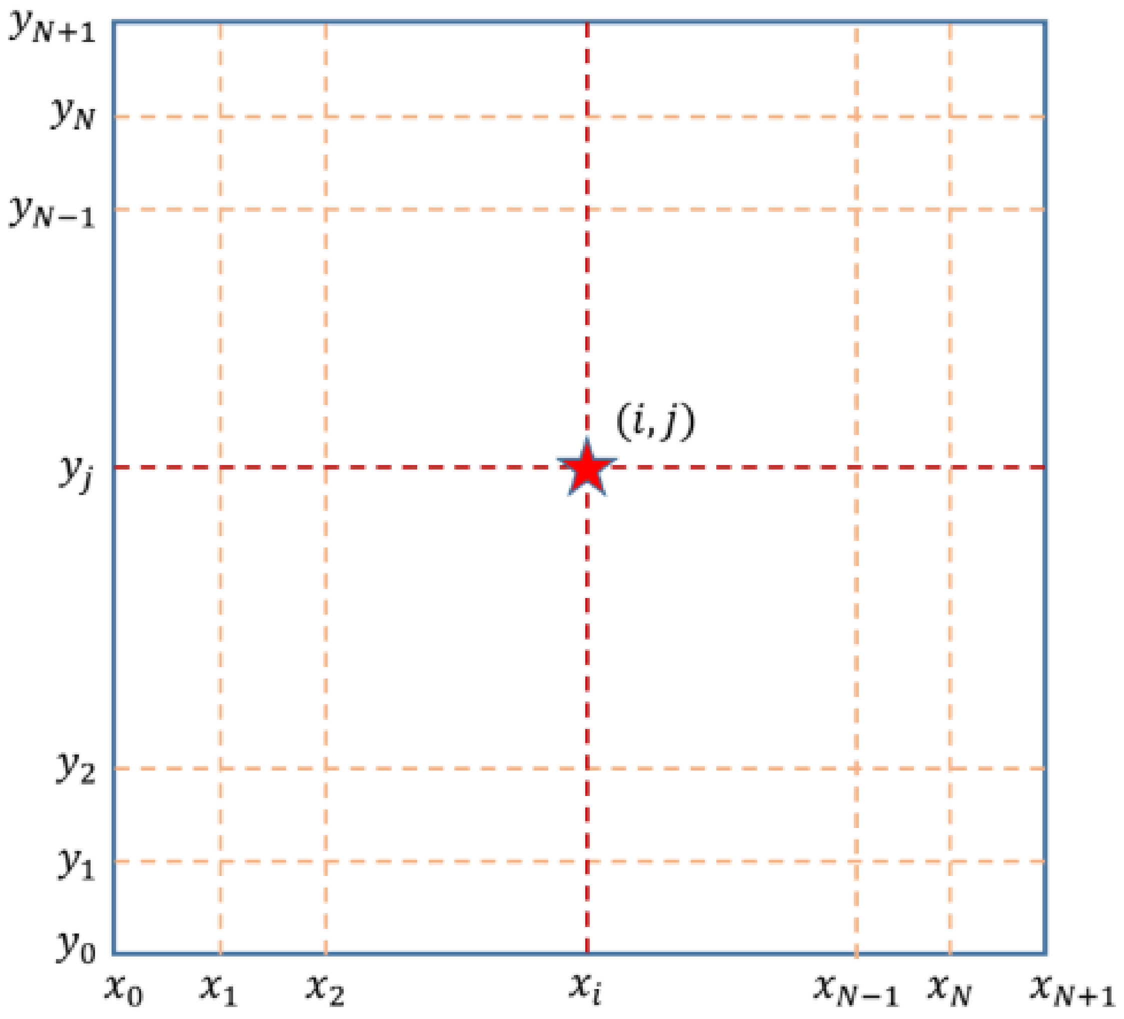
$$u(a,y) = u_a \quad ; \quad u(b,y) = u_b$$

$$u(x,c) = u_c \quad ; \quad u(x,d) = u_d$$

- Solution:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2}$$

$$+ \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} = f(x_i, y_j)$$



Finite Element Method

Introduction

- Toy problem:

$$-\frac{d^2u}{dx^2} = f(x) \qquad 0 \leq x \leq 1 \qquad u(0) = u(1) = 0$$

- FE Approximation:

$$u(x) \approx u_h(x) = \sum_{j=0}^{N} c_j \varphi_j(x)$$

- $\varphi_j(x)$ is the base function, which we know beforehand (decision).
- The problem is to find c_j which represents the solution best.

Finite Element Method

Introduction

- Toy problem:

$$-\frac{d^2u}{dx^2} = f(x) \qquad 0 \le x \le 1 \qquad u(0) = u(1) = 0$$

- FE Approximation:

$$u(x) \approx u_h(x) = \sum_{i=0}^{N} c_i \varphi_i(x)$$

- Let's assume that $\varphi_j(x = 0) = 0$ and $\varphi_j(x = 1) = 0$.
- And define residual as:

$$r(x) = \frac{d^2u_h}{dx^2} + f(x) = \sum_{i=0}^{N} c_i \frac{d^2}{dx^2}\varphi_i(x) + f(x) \neq 0$$

Finite Element Method

Introduction

- Approximation

$$r(x) = \frac{d^2 u_h}{dx^2} + f(x) = \sum_{i=0}^{N} c_i \frac{d^2}{dx^2} \varphi_i(x) + f(x) \neq 0$$

- Two paths:

1. Find c_j in a way the $r(x)$ be zero at all grid points (collocation method).
2. Find c_j in a way that

$$\int_0^1 r(x) \varphi_j(x)\, dx = 0 \qquad\qquad j = 1..N$$

Galerkin Method

Finite Element Method

Galerkin Method

- Approximation

$$r(x) = \sum_{i=0}^{N} c_i \frac{d^2}{dx^2}\varphi_j(x) + f(x) \neq 0$$

$$\int_0^1 r(x)\varphi_j(x)\,dx = 0 \qquad\qquad j = 1..N$$

$$\int_0^1 \left[\sum_{i=0}^{N} c_i \frac{d^2}{dx^2}\varphi_i(x) + f(x)\right]\varphi_j(x)\,dx = 0$$

$$\sum_{i=0}^{N} c_i \int_0^1 \left[\frac{d^2}{dx^2}\varphi_i(x)\varphi_j(x)\right] dx + \int_0^1 f(x)\varphi_j(x)dx = 0$$

$$c_1 a_{11} + c_2 a_{12} + \cdots + c_N a_{1N} = b_1$$

$$A\vec{c} = \vec{b}$$

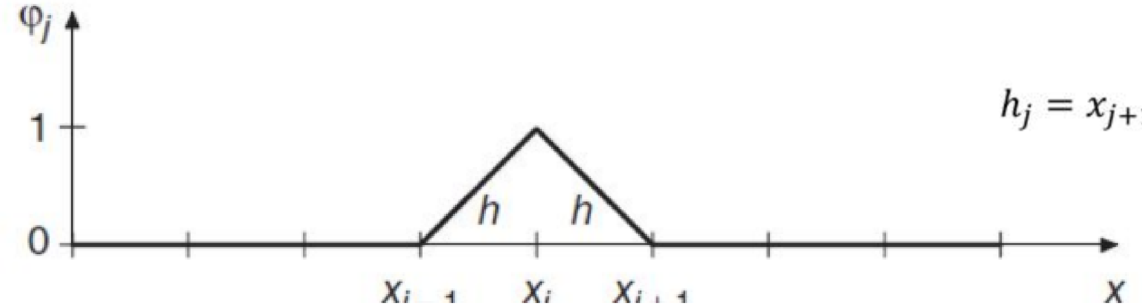
Finite Element Method

Summary

- Continuous solution.
- Different bases (kernels), different accuracies.

$$\phi_j(x) = x^j(x-1)$$

$$\phi_j(x) = \begin{cases} \dfrac{x - x_{j-1}}{x_j - x_{j-1}}, & x_{j-1} \le x \le x_j \\ \dfrac{x_{j+1} - x}{x_{j+1} - x_j}, & x_j \le x \le x_{j+1} \\ 0, & otherwise \end{cases}$$



$$\phi_j(x) = \begin{cases} g_1(\xi_{j-1}(x)), & x_{j-1} \le x \le x_j \\ g_1(1-\xi_j(x)), & x_j \le x \le x_{j+1} \\ 0, & otherwise \end{cases}$$

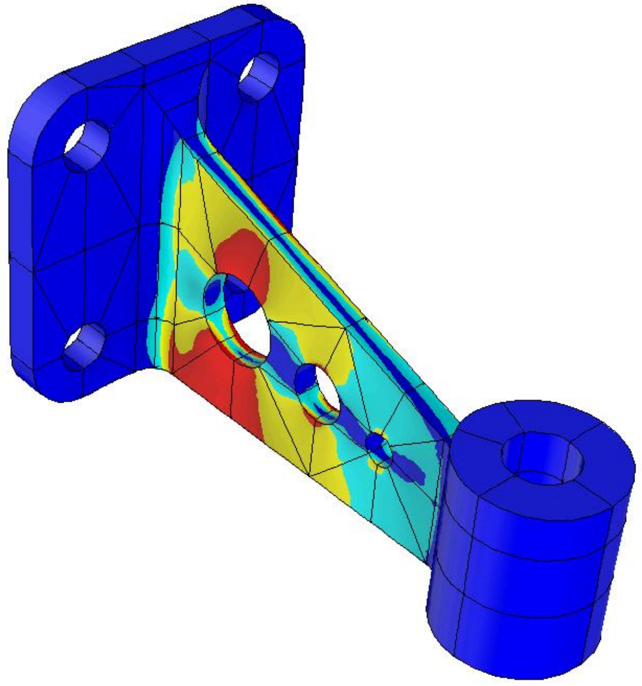
$$\frac{d\phi_j}{dx} = \begin{cases} h_{j-1} g_2(\xi_{j-1}(x)), & x_{j-1} \le x \le x_j \\ -h_j g_2(1-\xi_j(x)), & x_j \le x \le x_{j+1} \\ 0, & otherwise \end{cases}$$

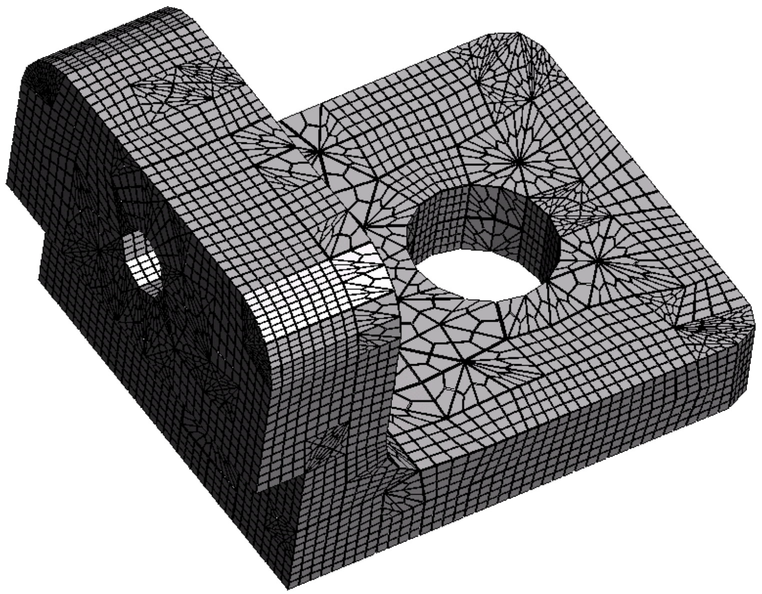
$$h_j = x_{j+1} - x_j, \qquad \xi_j(x) = \frac{x - x_j}{h_j}, \qquad g_1(x) = -2x^3 + 3x^2, \qquad g_2(x) = x^3 - x^2$$

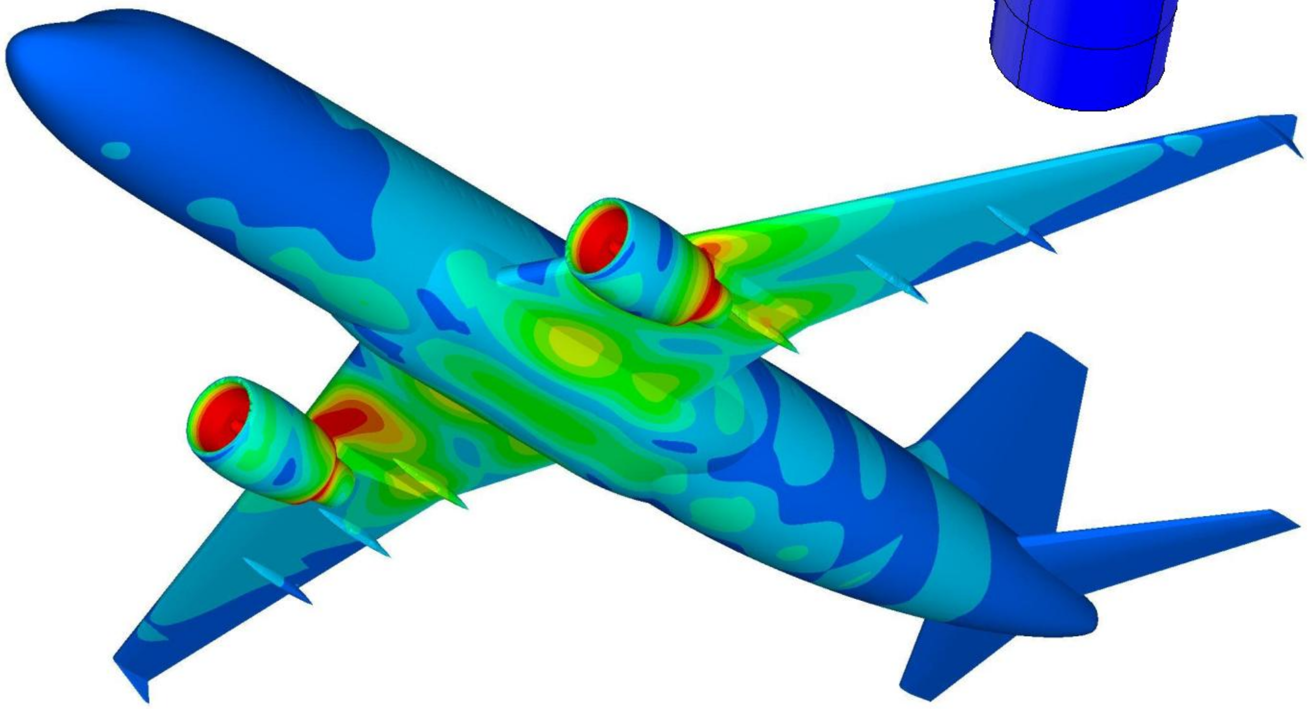
Finite Element Method

Summary

- Complex geometries,
- Different types of computational grid.







Finite Volume Method

Introduction

- Differential equations break down in the presence of discontinuities.
  - Material borders, Shock-waves, Constitutive equations.
- Solving in the integral form can address this issue effectively.

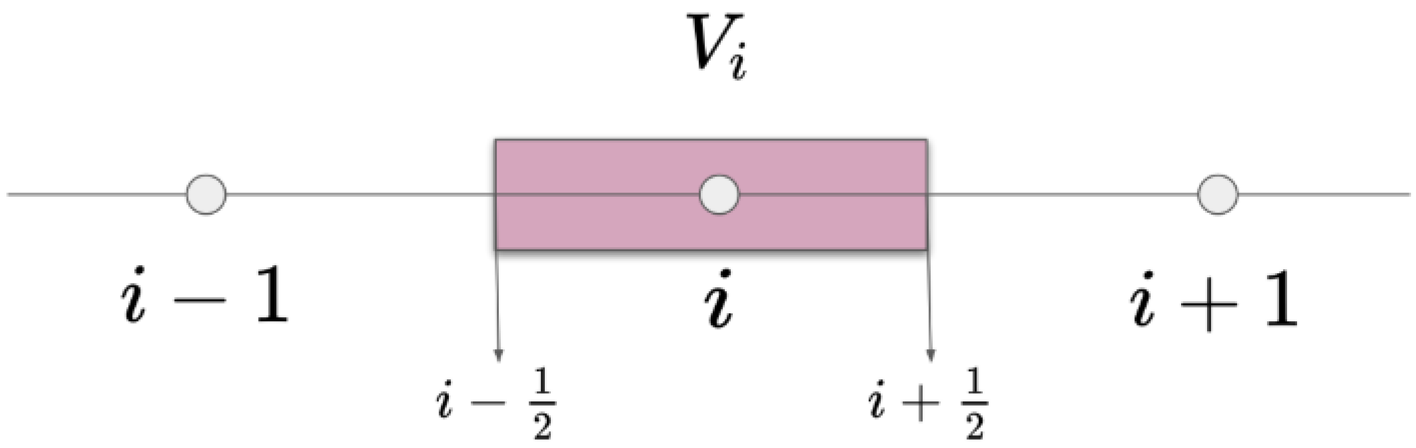
$$\frac{\partial \rho}{\partial t} + \nabla.(\rho u) = 0$$

$$\int_{\Omega}\left[\frac{\partial \rho}{\partial t} + \nabla.(\rho u)\right] dV = \int_{\Omega}\frac{\partial \rho}{\partial t} dV + \int_{\partial\Omega}\rho(n.u)\, dS = 0$$

Finite Volume Method

Introduction

- Computational grid in FV method:



- We will track the average value of a quantity in each cell.

$$\overline{u_i}(t_n) = \frac{1}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x, t_n)\,dx$$

Finite Volume Method

Introduction

- Toy problem (conservation):

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0$$

$$\int_{V_i} \frac{\partial u}{\partial t}\, dV + \left[f\left(u_{i+\frac{1}{2}}\right) - f\left(u_{i+\frac{1}{2}}\right) \right] = 0$$

$$\frac{\partial}{\partial t} \int_{V_i} u\, dV + \left[f\left(u_{i+\frac{1}{2}}\right) - f\left(u_{i+\frac{1}{2}}\right) \right] = 0$$

$$\Delta x \left(\frac{\partial \bar{u}_i}{\partial t} \right) + \left[f\left(u_{i+\frac{1}{2}}\right) - f\left(u_{i+\frac{1}{2}}\right) \right] = 0$$

$$\frac{\bar{u}_i^{n+1} - \bar{u}_i^{n}}{\Delta t} + \frac{f\left(u_{i+\frac{1}{2}}\right) - f\left(u_{i+\frac{1}{2}}\right)}{\Delta x} = 0$$

Classification

Second order equation

- Toy problem (conservation):

$$a\frac{\partial^2 u}{\partial x^2} + 2b\frac{\partial^2 u}{\partial x \partial y} + c\frac{\partial^2 u}{\partial y^2} + d\frac{\partial u}{\partial x} + e\frac{\partial u}{\partial y} + fu = g(x,y)$$

$b^2 - ac < 0$ Elliptic Equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

$b^2 - ac = 0$ Parabolic Equation

$$\frac{\partial u}{\partial t} = \kappa\frac{\partial^2 u}{\partial x^2}$$

$b^2 - ac > 0$ Hyperbolic Equation

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial t} = 0$$

Finite Volume Method

Introduction

- Toy problem (conservation):

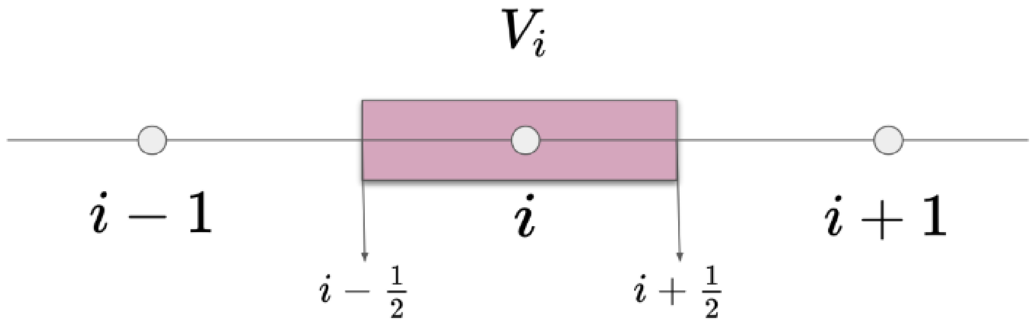
$$\frac{\bar{u}_i^{n+1} - \bar{u}_i^n}{\Delta t} + \frac{f\left(u_{i+\frac{1}{2}}\right) - f\left(u_{i+\frac{1}{2}}\right)}{\Delta x} = 0$$

- Naïve approximation:

$$u_{i+\frac{1}{2}} = \frac{1}{2}\left(\bar{u}_i + \bar{u}_{i+1}\right)$$

- Upwinding:

$$u_{i-\frac{1}{2}} = \bar{u}_{i-1} \qquad\qquad u_{i+\frac{1}{2}} = \bar{u}_i$$



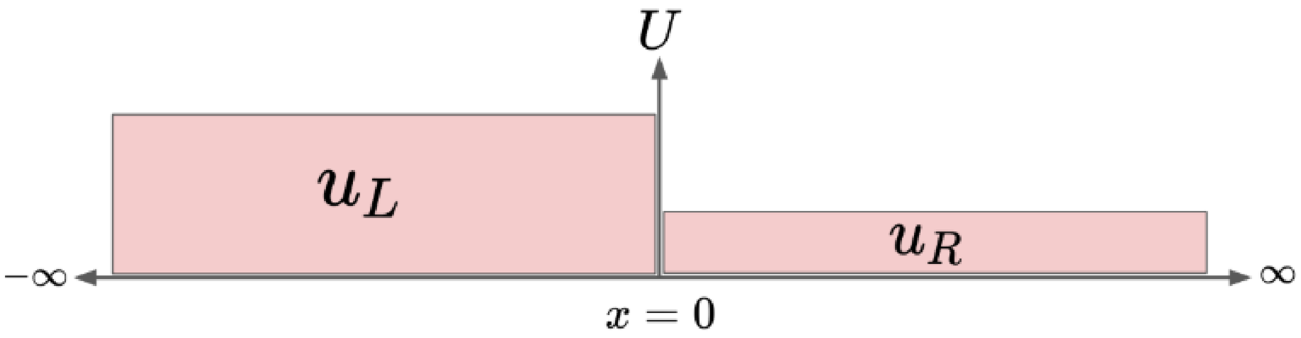
Riemann Problem

Introduction

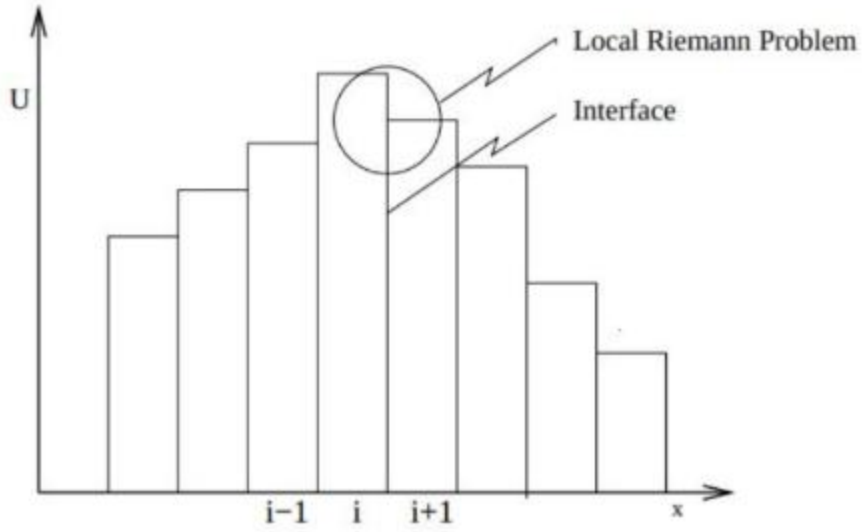
- Toy problem:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = 0$$

$$IC: \begin{cases} u(x < 0, t = 0) = u_L \\ u(x \geq 0, t = 0) = u_R \end{cases}$$



- Godunov scheme
  - Exact solution
  - Approximate solution (HLL,HLLE,HLLC,…)



Gas dynamics Equations

Euler Equations

- Euler equation (gas dynamics)

$$\frac{\partial \rho}{\partial t} + \nabla.(\rho \vec{u}) = 0$$

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla.(\rho \vec{u} \otimes \vec{u}) = \nabla.\sigma$$

$$\frac{\partial \rho e}{\partial t} + \nabla.(\rho e \vec{u}) = \sigma : \dot{\epsilon}$$

- Conservation (mass, momentum and energy),
- Compressible fluid,
- Hyperbolic equation.

Gas dynamics Equations

Navier-Stokes Equation

- Euler equation (gas dynamics)

$$\frac{\partial \rho}{\partial t} + \nabla.(\rho\vec{u}) = 0 \qquad \frac{\partial \rho\vec{u}}{\partial t} + \nabla.(\rho\vec{u}\otimes\vec{u}) = \nabla.\sigma \qquad \frac{\partial \rho e}{\partial t} + \nabla.(\rho e\vec{u}) = \sigma:\dot{\epsilon}$$

- Navier-Stokes equation

$$\nabla.(\vec{u}) = 0$$

$$\frac{\partial \vec{u}}{\partial t} + \nabla.(\vec{u}\otimes\vec{u}) = \frac{1}{\rho}\nabla.\sigma$$

- Incompressible fluid,
- Parabolic equation (elliptic in limit of steady state flow).

$$\sigma = -pI + \tau \quad \rightarrow \quad \frac{\partial \vec{u}}{\partial t} + \nabla.(\vec{u}\otimes\vec{u}) = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla.\tau$$

Gas dynamics Equations

Navier-Stokes Equation

- Navier-Stokes equation

$$\nabla.(\vec{u}) = 0$$

$$\frac{\partial \vec{u}}{\partial t} + \nabla.(\vec{u} \otimes \vec{u}) = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla.\tau$$

- Constitutive Equation (describes the fluid = property of fluid),
- Newtonian fluid

$$\tau = \mu[\nabla u + \nabla u^T] = 2\mu S$$

- Rate-of-strain tensor, $S = \frac{1}{2}[\nabla u + \nabla u^T]$, deformation rate $\dot{\gamma} = \sqrt{2S{:}\,S}$.

$$\frac{\partial \vec{u}}{\partial t} + \nabla.(\vec{u} \otimes \vec{u}) = -\frac{1}{\rho}\nabla p + \nu\,\nabla^2 u$$

Gas dynamics Equations

Navier-Stokes Equation

- Newtonian fluid:

$$\tau = \mu[\nabla u + \nabla u^T] = 2\mu S$$

- Generalized Newtonian Fluid (GNF):

$$\tau = 2\mu(\dot{\gamma})S$$

- Power-law type fluid: $\tau(\dot{\gamma}) = K\dot{\gamma}^n$
- Bingham Fluid:

$$\begin{cases} \tau = \tau_y + \mu\dot{\gamma} & if\ \tau > \tau_y \\ \dot{\gamma} = 0 & if\ \tau < \tau_y \end{cases}$$

- Herschel-Bulkley Fluid:

$$\begin{cases} \tau = \tau_y + K\dot{\gamma}^n & if\ \tau > \tau_y \\ \dot{\gamma} = 0 & if\ \tau < \tau_y \end{cases}$$

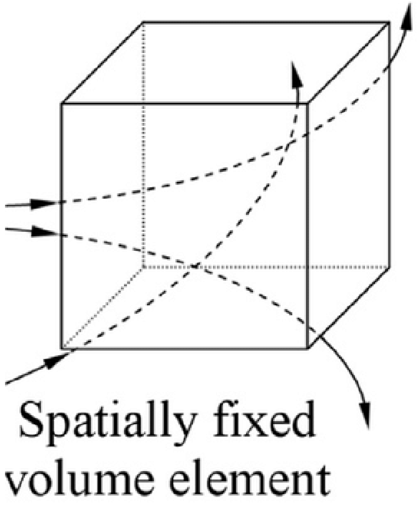
Modeling

$$\tau = \frac{\tau_y + K\dot{\gamma}^n}{\dot{\gamma}} S$$

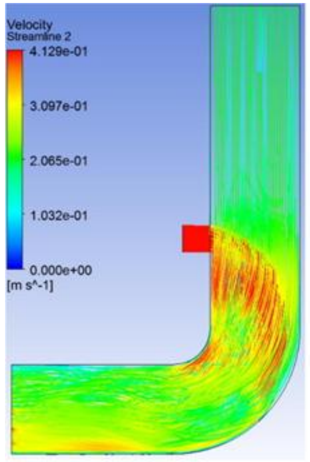
Lagrangian vs Eulerian

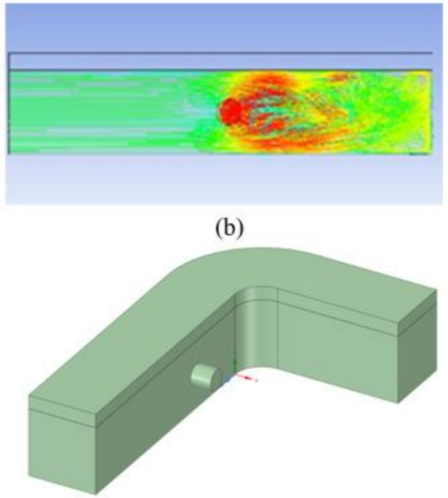
Introduction

Eulerian

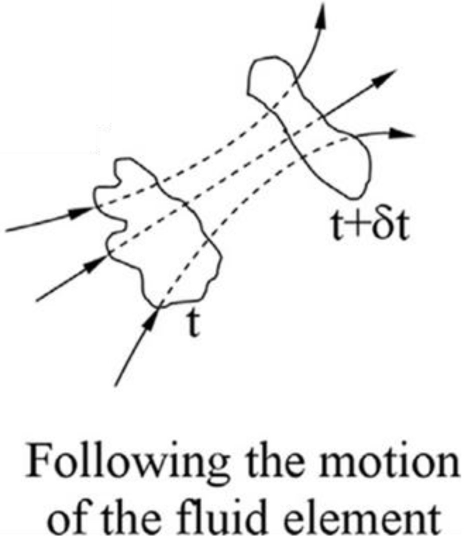


$$\frac{\partial \boxplus}{\partial t} + \nabla.(\boxplus\, \vec{u})$$

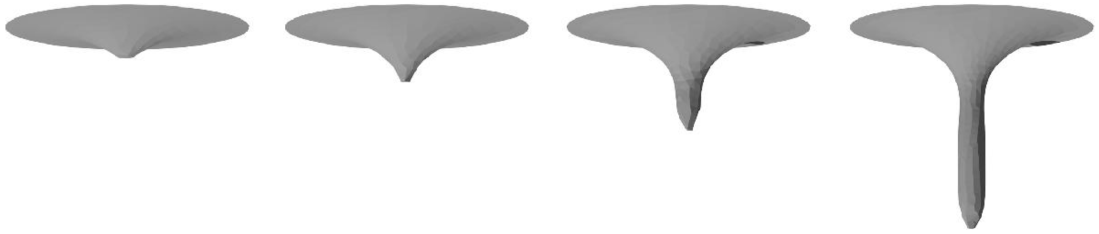




Lagrangian

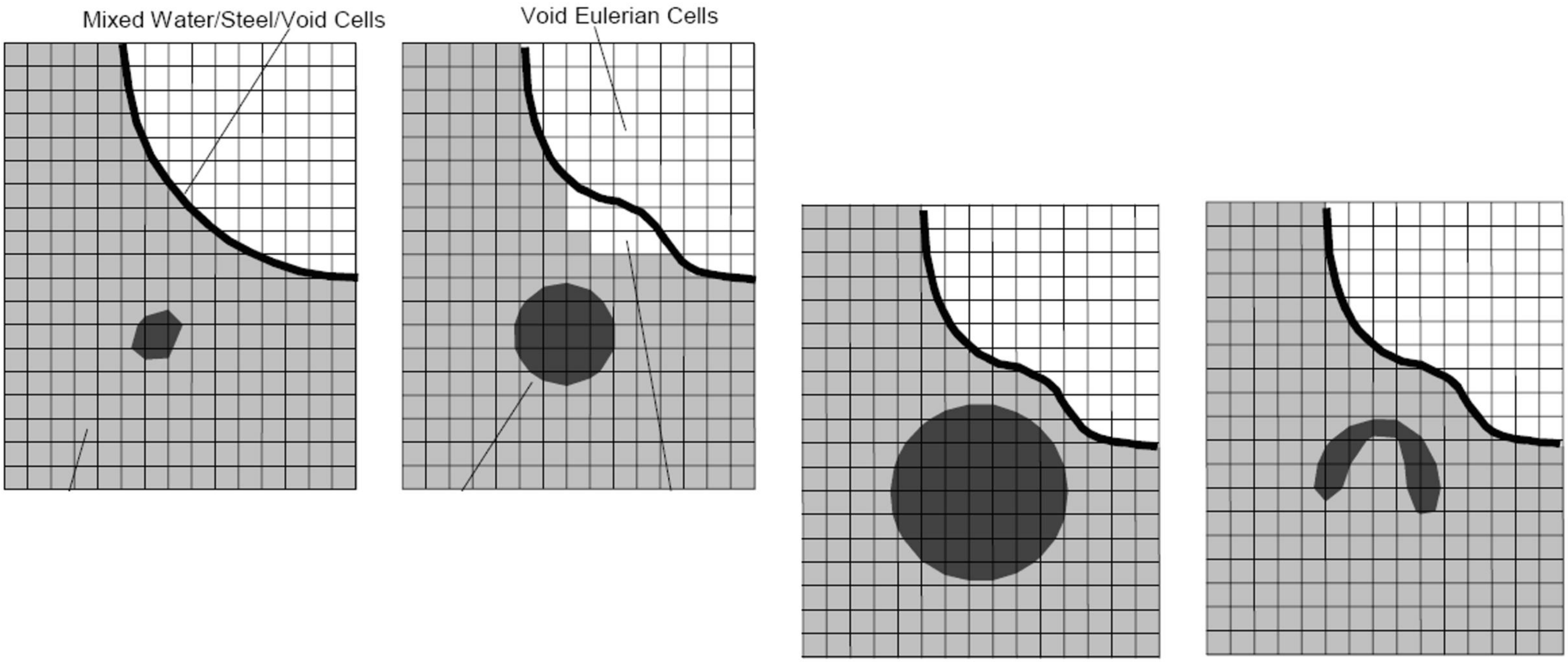


$$\frac{D \boxplus}{D t}$$



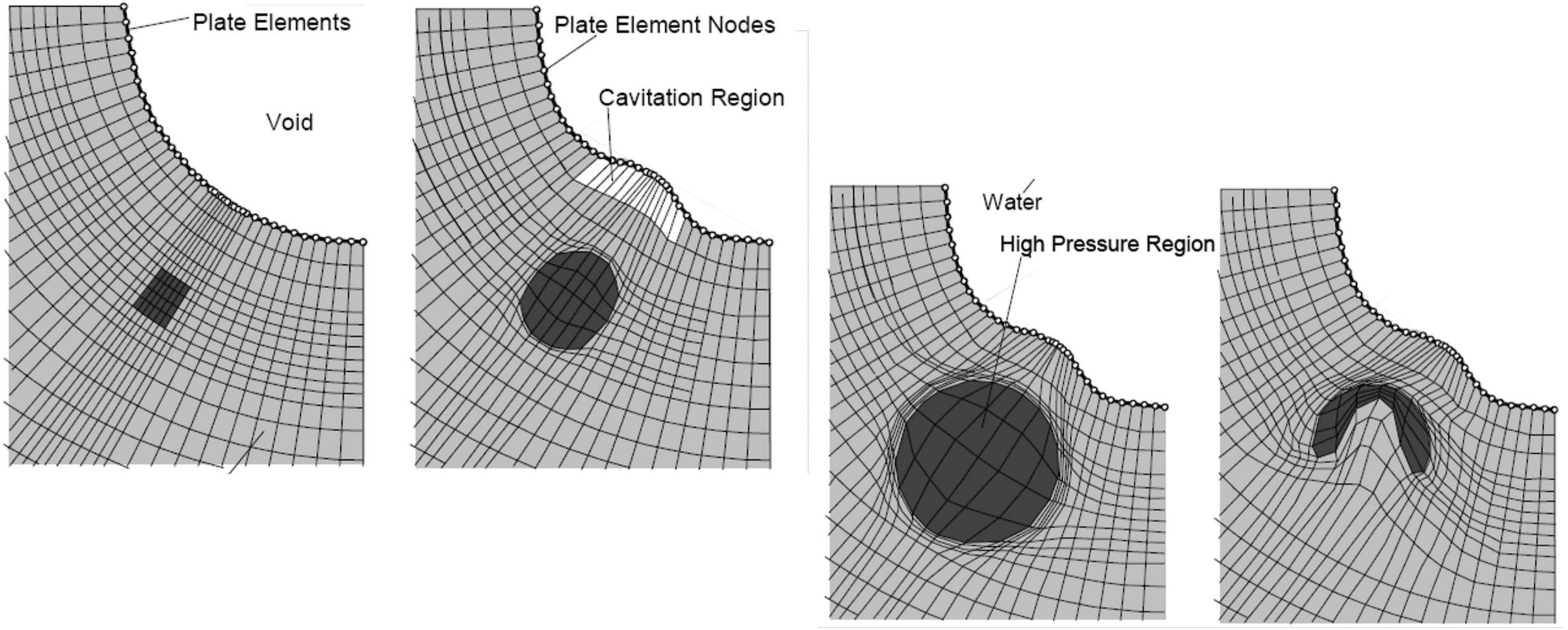
Framework

Eulerian view



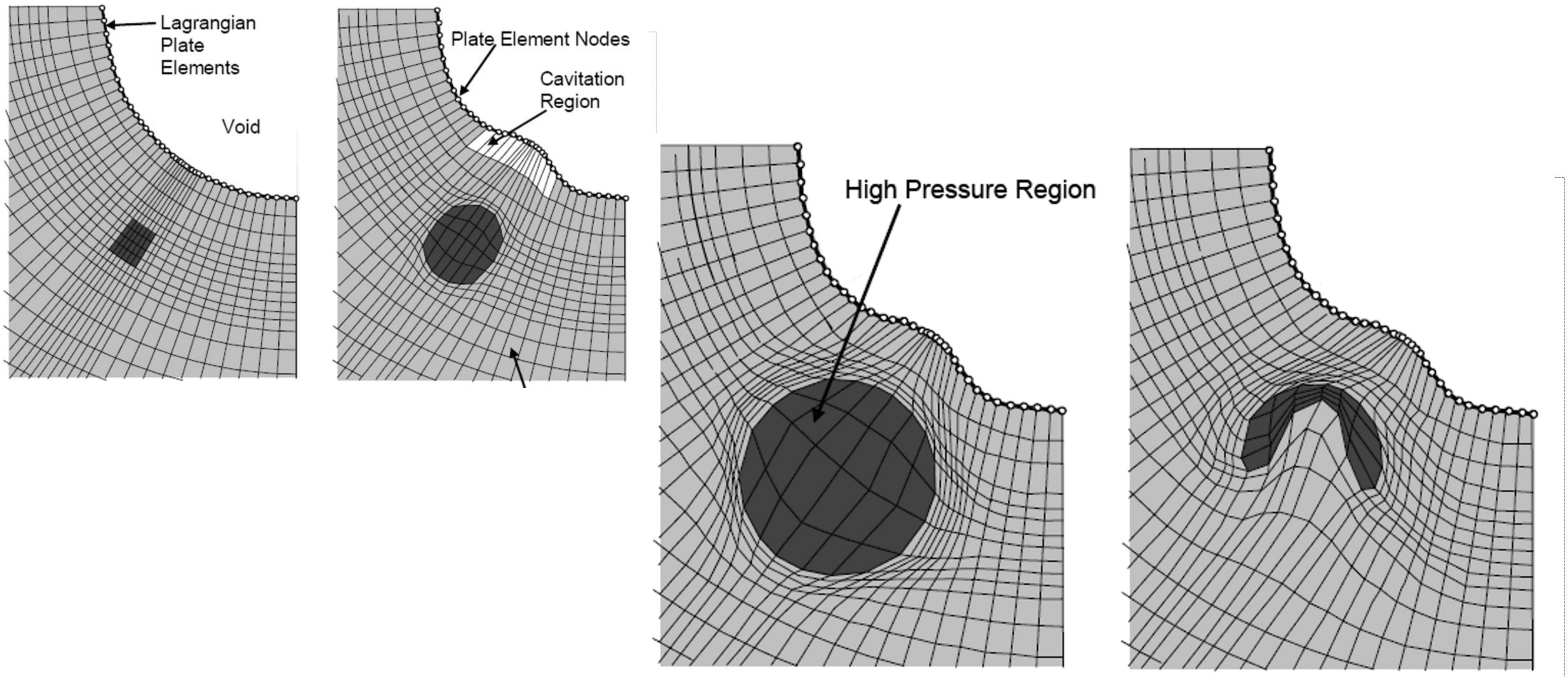
Framework

Lagrangian view



Framework

ALE view



Arbitrary Lagrangian-Eulerian

Formulation

Lagrangian	Eulerian	ALE
$\frac{1}{V}\frac{DV}{Dt} = \nabla.\vec{u}$ $\rho\frac{D\vec{u}}{Dt} = -\nabla p$ $\rho\frac{De}{Dt} = -p\,\nabla.\vec{u}$	$\frac{\partial\rho}{\partial t} + \nabla.(\rho\vec{u}) = 0$ $\frac{\partial\rho\vec{u}}{\partial t} + \nabla.(\rho\vec{u}\otimes\vec{u}) = -\nabla p$ $\frac{\partial\rho e}{\partial t} + \nabla.(\rho e\vec{u}) = -p\nabla.\vec{u}$	$\frac{\partial\rho}{\partial t} + \nabla.\left(\rho(\vec{u} - \vec{u}_g)\right) = 0$ $\frac{\partial\rho\vec{u}}{\partial t} + \nabla.\left(\rho\vec{u}\otimes(\vec{u} - \vec{u}_g)\right) = -\nabla p$ $\frac{\partial\rho e}{\partial t} + \nabla.\left(\rho e(\vec{u} - \vec{u}_g)\right)$ $= -p\nabla.\,\vec{u}$

An extra degree of freedom: $\vec{u}_g$

Arbitrary Lagrangian-Eulerian

Solution Strategy

- Functional Splitting method

$$\frac{\partial \rho\phi}{\partial t} + \nabla.\left(\rho\phi(\vec{u} - \vec{u}_g)\right) = S_\phi$$

Lagrangian

$$\frac{D\rho\phi}{Dt} = S_P$$

Advection

$$\frac{\partial \rho\phi}{\partial t} + \nabla.\left(\rho\phi(\vec{u} - \vec{u}_g)\right) = 0$$

ALE

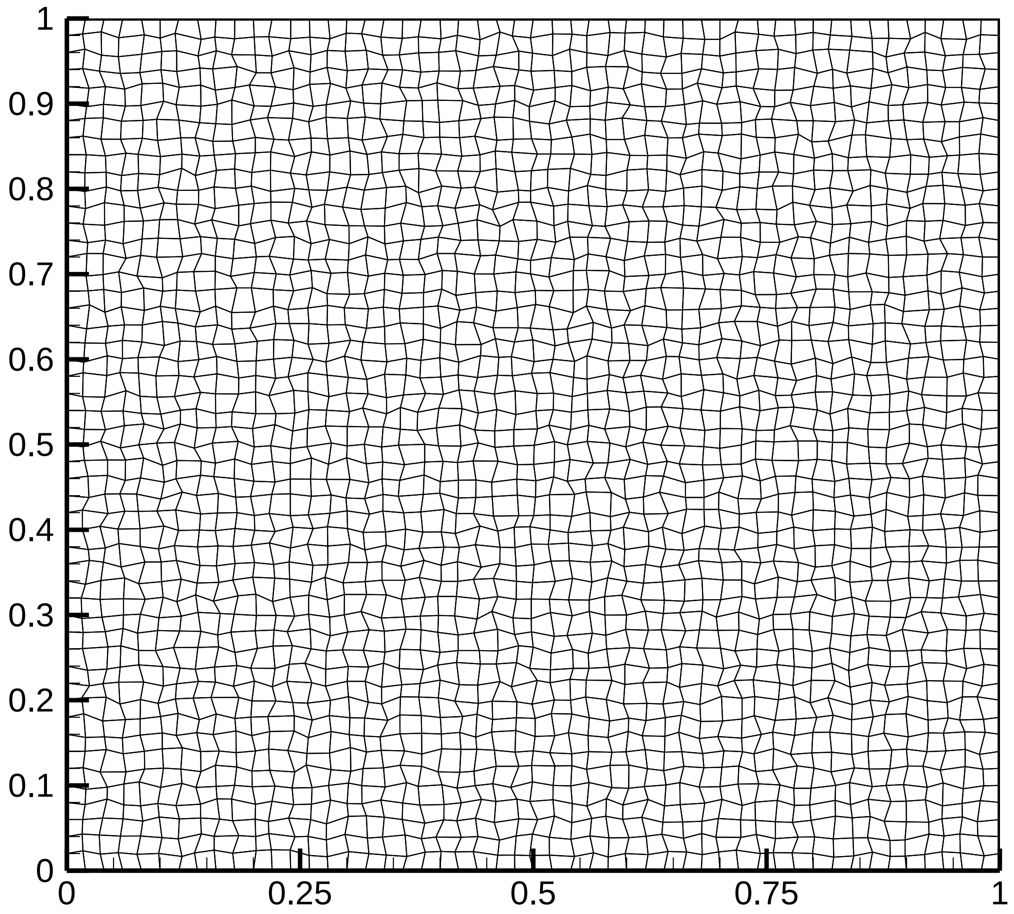
$$\frac{\partial \rho}{\partial t} + \nabla.\left(\rho(\vec{u} - \vec{u}_g)\right) = 0$$

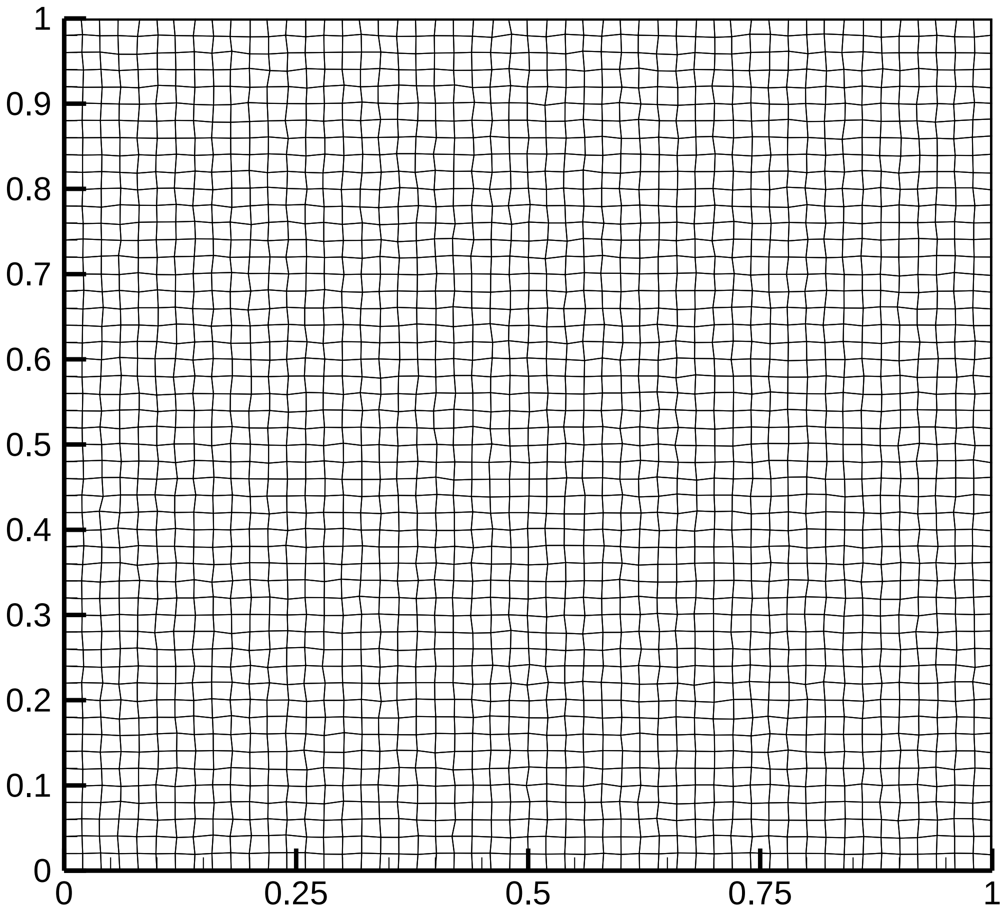
$$\frac{\partial \rho\vec{u}}{\partial t} + \nabla.\left(\rho\vec{u}\otimes(\vec{u} - \vec{u}_g)\right) = -\nabla p$$

$$\frac{\partial \rho e}{\partial t} + \nabla.\left(\rho e(\vec{u} - \vec{u}_g)\right)$$
$$= -p\nabla.\,\vec{u}$$

Arbitrary Lagrangian-Eulerian

Rezoning

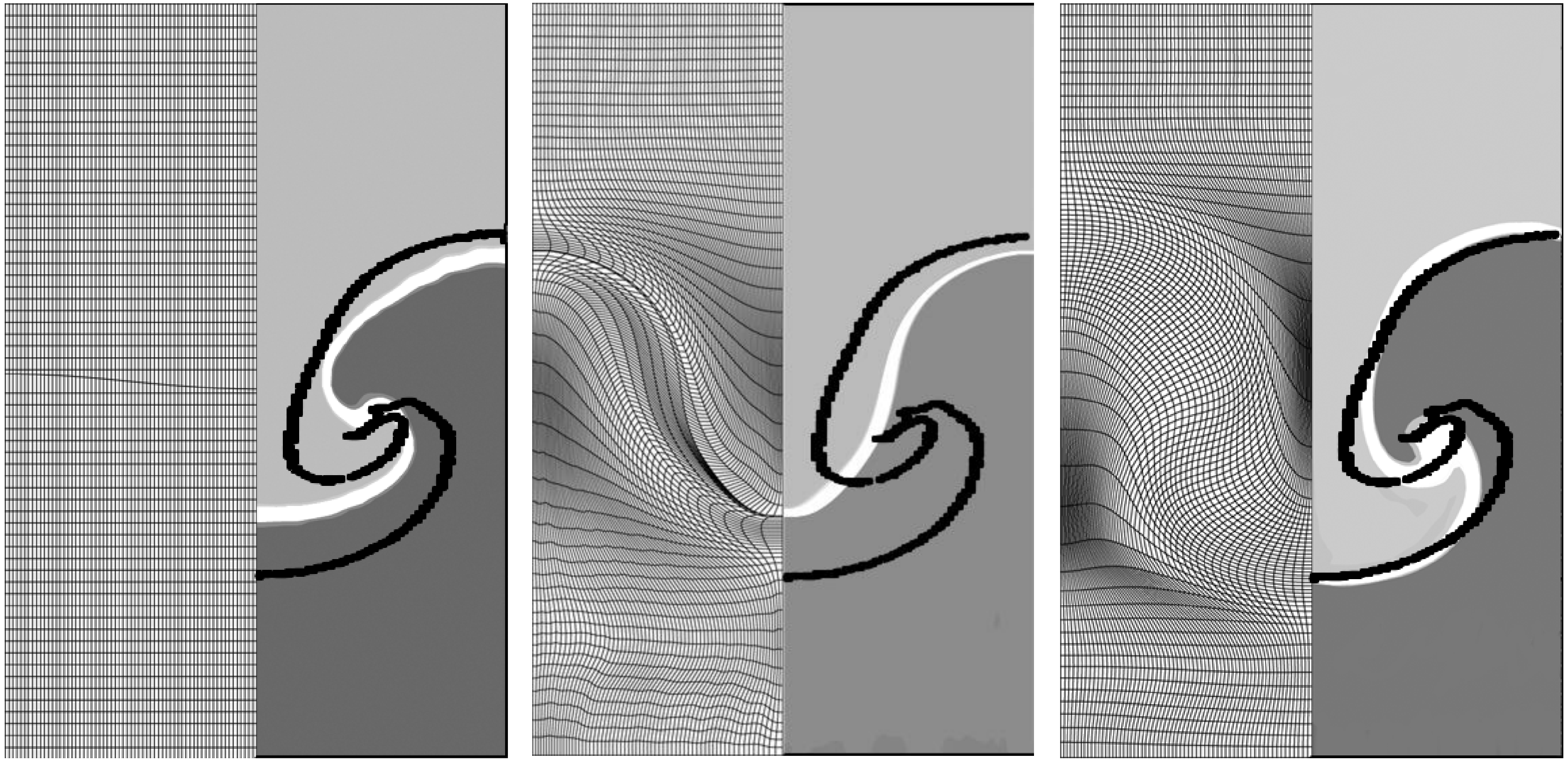




Smoothing computational grid

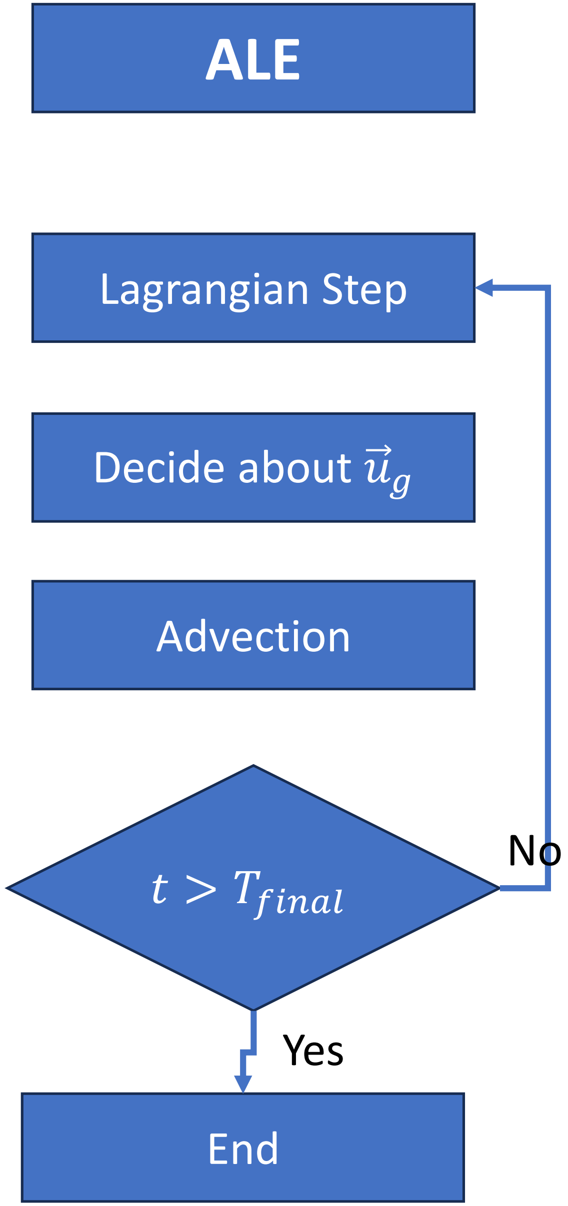
Arbitrary Lagrangian-Eulerian

Solution Strategy



Eulerian   Lagrangian   ALE

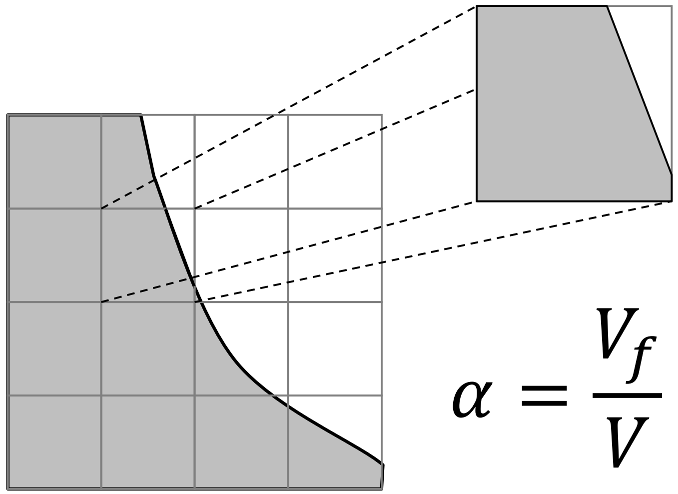
Rayleigh-Taylor Instability



Multi-Material Problems

Solution Strategy

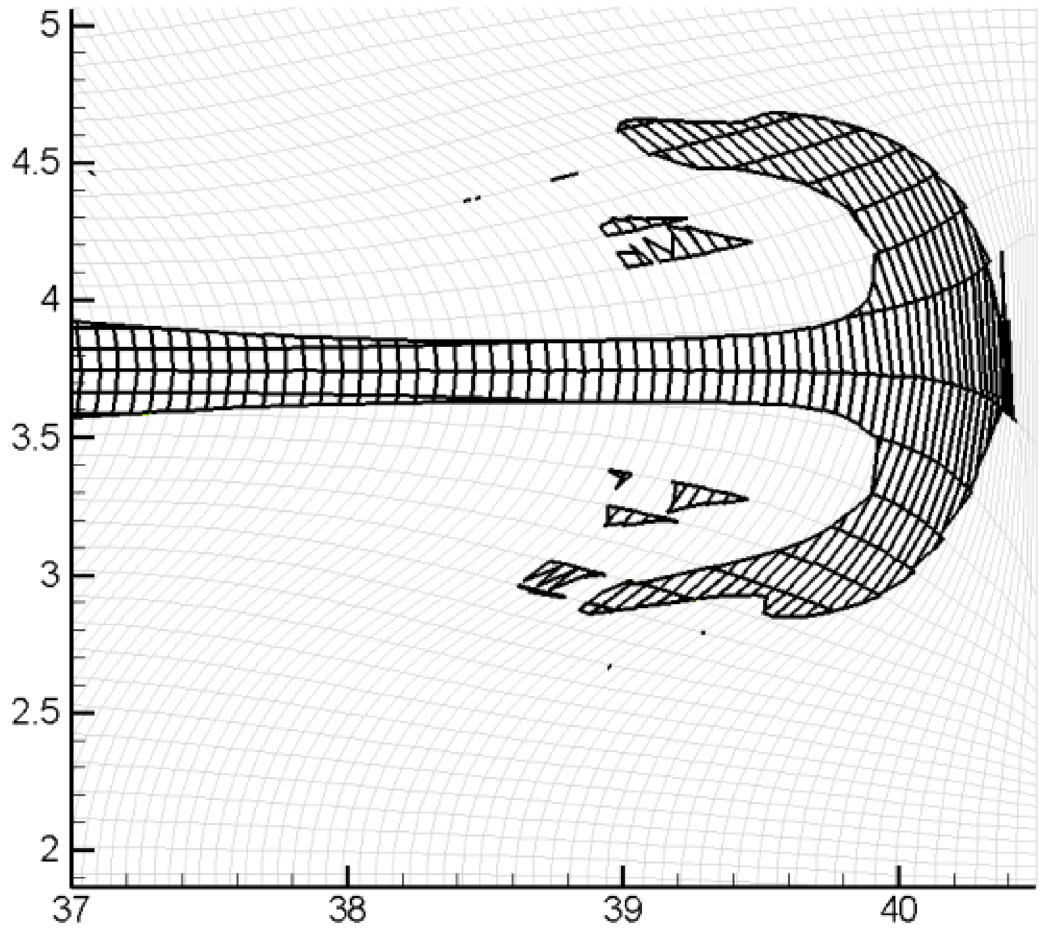
- Fluid mixtures:
  - Miscible fluid mixtures: mix completely, form homogenous mix without any separating layer.
  - Immiscible fluid mixtures: do not mix and instead form distinct separate layers.
- Strategies to model immiscible mixtures:
  - Level set,
  - Interface tracking,
  - Interface Reconstruction (VOF)
    - Advection of VOFs,
    - Reconstruction of interfaces,
    - Mixed cell dynamics.

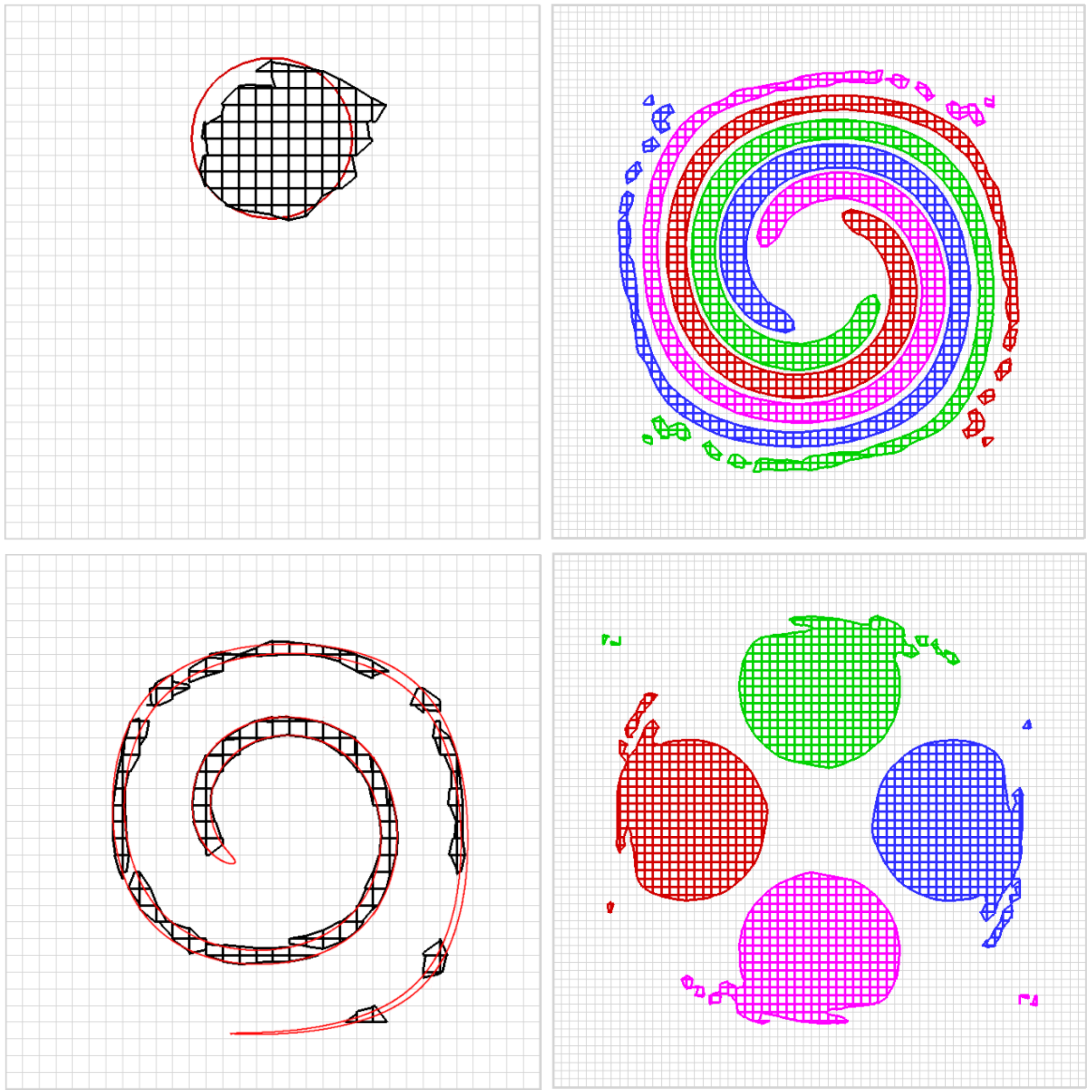


$$\alpha = \frac{V_f}{V}$$

Multi-Material Problems

Volume of fluid





Spectral Methods

Introduction

- Navier-Stokes Equation:

$$\nabla.\vec{u} = 0 \qquad \frac{\partial \vec{u}}{\partial t} + \nabla.(\vec{u} \otimes \vec{u}) = -\frac{1}{\rho}\nabla p + \nu\, \nabla^2 u$$

- Fourier Transform:

$$\hat{u}(k,t) = \int \vec{u}(x,t) e^{-ik.x} dx$$

- Important Property:

$$\mathcal{F}\{f'\}(k) = ik\mathcal{F}\{f\}(k)$$

- Applied to NS Eq.:

$$\frac{\partial \hat{u}}{\partial t} + \widehat{u.\nabla u} = -ik\hat{p} - \nu k^2 \hat{u}$$

Spectral Methods

Introduction

- Navier-Stokes Equation:

$$\nabla . \vec{u} = 0 \qquad\qquad \frac{\partial \vec{u}}{\partial t} + \nabla . (\vec{u} \otimes \vec{u}) = -\frac{1}{\rho} \nabla p + \nu \, \nabla^2 u$$

$$\nabla . \left(\frac{\partial \vec{u}}{\partial t} + \nabla . (\vec{u} \otimes \vec{u}) \right) = \nabla . \left(-\frac{1}{\rho} \nabla p + \nu \, \nabla^2 u \right)$$

$$\nabla . (\vec{u} . \nabla \vec{u}) = \nabla . \left(-\frac{1}{\rho} \nabla p \right)$$

$$\nabla^2 p = \rho \, \nabla . (-\vec{u} . \nabla \vec{u})$$

$$\hat{p} = \frac{i}{k^2} k . \widehat{\vec{u} . \nabla \vec{u}}$$

Spectral Methods

Introduction

- Navier-Stokes Equation:

$$\nabla . \vec{u} = 0$$

$$\frac{\partial \vec{u}}{\partial t} + \nabla . (\vec{u} \otimes \vec{u}) = -\frac{1}{\rho} \nabla p + \nu \, \nabla^2 u$$

- NS in Fourier Space:

$$\frac{\partial \hat{u}}{\partial t} + \widehat{\vec{u} . \nabla \vec{u}} = -ik\hat{p} - \nu k^2 \hat{u}$$

$$\hat{p} = \frac{i}{k^2} k . \widehat{\vec{u} . \nabla \vec{u}}$$

Numerical Implementation

Solution Steps:

1. Initialize velocity field $\vec{u}(x, t = 0)$ in physical space, and transfer it to Fourier space $\hat{u}(k, t = 0)$ using FFT.
2. Compute $\vec{u} . \nabla \vec{u}$ (PS) and transfer it to FS using FFT.
3. Compute $\hat{p}(k, t)$.
4. Solve for velocity $\hat{u}(k, t)$.
5. Transfer $\hat{u}$ to $\vec{u}(x, t)$ (PS) using iFFT.
6. Time advance using RK4.

Spectral Methods

Introduction

- Advantages:
  1. High Accuracy: exponential convergence $E = O(e^{-\alpha N})$,
  2. Efficient in FS: derivatives in FS are simpler,
  3. No numerical dissipation: unlike FD,FV and FE, SM does not introduce artificial dissipation, important in turbulent studies.
- Disadvantages:
  1. Problem in Irregular domains: periodic boundary condition,
  2. Computationally expensive for non-uniform grids,
  3. Cannot handle complex geometries easily.

Direct Numerical Simulation (DNS)

Introduction

- What is DNS?
  - Solving Navier-Stokes equation without any turbulence modeling (LES, RANS).
  - All turbulence scales must be resolved (Large energy-containing to smallest dissipative eddies).

- Needs extremely fine spatial and temporal grids:
  - Very expensive to compute,
  - Time consuming,
- Spatial grid must capture smallest turbulent scale (Kolmogorov size):

$$\eta = \left(\frac{v^3}{\varepsilon}\right)^{\frac{1}{4}}$$

Direct Numerical Simulation (DNS)

Introduction

- Spatial grid must capture smallest turbulent scale (Kolmogorov size):

$$\eta = \left(\frac{v^3}{\varepsilon}\right)^{\frac{1}{4}}$$

- Non-Newtonian fluids: What is the Kolmogorov scales?

$\tau = K\dot{\gamma}^n$ (Power-law type fluid)

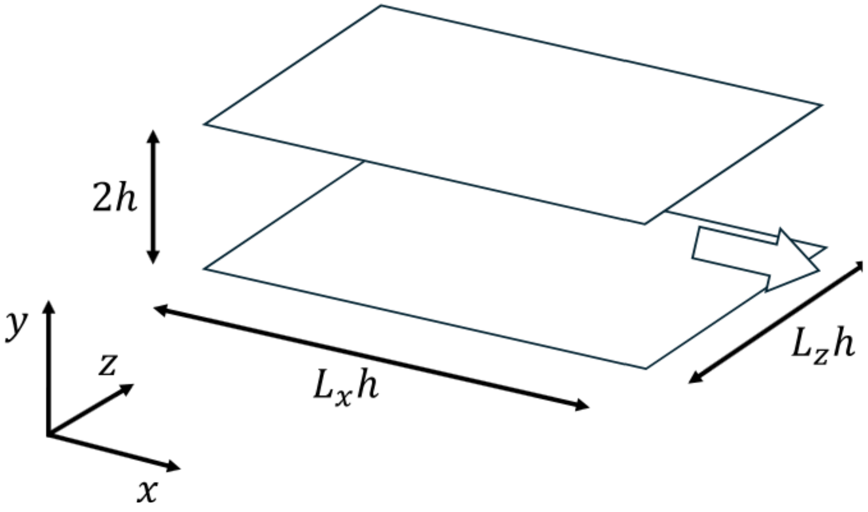
$$\eta \sim \left(\frac{K^3}{\rho^3 \varepsilon^{2-n}}\right)^{\frac{1}{2(n+1)}} \quad ; \quad u_\eta \sim \left(\frac{K\varepsilon^n}{\rho}\right)^{\frac{1}{2(n+1)}} .$$

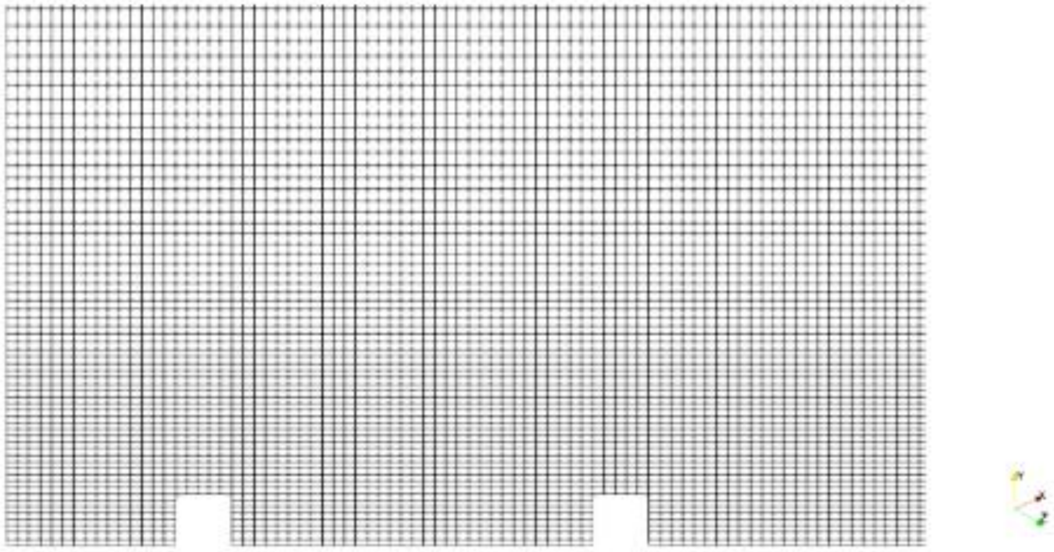
$$\frac{\eta}{L} \sim Re_{n,L}^{-\frac{3}{2(n+1)}} \quad ; \quad \frac{u_\eta}{V_L} \sim Re_{n,L}^{-\frac{1}{2(n+1)}} ,$$

Direct Numerical Simulation (DNS)

Geometry

- Turbulent flow:
  - 2D/3D Isotropic turbulence,
  - 3D Canonical flow (pipe, channel,..)
    - Spectral discretization in period directions,
    - High order finite difference discretization in other directions.

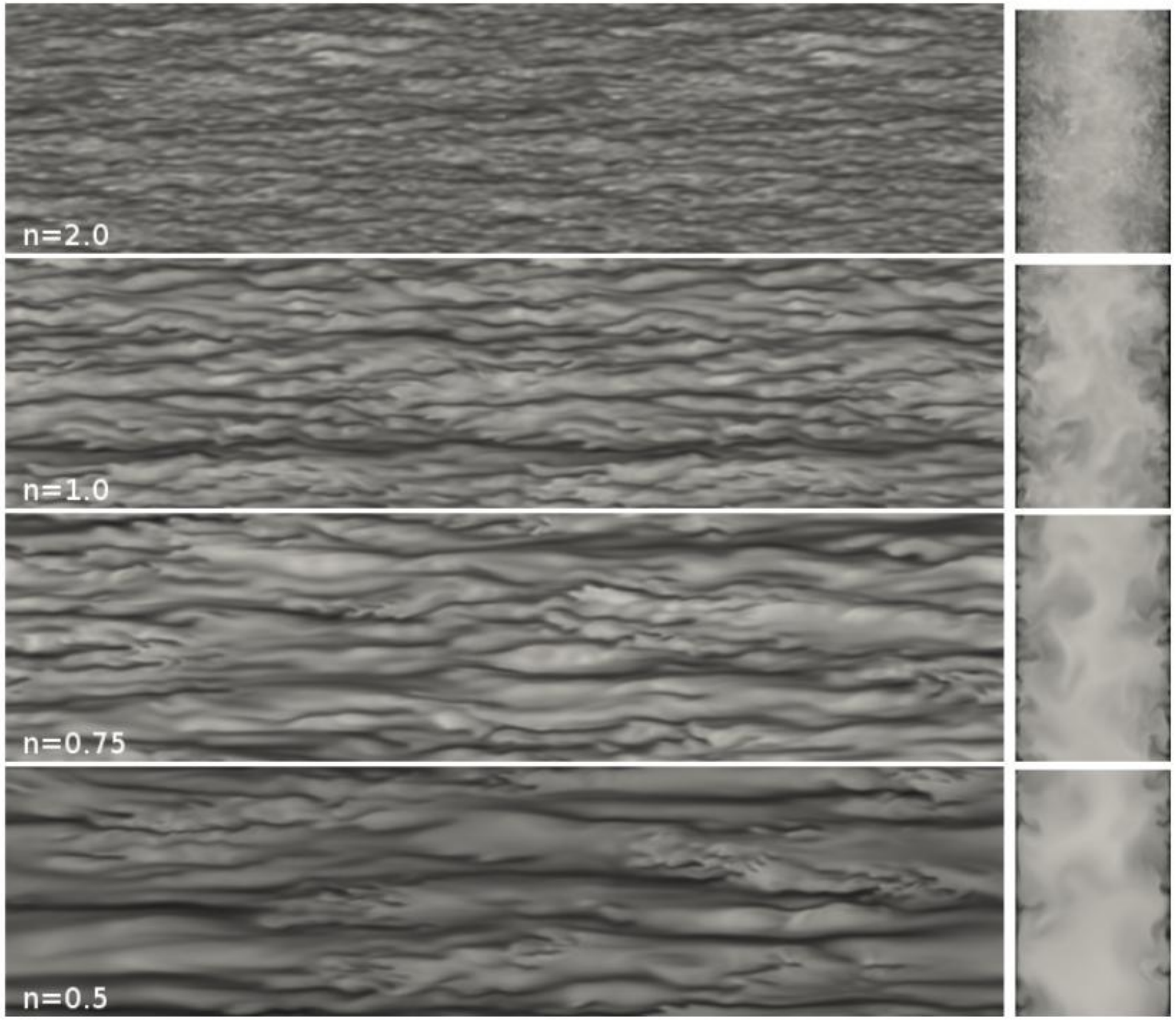




Direct Numerical Simulation (DNS)

Results

- Result of DNS:
  - Instantaneous velocity,
  - In all cells,
  - For many time steps.



Direct Numerical Simulation (DNS)

Simulations

Turbulent flow of non-Newtonian fluids

Power-law fluid
- shear thinning (n=0.75 and 0.5)
- shear thickening (n=2.0)

Channel Flow

$$Re_\tau = 395$$

Department of Applied Mathematics &
Mechanical Engineering Program
Federal University of Rio de Janeiro

H.R. Anbarlooei, D.O.A. Cruz, G.E.O. Celis, M.S.S. Macedo, R.L. Thompson

June 6, 2024

https://lcadame.github.io/

Laboratório de Computação de Alto Desempenho e Aprendizado de Máquina em Engenharia (L'CADAME)

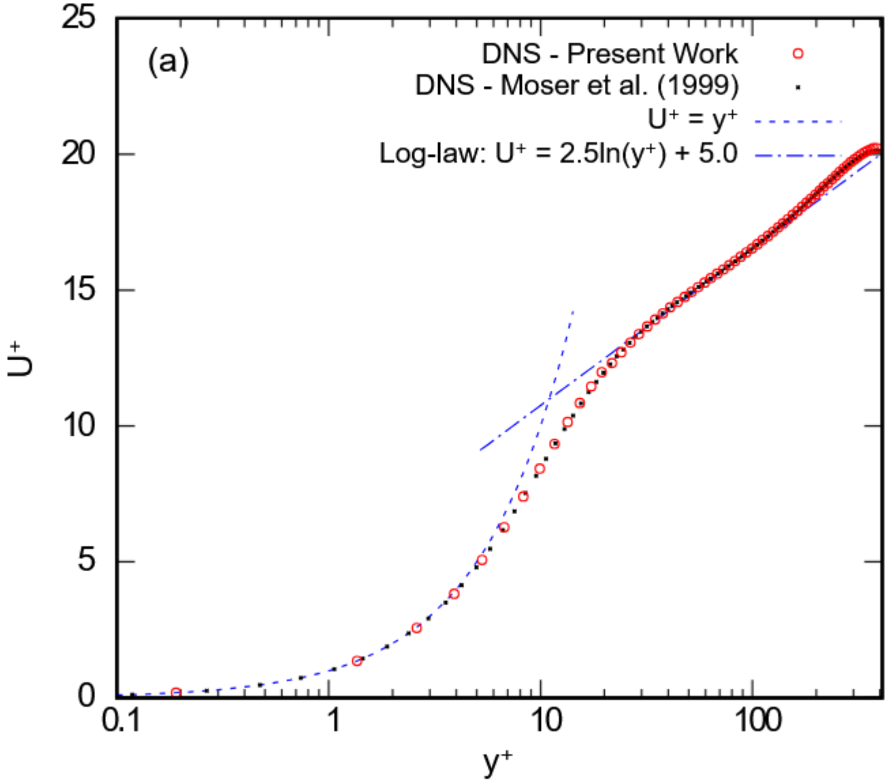
UFRJ

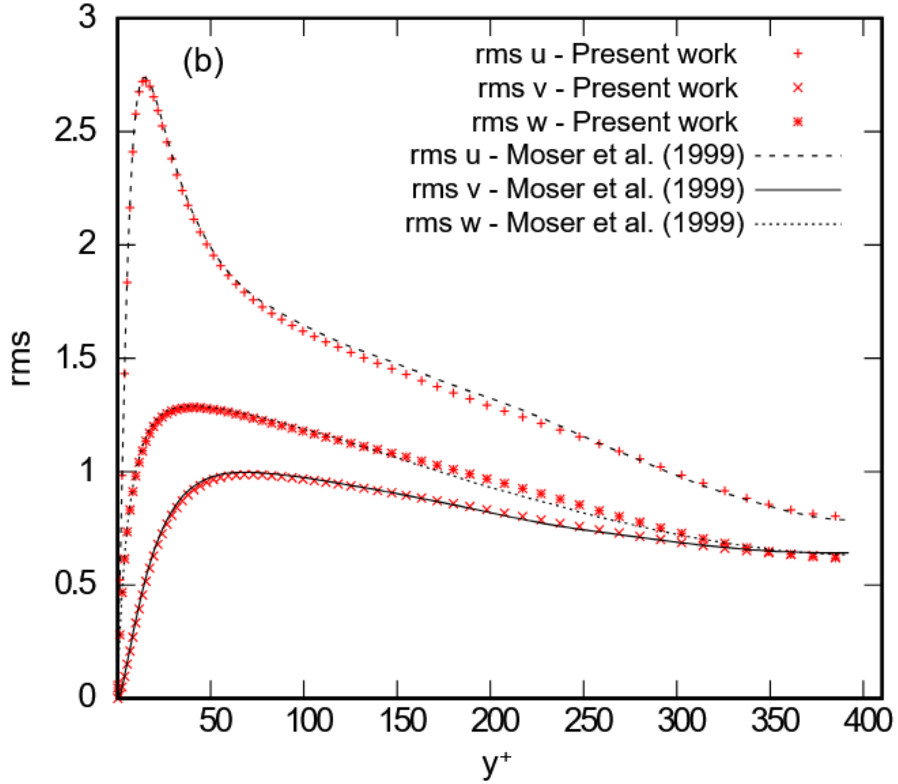
Direct Numerical Simulation (DNS)

Results

$$u = \bar{u} + \acute{u}$$
$$\langle u \rangle = \bar{u}$$

Averaging Operator

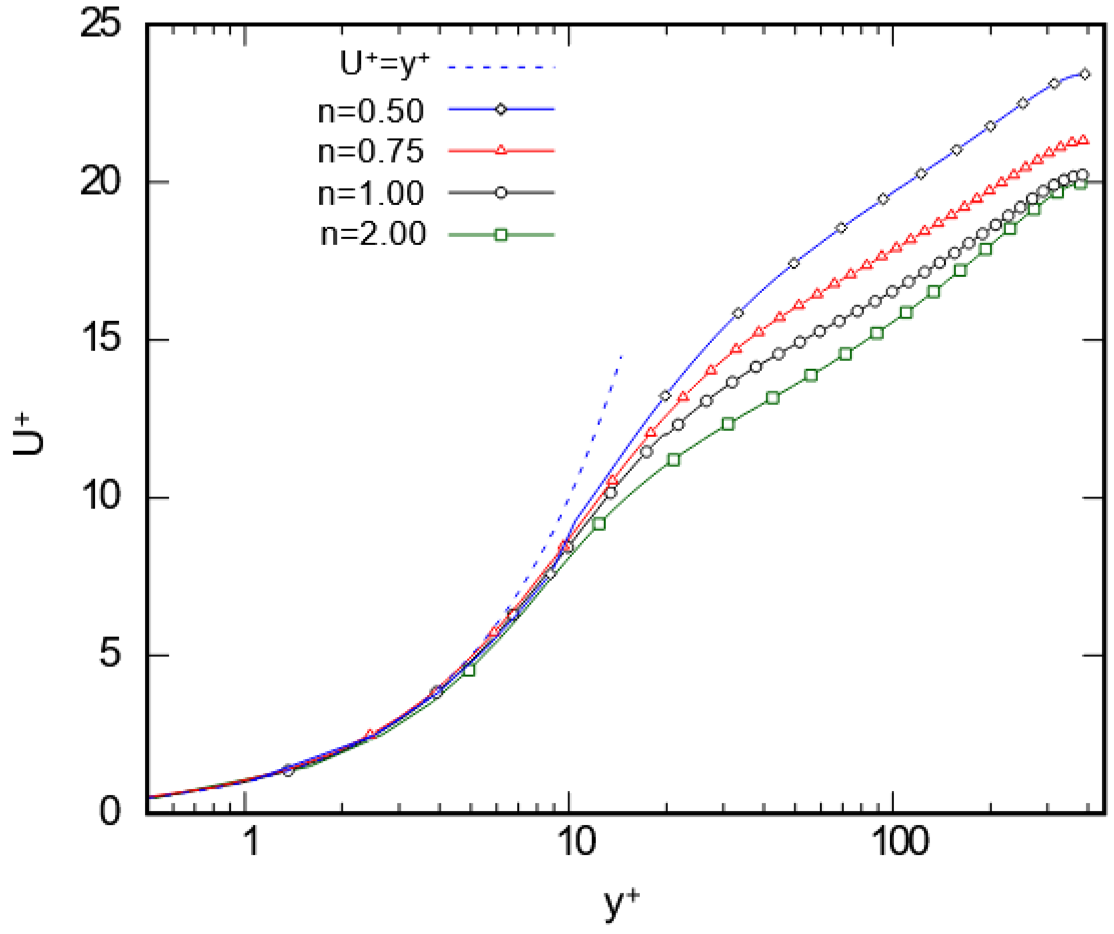


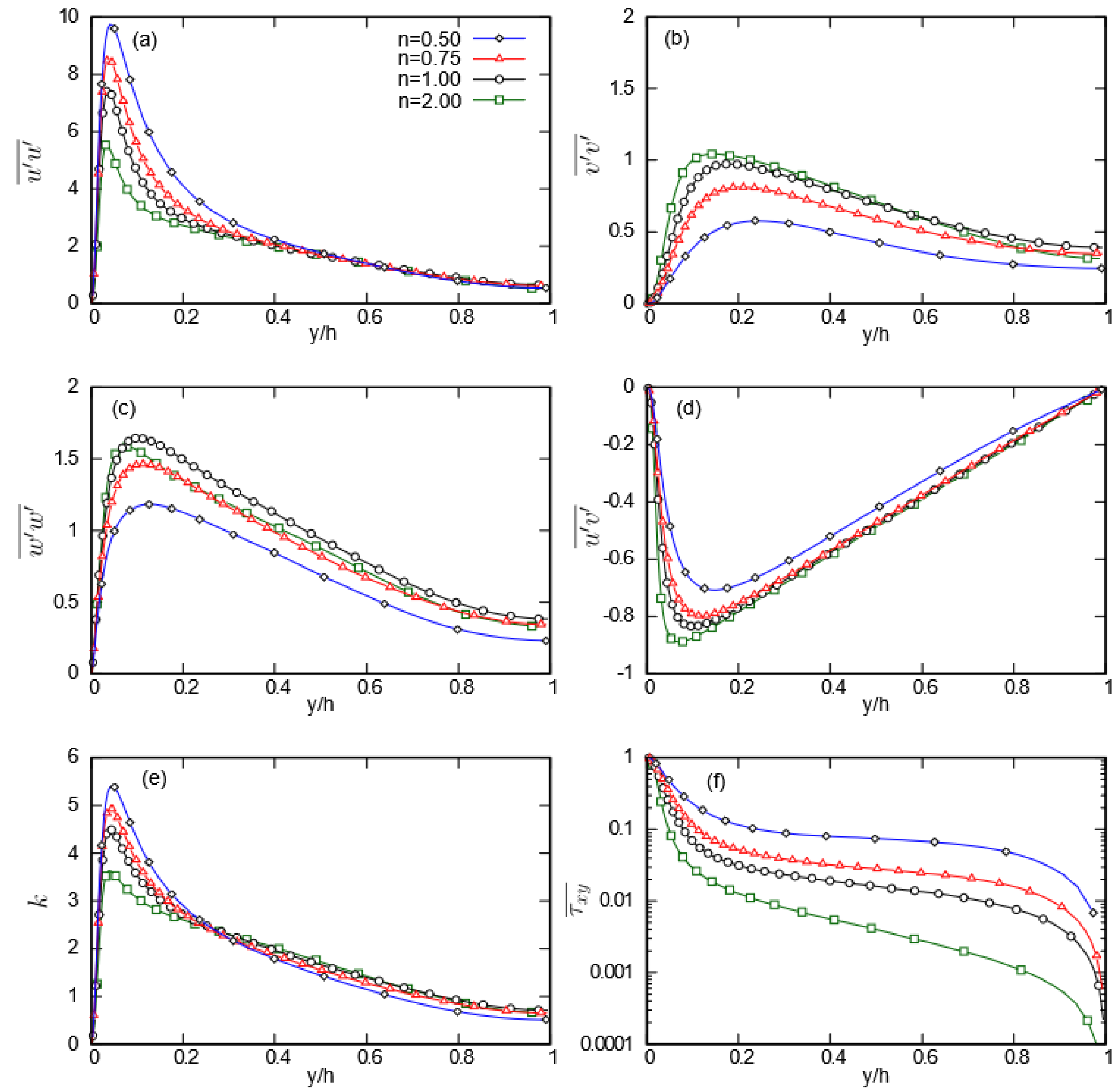


$$u = \bar{u} + \acute{u}$$
$$\langle u \rangle = \bar{u}$$
$$\langle uu \rangle = \langle (\bar{u} + \acute{u}) \times (\bar{u} + \acute{u}) \rangle = \langle \bar{u}^2 + 2\bar{u}\acute{u} + \acute{u}^2 \rangle$$
$$\langle uu \rangle = \langle \bar{u}^2 \rangle + \langle 2\bar{u}\acute{u} \rangle + \langle \acute{u}^2 \rangle = \langle \bar{u}^2 \rangle + \langle \acute{u}^2 \rangle$$

Direct Numerical Simulation (DNS)

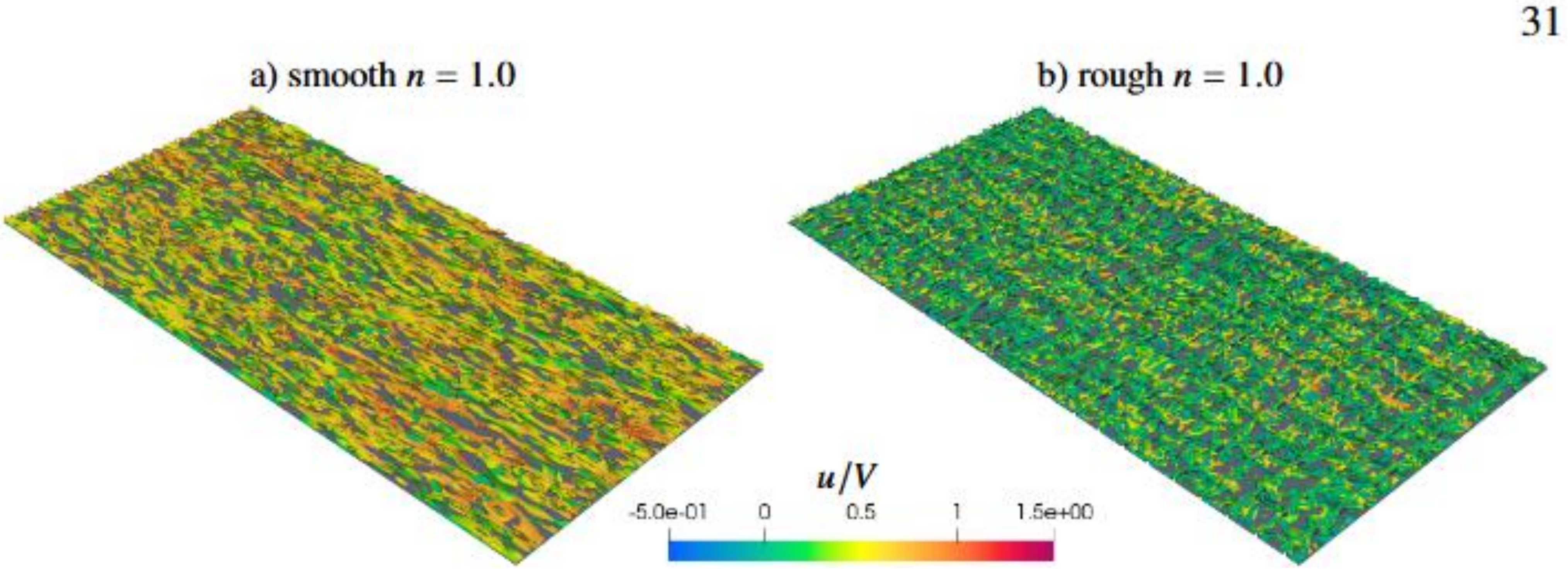
Non-Newtonian fluids

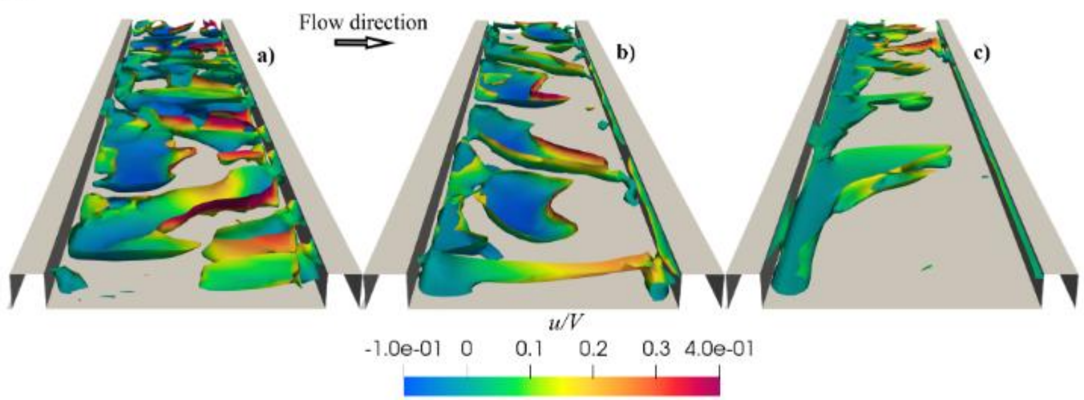




Direct Numerical Simulation (DNS)

Non-Newtonian fluids – rough channel





Integrating Machine Learning with Fluid Simulation

Laboratório de Computação de Alto Desempenho e Aprendizado de Máquina em Engenharia (L'CADAME)

Kalman Filter

Introduction

- For many practical applications, we don't have proper IC.
  - Atmosphere or ocean current modeling
- But we have some partial observations in time.
  - Satellite weather measurements,
  - Temperature, humidity and … in local stations.
- Usually we start from some IC which make sense,
- The goal is to improve the IC, as soon as a new observations arrive (data assimilation).
  - Observations are not error free,
  - Different observations have different fidelities.

Kalman Filter

Linear model

- Prediction step (numerical simulation with bad IC):

$$\hat{x}_{k|k-1} = A\,\hat{x}_{k-1|k-1} + B\,u_k$$

$$P_{k|k-1} = A\,P_{k-1|k-1}A^T + Q$$

- Update step(correction using new observations):

$$K_k = P_{k|k-1}H^T\left(HP_{k|k-1}H^T + R\right)^{-1}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(z_k - H\hat{x}_{k|k-1}\right)$$

$\hat{x}_{k|k-1}$: numerical simulation

z_k: observation

P: covariance matrix

R:observation noise

Ensemble Kalman Filter

Non-linear model

- Prediction step (numerical simulation with bad IC):

$$\hat{x}^i_{k|k-1} = f\left(\hat{x}^i_{k-1|k-1}\right) + W^i_k$$

$$P_{k|k-1} = \mathbb{E}[(X - \bar{X})(X - \bar{X})^T]$$

- Update step(correction using new observations):

$$K_k = P_{k|k-1}\left(P_{k|k-1} + R\right)^{-1}$$

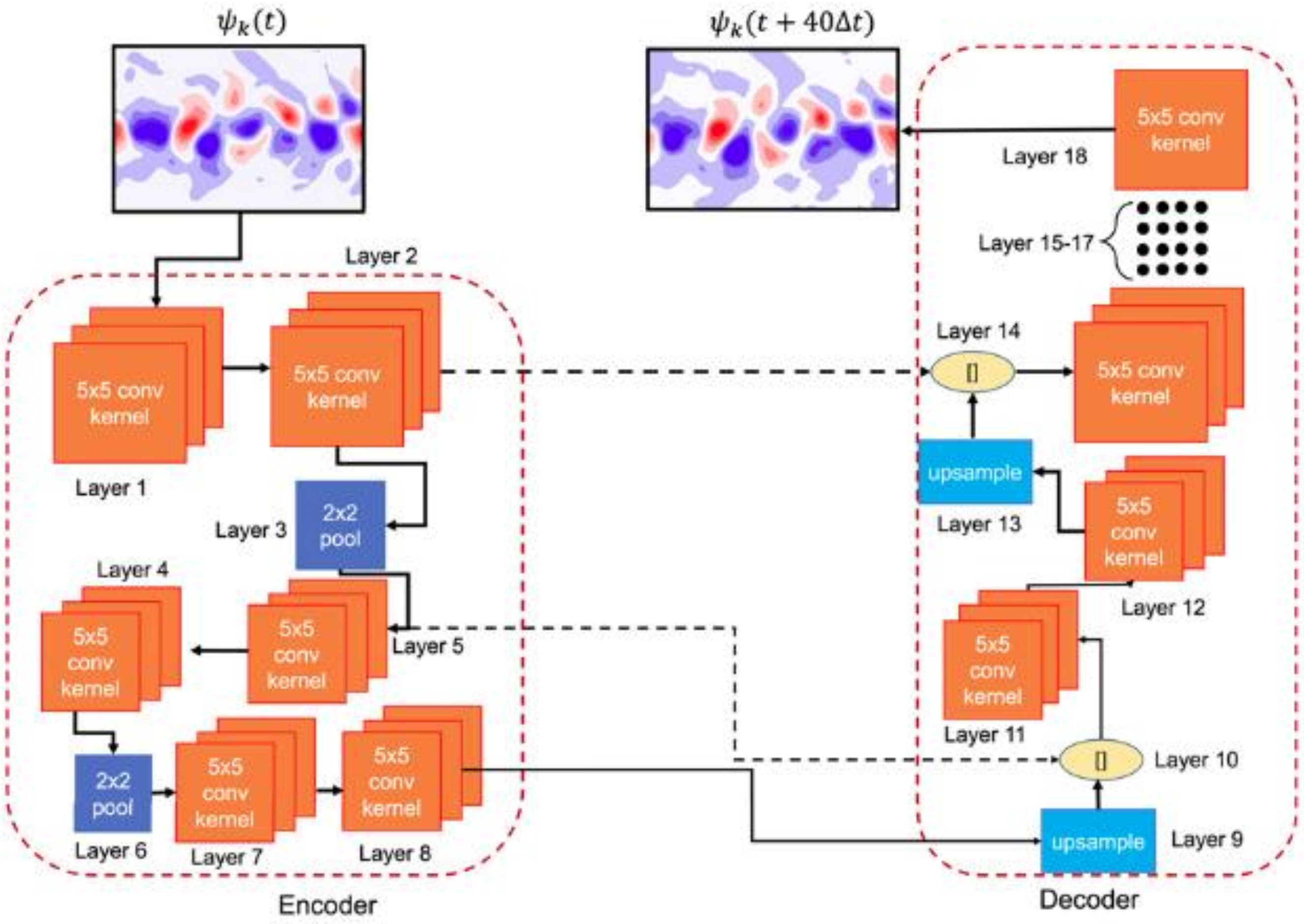
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(z_k - \hat{x}_{k|k-1}\right)$$

- Calculating a good $P_{k|k-1}$ needs a large ensemble of numerical simulations, which is very expensive.

Ensemble Kalman Filter

Machine learning to replace numerical integration

- Numerical methods:
  - Expensive,
  - Time marching,
  - High accuracy.
- Machine learning:
  - Long time integration,
  - Cheap,
  - Lower accuracy.



The U-NET model (surrogate) for approximating **X**.

Ensemble Kalman Filter

Numerical + Machine Learning

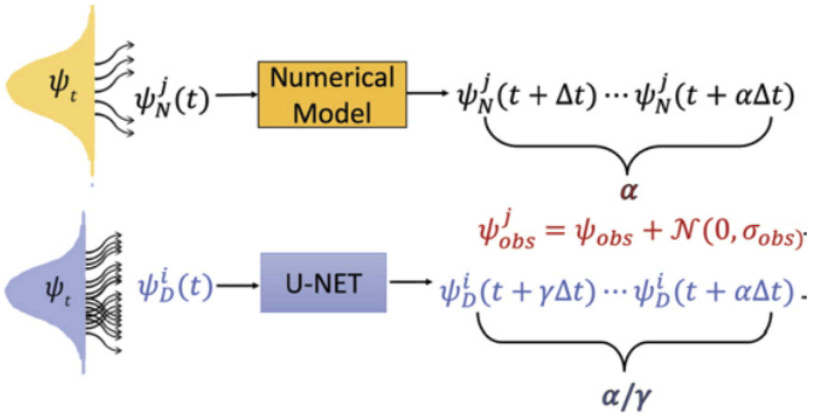
Filter

$$\hat{x}^i_{k|k-1} = f\left(\hat{x}^i_{k-1|k-1}\right) + W^i_k$$

$$P_{k|k-1} = \mathbb{E}[(X-\bar{X})(X-\bar{X})^T]$$

$$K_k = P_{k|k-1}\left(P_{k|k-1}+R\right)^{-1}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(z_k - \hat{x}_{k|k-1}\right)$$



Ensemble Kalman Filter U-NET model (surrogate) for approximating **X**.

$$P_D = \mathbb{E}\left[\left(\psi^i_D(t+\alpha\Delta t) - \bar{\psi}(t+\alpha)\right)\left(\psi^i_D(t+\alpha\Delta t) - \bar{\psi}(t+\alpha\Delta t)^T\right)\right]$$

$$K_D = P_D(P_D + \sigma_{\text{obs}} I)^{-1}$$

$$\psi^j_a = \psi^j_N(t+\alpha\Delta t) - K_D\left(\psi_{\text{obs}} - \psi^j_N(t+\alpha\Delta t)\right)$$

Ensemble Kalman Filter

Results

Data Assimilation and Machine Learning

Ensemble Kalman Filter

Department of Applied Mathematics
Federal University of Rio de Janeiro

R. Arefidamghani, H.R. Anbarlooei

June 18, 2024

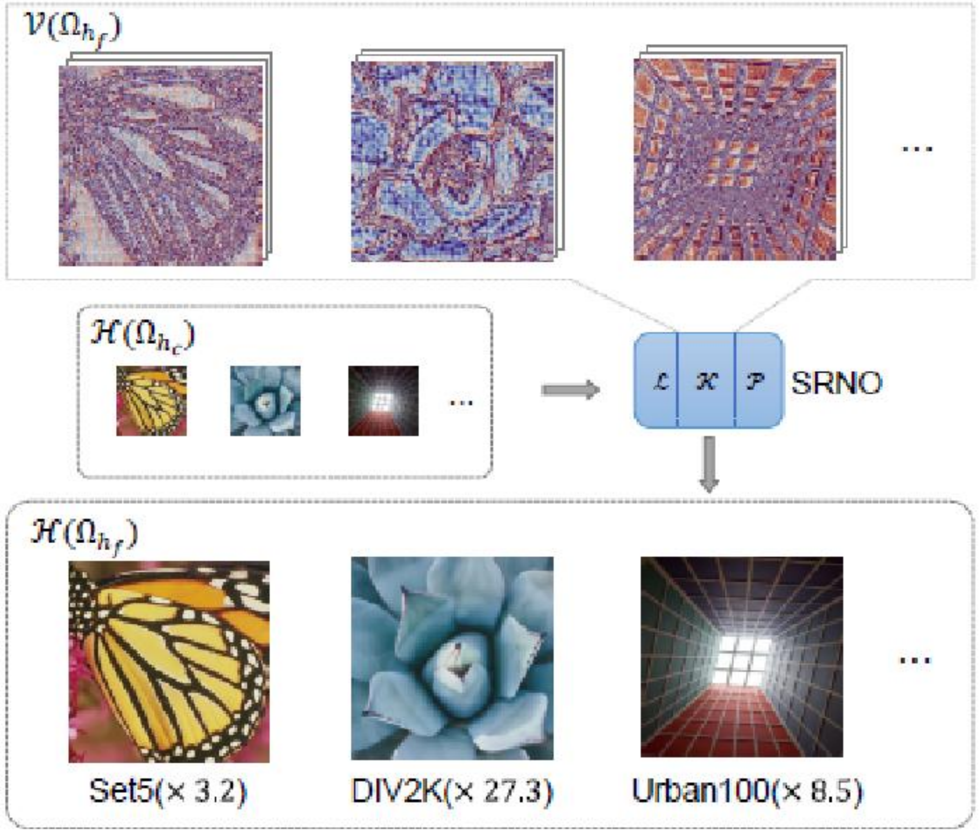
Laboratório de Computação de Alto Desempenho e Aprendizado de Máquina em Engenharia (L'CADAME)

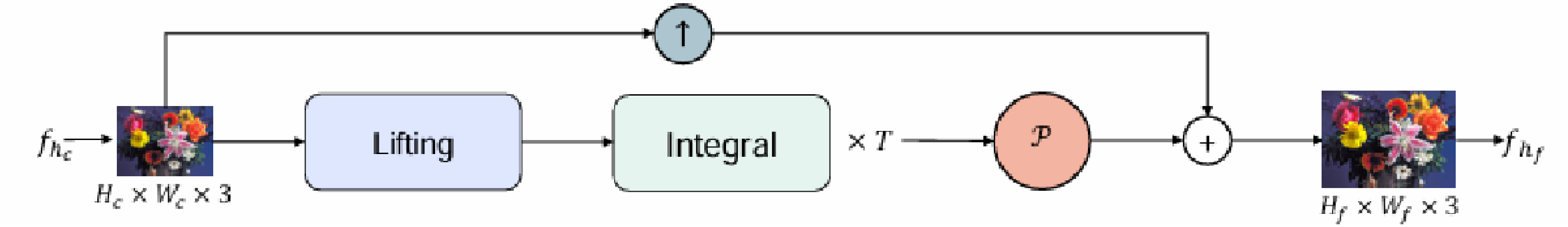
UFRJ

Super-Resolution

Introduction

- Super-resolution: low resolution image to high quality image using machine learning.
- Idea: perform fast low resolution simulations and using ML to improve the quality of the results.
- Goal: recover the missing features in low res simulations.

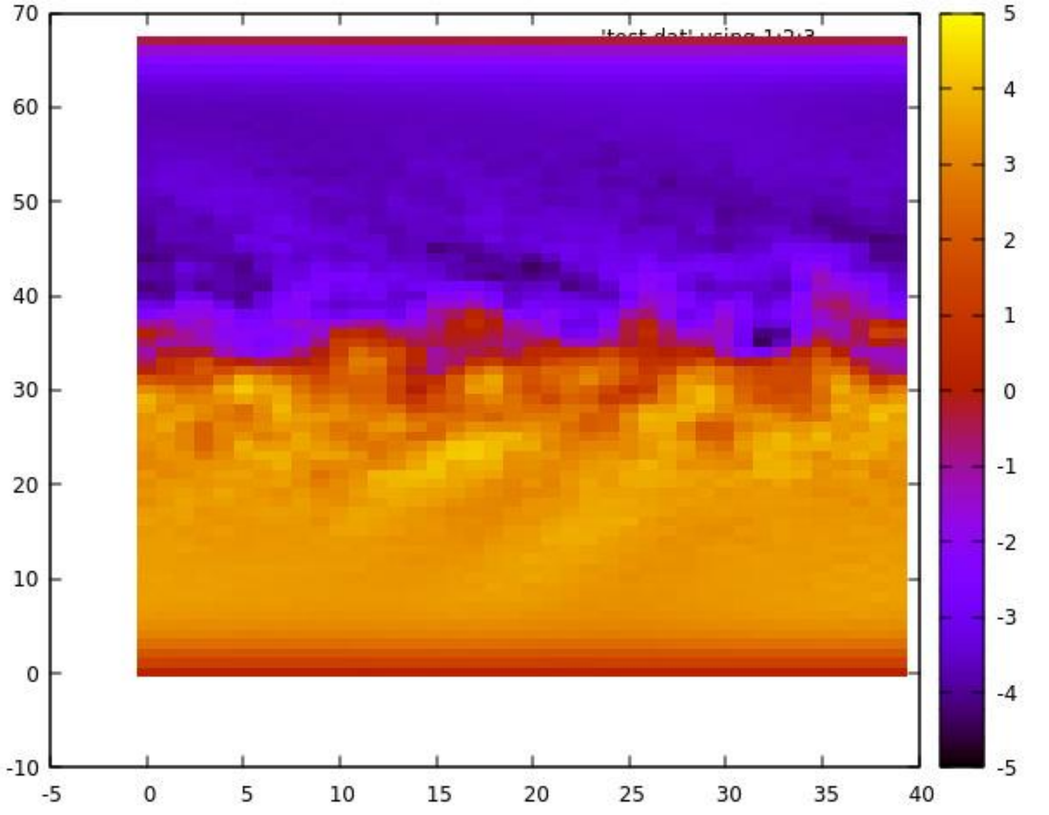




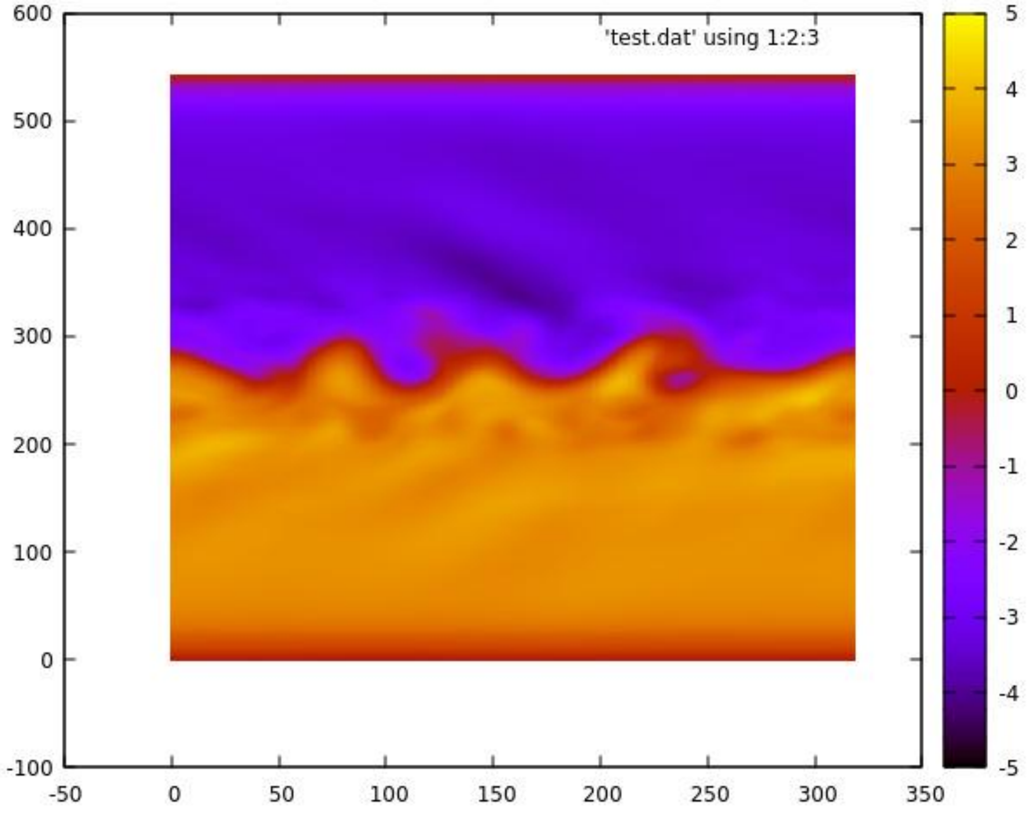
SRNO architecture.

Super-Resolution

Results



Low Res. Numerical Simulations

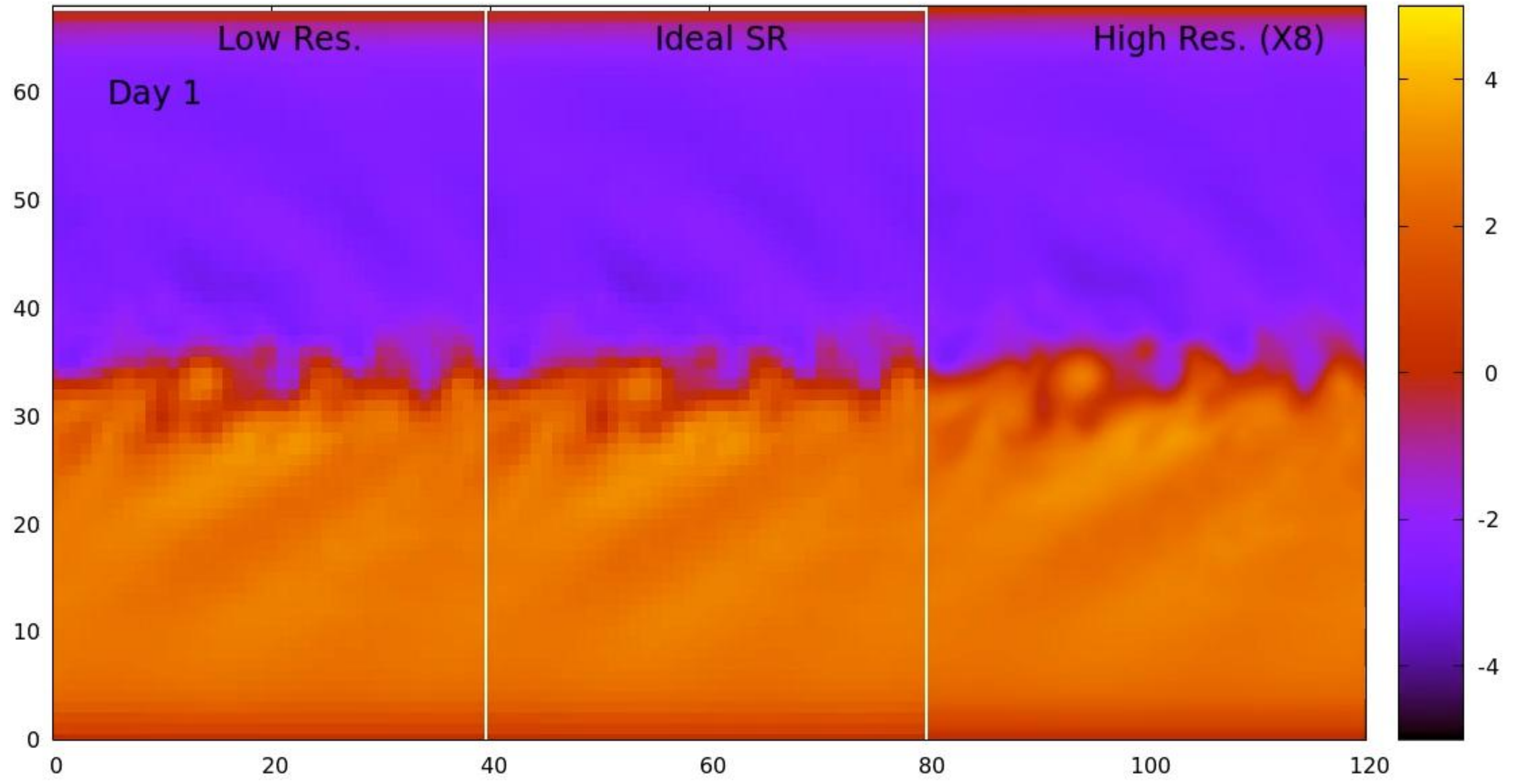


High Res. Super-Resolution

Super-Resolution

Results

- Low res simulations lose accuracy in time,
- Some fine features does not appear in low res simulations.



Pipeline

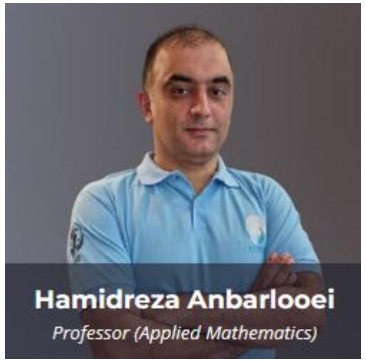
1- Low res. Simulation for a period of time,

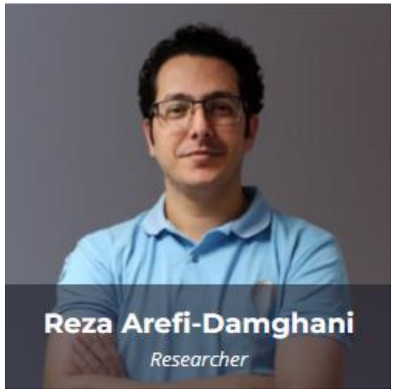
2- One high res. reconstruction,

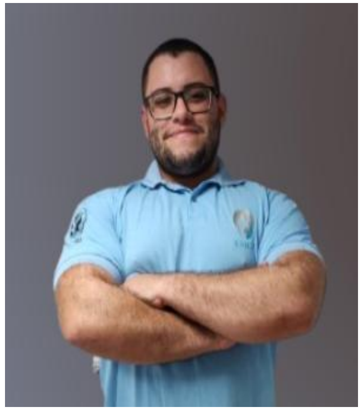
3- Down grading the high res. Data to produce IC for low res. simulation

L’CADAME

Team







L’CADAME

Laboratório de Computação de Alto Desempenho e Aprendizado de Máquina em Engenharia (L'CADAME)

https://lcadame.github.io/

Email: cadamelab@gmail.com

Thank You!

For your time and attention.

Feel free to reach out for any questions or further discussion.

Contact Information: hamidreza@im.ufrj.br

Laboratório de Computação de Alto Desempenho e Aprendizado de Máquina em Engenharia (L'CADAME)

UFRJ