Rescuer Worshop 2: Introduction and review on the solution of the shallow water equations, Focusing on higher order numerical schemes for the shallow water equations

Maria Kazolea [1]

INRIA Center at the University of Boredeaux [1]
Cardamom team

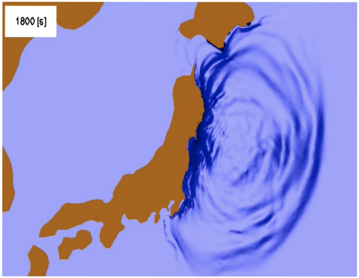
March 12-14, 2025

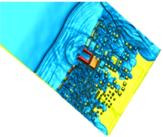
RESCUER

Funded by the European Union

Motivation

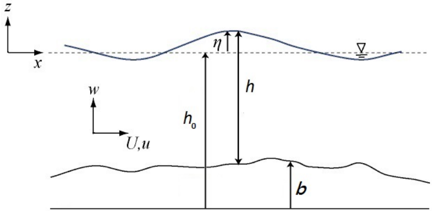
Bonneton et al (2015)





2011 great Tohoku tsunami,
Naka river at Hitachinaka city

The physical problem



η: **free surface elevation**;
h_0: **steel water level**;
$b(x)$: **bottom's topography variation**;
$h(x,t) = h_0 + \eta(x,t) - b(x)$: **total water depth**;
$u(x,t)$: **flow velocity**;

Basic Assumptions

- The fluid is incompressible and inviscid.
- The horizontal velocity components are much larger than the vertical.
- The flow is hydrostatic: vertical acceleration is negligible.
- The bottom topography can not vary, $z = b(x)$.
- The free surface elevation is $z = h(x,t) + b(x)$.

Mass Conservation (Continuity Equation)

Consider a small control volume of width $dx = [x_1,\ x_2]$, depth $h(x,t)$, and velocity $u(x,t)$. Applying conservation laws to this volume leads to the governing equations. The rate of change of fluid's mass in t is:

$$\frac{d}{dt}\int_{x_1}^{x_2} \rho h(x,t)dx$$

The mass flux is:

$$\int_{b}^{h+b} \rho u(x,t)dy = \rho u(x,t)h(x,t)$$

The integral form of conservation of mass:

$$\int_{t}^{t+\Delta t}\int_{x_1}^{x_2} h(x,t) + \frac{d}{dx}(hu)dx = 0$$

Differential form:

$$\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} = 0$$

Momentum Conservation Using Newton's Second Law

Newton's second law states that:

Rate of change of momentum $==$ to the sum of external forces:

Rate of change of momentum:

$$\frac{D}{Dt}\int_{x_1}^{x_2}\int_{h}^{h+b}\rho u dy dx = \frac{d}{dt}\rho hu\, dx + \int_{x_1}^{x_2}\int_{h}^{h+b}\rho\frac{d}{dx}u^2 dy dx$$

Side forces:

$$g[\rho(y-h-b)dy]_{x_1}^{x_2} = [-\frac{1}{2}g\rho h^2]_{x_1}^{x^2}$$

Topography forces:

$$-g\int_{x_1}^{x_2}\rho h b_x$$

Momentum Conservation Using Newton's Second Law

Expanding the time derivative and using mass conservation:

$$\frac{\partial(hu)}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) = -gh\frac{\partial b}{\partial x}$$

This is the momentum equation in shallow water form.

Shallow Water Equations in Vector Form

System of Conservation Laws:

$$\mathbf{q}_t + \mathbf{F}(\mathbf{q})_x = S(q)$$

$$\mathbf{q} = \begin{bmatrix} h \\ hu \end{bmatrix}, \quad \mathbf{F}(\mathbf{q}) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix}, \quad S(q) = \begin{bmatrix} 0 \\ -gh\frac{\partial b}{\partial x} \end{bmatrix}.$$

Linearization: Jacobian Matrix

Computing the Jacobian matrix:

$$\mathbf{F}'(\mathbf{q}) = \begin{bmatrix} 0 & 1 \\ -u^2 + gh & 2u \end{bmatrix}$$

The eigenvalues of this system characterize wave propagation.

Wave Speeds and Eigenvectors

Characteristic Wave Speeds:

$$\lambda^1 = u - c, \quad \lambda^2 = u + c \quad \text{where } c = \sqrt{gh}$$

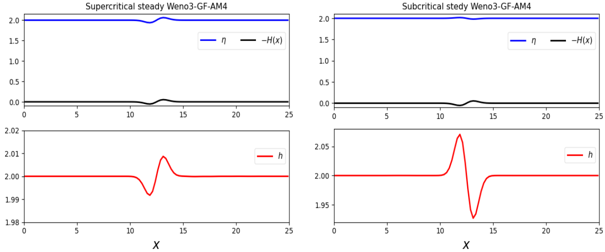
$$\mathbf{r}_1 = \begin{bmatrix} 1 \\ u - c \end{bmatrix}, \quad \mathbf{r}_2 = \begin{bmatrix} 1 \\ u + c \end{bmatrix}$$

These describe the propagation of shallow water waves.

Froude number Fr

$$Fr = \frac{u}{c}$$

$Fr > 1$ supercritical flow , $Fr < 1$ subcritical flow



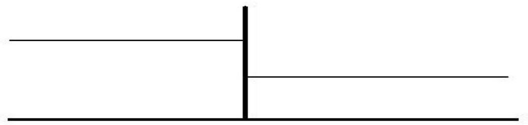
The Riemann problem

Definition of the Riemann Problem

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = 0$$

The initial condition is a piecewise constant function with a single discontinuity at $x = 0$:

$$\mathbf{q}(x,0) = \begin{cases} \mathbf{q}_L, & x < 0 \\ \mathbf{q}_R, & x > 0 \end{cases}$$

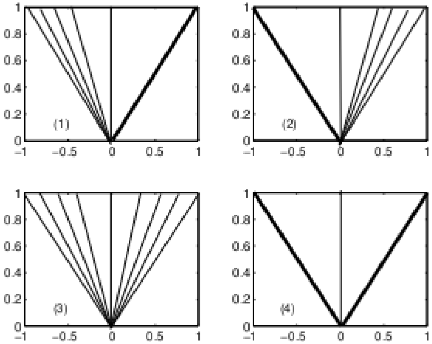


- The Riemann problem consists of solving a hyperbolic system of conservation laws with discontinuous initial conditions.

Possible Solutions to the Riemann Problem

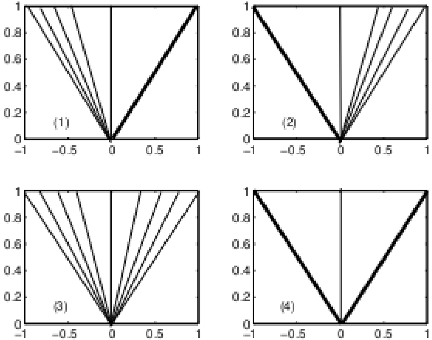
Types of Solutions

- The solution consists of waves propagating from the discontinuity.
- Depending on the initial states, we can have:
  1. Two shock waves (strong discontinuities)
  2. Two rarefaction waves (smooth expansion waves)
  3. A shock-rarefaction combination



Possible Solutions to the Riemann Problem

Types of Solutions



Wave Structure

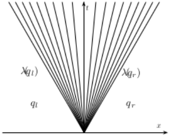
$$\lambda_1 = u - c, \quad \lambda_2 = u + c$$

where $c = \sqrt{gh}$ is the wave speed.

Sketch of the solution

- By solving the Riemann problem, two families of characteristic curves arise over which the solution is simplified and it holds that $\frac{dx}{dt} = \lambda_1$ for the first family and $\frac{dx}{dt} = \lambda_2$ for the second. To keep: The solution of the Riemann problem always remain constant in $x = \epsilon t$ with $\epsilon \in R$
- q_l and q_l are connected through a smooth wave

$$\mathbf{q}(x,0) = \begin{cases} \mathbf{q}_L, & x < \epsilon_1 t \\ \mathbf{W}(x/t), & \epsilon_1 t < x < \epsilon_2 t \\ \mathbf{q}_R, & x > \epsilon_2 t \end{cases}$$



By solving an ODE system with an initial condition $\mathbf{q}_l$ we can find :

$$u - u_l = 2\left(\sqrt{gh_l} - \sqrt{gh}\right) \rightarrow h_r < h_l \text{ and } u_r > u_l \quad (1)$$

$$u - u_l = -2\left(\sqrt{gh_l} - \sqrt{gh}\right) \rightarrow h_r > h_l \text{ and } u_r > u_l \quad (2)$$

Sketch of the solution

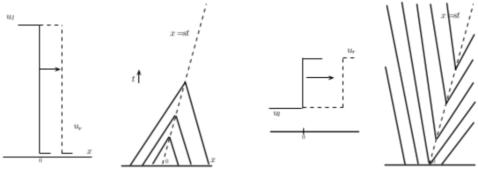
- q_l and q_l are connected through a shock wave.
- Rankie-Hugoniot conditions ensure that mass and momentum are conserved across the discontinuity. $\mathbf{f}(\mathbf{q_r}) - \mathbf{f}(\mathbf{r_r}) = s(\mathbf{q_r} - \mathbf{q_l})$
- Shock wave is propagating at speed s

$$s = \frac{h_r u_r - hl_u l}{h_r - h_l}$$

Entropy Conditions

What is an Entropy Condition?

- The Rankine-Hugoniot jump conditions allow multiple weak solutions.
- Entropy conditions select the physically correct (entropy-satisfying) shock.
- Ensures that information does not travel backward and that shocks are stable.



Shock waves: entropy satisfying (left), not satisfying (right)

Entropy Conditions

Lax's Entropy Condition (1957)

- Ensures that characteristics enter the shock but do not leave it.
- For a shock traveling at speed S, the characteristics satisfy:

$$\lambda_1^R < s < \lambda_1^L \quad \text{or} \quad \lambda_2^R < s < \lambda_2^L$$

Oleinik's Entropy Condition

- Strengthens Lax's condition by requiring a decrease in wave speed across the shock.
- The shock speed S satisfies:

$$S > \frac{u_L + u_R}{2}$$

ensuring that the shock does not allow expansion waves to develop.

1D SW without a source term

Shallow water equations written in a vector form

$$\mathbf{q_t} + \mathbf{F(q)_x} = \mathbf{0}$$

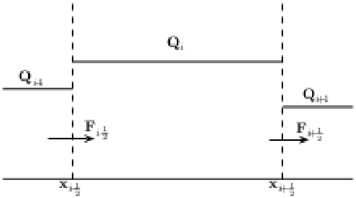
$$\mathbf{q} = \begin{bmatrix} h \\ hu \end{bmatrix}, \quad \mathbf{F(q)} = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix}.$$

$$\mathbf{F}'(\mathbf{q}) = \begin{bmatrix} 0 & 1 \\ -u^2 + gh & 2u \end{bmatrix} \quad \text{Jacobian Matrix}$$

$$\lambda^1 = u - c, \quad \lambda^2 = u + c \quad \text{eigenvalues and} \quad c = \sqrt{gh}$$

$$r_1 = \begin{bmatrix} 1 \\ u - c \end{bmatrix}, \quad r_2 = \begin{bmatrix} 1 \\ u + c \end{bmatrix} \quad \text{eigenvectors}$$

A first order Finite Volume Method for the Shallow Water Equations



Control volume for a FV scheme

Key Concepts:

- Divide the domain into control volumes (cells).
- Approximate the solution by cell averages.
- Use numerical fluxes to update the solution.

Deriving the Finite Volume Scheme

Computational Grid:

- Divide the domain into cells: $C_i = [x_{i-1/2}, x_{i+1/2}]$.
- The cell center is at x_i.
- The cell width is $\Delta x = x_{i+1/2} - x_{i-1/2}$.

Cell Averages:

$$\bar{\mathbf{Q}}_i(t) = \frac{1}{\Delta x} \int_{C_i} \mathbf{q}(x,t)\, dx.$$

Integrating the conservation law in the control volume and in time interval τ^n we get:

$$\frac{d}{dt} \int_{\tau^n} \bar{\mathbf{Q}}_i dt + \int_{\tau^n} \mathbf{f}(\mathbf{q}(x_{i+1/2})) - \mathbf{f}(\mathbf{q}(x_{i-1/2})) dt = 0$$

Deriving the Finite Volume Scheme

Discrete Conservation Law:

$$\frac{d}{dt}\bar{\mathbf{Q}}_i + \frac{1}{\Delta x}\left[\mathbf{F}^n_{i+1/2} - \mathbf{F}^n_{i-1/2}\right] = 0.$$

Using an explicit first order time discretization (Euler method) we get:

$$\bar{\mathbf{Q}}_i^{n+1} = \bar{\mathbf{Q}}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{F}^n_{i+1/2} - \mathbf{F}^n_{i-1/2}\right).$$

We need to define the numerical flux

A naive choice:

$$\mathbf{F^n_{i-1/2}} = \frac{\mathbf{1}}{\mathbf{2}}\left(\mathbf{Q^n_{i-1}} + \mathbf{Q^n_i}\right)$$

Unstable

Deriving the Finite Volume Scheme

Discrete Conservation Law:

$$\frac{d}{dt}\bar{\mathbf{Q}}_i + \frac{1}{\Delta x}\left[\mathbf{F}^n_{i+1/2} - \mathbf{F}^n_{i-1/2}\right] = 0.$$

Using an explicit first order time discretization (Euler method) we get:

$$\bar{\mathbf{Q}}^{n+1}_i = \bar{\mathbf{Q}}^n_i - \frac{\Delta t}{\Delta x}\left(\mathbf{F}^n_{i+1/2} - \mathbf{F}^n_{i-1/2}\right).$$

We need to define the numerical flux

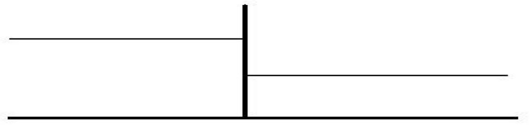
Lax-Friedrichs Flux:

$$\mathbf{F^n_{i-1/2}} = \frac{\mathbf{1}}{\mathbf{2}}\left(\mathbf{Q^n_{i-1}} + \mathbf{Q^n_i}\right) - \frac{\mathbf{\Delta x}}{\mathbf{2\Delta t}}\left(\mathbf{Q^n_i} - \mathbf{Q^n_{i-1}}\right)$$

Very diffusive

Godunov's Method and Riemann Problems

Godunov's method approximates the flux integral in by means of the solution of a Riemann problem, which furnishes an approximation to the integrand.



Riemann problems at each cell interface.

The method computes the numerical flux $\mathbf{F}_{i+1/2}$ from the solution of the local Riemann problem at the cell interface $x = x_{i+1/2}$.

Stability and the CFL Condition

For stability, Godunov's method requires:

$$\frac{c\Delta t}{\Delta x} \leq \frac{1}{2}.$$

This ensures that waves do not interact within a single time step. However, it can be extended to Courant numbers up to 1.

Numerical Flux in Godunov's Method

The numerical flux function is derived as:

$$F_{i-1/2}^n = f\left(q^*(Q_{i-1}^n, Q_i^n)\right),$$

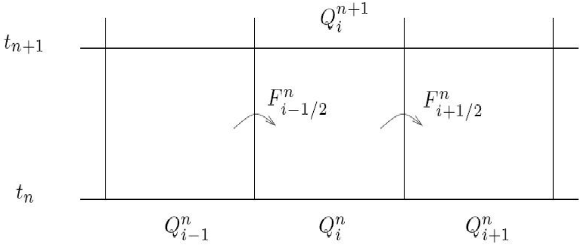
where q^* is the solution of the Riemann problem at the interface.

- Compute q^* by solving the Riemann problem.
- Define the numerical flux using $f(q^*)$.
- Use the flux-differencing formula:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left(F_{i+1/2}^n - F_{i-1/2}^n\right).$$

Spatial descretization

- Finite Volume scheme



$$\frac{d}{dt}\mathbf{Q}_i = -\frac{1}{\Delta x}\left[\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}\right] + \frac{1}{\Delta x}\Delta\mathbf{S}_{b\,i} + \overline{\boldsymbol{\Phi}}$$

$Q_i \approx \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} q(x,t)$ Approximate cell average of the solution

$F_{1+1/2}$: Approximation of the average flux at $x_{i-1/2}$

$C_i = [x_{i-1/2},\ x_{i+1/2}]$: Computational cell

Approximate Riemann solvers

We have to

- Find approximate solutions in the star (intermediate) region of the Riemann problem.
- Choose with in Linear or nonlinear solvers
- Examine whether a linear solver is accompanied with an effective entropy fix to exclude the computation of unphysical entropy-violating shocks.

Approximate Riemann solver of Roe (1981)

Riemann problem of the Linearized system:

$$\mathbf{q}_t + \hat{\mathbf{A}}_{i-1/2}\mathbf{q}_x = 0, \quad \hat{\mathbf{A}}_{\mathbf{i-1/2}} \text{ Roe matrix}$$

$$\hat{\lambda}^1_{i-1/2} = \hat{u}_{i-1/2} - \hat{c}_{i-1/2}, \quad \hat{\lambda}^2_{i-1/2} = \hat{u}_{i-1/2} + \hat{c}_{i-1/2}, \quad \hat{c}_{i-1/2} = \sqrt{g\hat{h}_{i-1/2}}$$

$$\hat{r}_1 = \begin{bmatrix} 1 \\ \hat{u} - \hat{c} \end{bmatrix}, \quad \hat{r}_2 = \begin{bmatrix} 1 \\ \hat{u} + \hat{c} \end{bmatrix}$$

$$\text{mean values:} \quad \hat{h}_{i-1/2} = \frac{h_i + h_{i-1}}{2}, \quad \hat{u}_{i-1/2} = \frac{u_{i-1}\sqrt{gh_{i-1}} + u_i\sqrt{gh_i}}{\sqrt{gh_{i-1}} + \sqrt{gh_i}}$$

Numerical flux:

$$\mathbf{F}_{i+1/2} = \frac{1}{2}\left(\mathbf{F}_{i+1} + \mathbf{F}_i\right) - \frac{1}{2}\sum_{p=1}^{2}\left[\hat{\alpha}_p|\hat{\lambda}_p|\hat{r}_p\right]_{i+1/2}$$

$a1 = ..., a2 = ...$

NOT Entropy satisfying

HLL Approximate Solver

The HLL (Harten, Lax and van Leer) approach assumes estimates S_L and S_R for the smallest and largest signal velocities in the solution of the Riemann problem.
The approximate solution consists of two wave speeds S_L and S_R, separating three states:

$$S_L < S_* < S_R.$$

The solution is given by:

$$q(x,t) = \begin{cases} q_L, & x < S_L t, \\ q^*, & S_L t < x < S_R t, \\ q_R, & x > S_R t. \end{cases}$$

HLL Approximate Solver

The numerical flux for the HLL solver is defined as:

$$F_{HLL} = \begin{cases} F_L, & S_L > 0, \\ \frac{S_R F_L - S_L F_R + S_L S_R (q_R - q_L)}{S_R - S_L}, & S_L \leq 0 \leq S_R, \\ F_R, & S_R < 0. \end{cases}$$

where:

- $F_L = f(q_L)$, $F_R = f(q_R)$ are the fluxes at left and right states.
- S_L and S_R are wave speed estimates, often taken as:

$$S_L = u_L - c_L q_L, \quad S_R = u_R - c_R q_R$$

where

$$q_{LR} = \sqrt{0.5 \frac{(\hat{h} + h_{LR})\hat{h}}{h2_{LR}}} \quad if \hat{h} > h_{LR} \text{ else is } 0.$$

with $\hat{h} = \frac{1}{g} \left(0.5(c_L + c_R) + 0.25(u_L - u_R)\right)^2$

See Toro, Comptational Algorithms for Shallow water Equations, Springer 2024

The DOT Riemann solver

Reminder: If $\mathbf{A}(\mathbf{Q}) = \frac{\partial \mathbf{F}(\mathbf{Q})}{\partial \mathbf{U}}$ we can write:

$$\mathbf{A}(\mathbf{Q}) = \mathbf{R}(\mathbf{Q})\boldsymbol{\Lambda}(\mathbf{Q})\mathbf{R}(\mathbf{Q})^{-1}$$

then $|\boldsymbol{\Lambda}(\mathbf{Q})|$ = diag$(|\lambda_1|, \ldots |\lambda_p|)$, so $|\mathbf{A}(\mathbf{Q})| = \mathbf{R}(\mathbf{Q})|\boldsymbol{\Lambda}(\mathbf{Q})|\mathbf{R}(\mathbf{Q})^{-1}$ but $|\lambda_i(\mathbf{Q})| = \lambda_i^+(\mathbf{Q}) - \lambda_i^-(\mathbf{Q})$ with $\lambda_i^+(\mathbf{Q}) = \max(\lambda_i(\mathbf{Q}), 0)$, $\lambda_i^-(\mathbf{Q}) = \min(\lambda_i(\mathbf{Q}), 0)$ and hence $|\boldsymbol{\Lambda}(\mathbf{Q})| = \Lambda + (\mathbf{Q}) - \Lambda^-(\mathbf{Q})$ which leads to

$$\mathbf{A}(\mathbf{Q}) = \mathbf{A}^+(\mathbf{Q}) + \mathbf{A}^-(\mathbf{Q})$$

We assume that there exist a vector-valued functions $\mathbf{F}^+(\mathbf{Q})$ and $\mathbf{F}^-(\mathbf{Q})$ that satisfy

$$\mathbf{F}(\mathbf{Q}) = \mathbf{F}^+(\mathbf{Q}) + \mathbf{F}^-(\mathbf{Q})$$

The DOT Riemann solver

And

$$\frac{\mathbf{F}^{+}}{\partial \mathbf{Q}} = \mathbf{A}^{+}(\mathbf{Q}), \quad \frac{\mathbf{F}^{+}}{\partial \mathbf{Q}} = \mathbf{A}^{+}(\mathbf{Q})$$

If $\mathbf{Q_L} = \mathbf{Q_i} = \mathbf{Q_0}$ and $\mathbf{Q_R} = \mathbf{Q_{i+1}} = \mathbf{Q_1}$ we write:

$$\mathbf{F}_{i+1/2} = \mathbf{F}^{+}(\mathbf{Q}_0) + \mathbf{F}^{-}(\mathbf{Q}_1)$$

Using the integral relations:

$$\int_{Q_0}^{Q_1} \mathbf{A}^{-}(\mathbf{Q}) d\mathbf{Q} = \mathbf{F}^{-}(\mathbf{Q}_1) - \mathbf{F}^{-}(\mathbf{Q}_0)$$

$$\int_{Q_0}^{Q_1} \mathbf{A}^{+}(\mathbf{Q}) d\mathbf{Q} = \mathbf{F}^{+}(\mathbf{Q}_1) - \mathbf{F}^{+}(\mathbf{Q}_0)$$

so we can write:

The DOT Riemann solver

And

$$\frac{\partial \mathbf{F}^{+}}{\partial \mathbf{Q}} = \mathbf{A}^{+}(\mathbf{Q}), \quad \frac{\partial \mathbf{F}^{-}}{\partial \mathbf{Q}} = \mathbf{A}^{-}(\mathbf{Q})$$

If $\mathbf{Q_L} = \mathbf{Q_i} = \mathbf{Q_0}$ and $\mathbf{Q_R} = \mathbf{Q_{i+1}} = \mathbf{Q_1}$ we write:

$$\mathbf{F}_{i+1/2} = \mathbf{F}^{+}(\mathbf{Q}_0) + \mathbf{F}^{-}(\mathbf{Q}_1)$$

Using the integral relations:

$$\int_{Q_0}^{Q_1} \mathbf{A}^{-}(\mathbf{Q}) d\mathbf{Q} = \mathbf{F}^{-}(\mathbf{Q}_1) - \mathbf{F}^{-}(\mathbf{Q}_0)$$

$$\int_{Q_0}^{Q_1} \mathbf{A}^{+}(\mathbf{Q}) d\mathbf{Q} = \mathbf{F}^{+}(\mathbf{Q}_1) - \mathbf{F}^{+}(\mathbf{Q}_0)$$

so we can write:

The DOT Riemann solver

$$\mathbf{F}_{i+1/2} = \mathbf{F}(\mathbf{Q}_0) + \int_{Q_0}^{Q_1} \mathbf{A}^{-}(\mathbf{Q})d\mathbf{Q} = \mathbf{F}(\mathbf{Q}_1) - \int_{Q_0}^{Q_1} \mathbf{A}^{+}(\mathbf{Q})d\mathbf{Q}$$

or

$$\mathbf{F}_{i+1/2} = \left[\frac{1}{2}\mathbf{F}(\mathbf{Q}_0) + \mathbf{F}(\mathbf{Q}_1)\right] - \frac{1}{2}\int_{Q_0}^{Q_1} |\mathbf{A}(\mathbf{Q})|d\mathbf{Q}$$

The flux requires the evaluation of an integral in phase space, which depends on the chosen integration path joining Q_0 to Q_1. Dumbser and

Toro made two simple but effective suggestions: (i) choose any path, without considerations regarding computational tractability of the scheme; (ii) evaluate matrices by numerical integration in phase space.

The DOT Riemann solver

The simplest path is

$$\varphi(s; , \mathbf{Q_0}, \mathbf{Q_1}) = Q_0 + s(\mathbf{Q_1} - \mathbf{Q_0}), s \in [0,\ 1].$$

There, under a change of variables we obtain:

$$\mathbf{F}_{i+1/2} = \left[\frac{1}{2}\mathbf{F}(\mathbf{Q}_0) + \mathbf{F}(\mathbf{Q}_1)\right] - \frac{1}{2}\left(\int_0^1 |\mathbf{A}(\varphi(\mathbf{s}; \mathbf{Q_0}, \mathbf{Q_1}))| ds\right)(\mathbf{Q_1} - \mathbf{Q_0})$$

Other Approximate Solvers

- The HLLC approximate Riemann solver
- Rusanov flux -Lax Friedrichs flux

See Toro, Comptational Algorithms for Shallow water Equations, Springer 2024

Higher order schemes

Lax-Wendroff Method for a linear sysytem: Taylor Series Expansion

$$q(x, t^{n+1}) = q(x, t^n) + \Delta t q_t + \frac{(\Delta t)^2}{2} q_{tt} + \mathcal{O}(\Delta t^3).$$

Using the equation $q_t = -Aq_x$ and differentiating again:

$$q_{tt} = A^2 q_{xx}.$$

Substituting these terms into the Taylor series gives:

$$q(x, t^{n+1}) = q(x, t^n) - \Delta t A q_x + \frac{(\Delta t)^2}{2} A^2 q_{xx}.$$

Lax-Wendroff Finite Difference Formulation

Approximating derivatives with centered finite differences:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{2\Delta x} A(Q_{i+1}^n - Q_{i-1}^n) + \frac{(\Delta t)^2}{2(\Delta x)^2} A^2 (Q_{i-1}^n - 2Q_i^n + Q_{i+1}^n).$$

This method achieves second-order accuracy but introduces numerical oscillations near discontinuities.

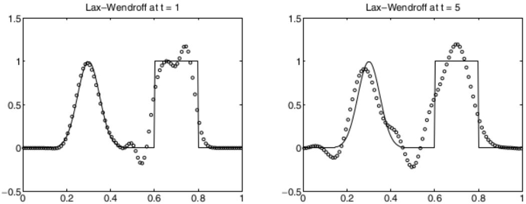
Second order in time and space

Lax-Wendroff as a Finite Volume Method

Lax-Wendroff can be reinterpreted as a finite volume method with a modified numerical flux:

$$F^n_{i-1/2} = \frac{1}{2}\left[AQ^n_{i-1} + AQ^n_i\right] - \frac{\Delta t}{2\Delta x}A^2(Q^n_i - Q^n_{i-1}).$$

This flux formulation ensures second-order accuracy by correcting the standard upwind flux with a diffusion term.



Advection equation using the LW flux. Picture taken from the book Finite Volume Methods for Hyperbolic Problems, LeVeque, 2001

Higher order schemes

We want schemes that will:

1. Provide at least second order accuracy in smooth regions.
2. Reduces numerical diffusion compared to Godunov's method.
3. Preserves sharp discontinuities without oscillations

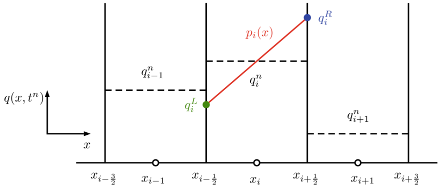
MUSCL Scheme: Piecewise Linear Reconstruction, Vanleer 1979

Instead of piecewise constant values, MUSCL reconstructs a **piecewise linear approximation**:

$$\tilde{q}^n(x) = Q_i^n + \sigma_i^n (x - x_i), \quad x \in C_i.$$

where σ_i^n is the slope in the i-th cell, computed using various methods.

Key Idea: Instead of solving the Riemann problem with cell averages, we use reconstructed values at cell interfaces.



Polynomial reconstruction. Picture taken from Toro, 2024

Second order in space

Slope Computation in MUSCL

The choice of slope σ_i^n affects accuracy and stability. Three common choices:

- **Centered slope (Fromm's method):**

$$\sigma_i^n = \frac{Q_{i+1}^n - Q_{i-1}^n}{2\Delta x}.$$

- **Upwind slope (Beam-Warming method):**

$$\sigma_i^n = \frac{Q_i^n - Q_{i-1}^n}{\Delta x}.$$

- **Downwind slope (Lax-Wendroff method):**

$$\sigma_i^n = \frac{Q_{i+1}^n - Q_i^n}{\Delta x}.$$

Note: Unrestricted linear slopes can introduce oscillations near discontinuities!

MUSCL-Based Finite Volume Update

Using the reconstructed states, the finite volume update equation is:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left(F_{i+1/2} - F_{i-1/2}\right),$$

where the flux at the interface is computed using:

$$F_{i+1/2} = f(Q^L_{i+1/2}, Q^R_{i+1/2}),$$

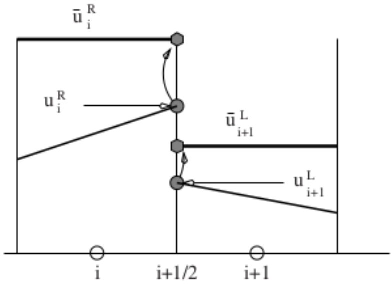
and $Q^L_{i+1/2}, Q^R_{i+1/2}$ are the left and right states obtained from the MUSCL reconstruction.

MUSCL-Hancock Method

1. Data reconstruction as in MUSCL scheme
2. Evolution of q_L and q_R by a time $\frac{1}{2}\Delta t$ according to

$$\hat{q}_i^{L,R} = q_i^{L,R} + \frac{\Delta t}{2\Delta x}\left[f(q_i^L) - f(q_i^R)\right]$$

3. Solution of the Riemann problem (exact or approximate) using $\hat{q}_i{}^R$ if $x < 0$ and $\hat{q}_i{}^L$ if $x > 0$.



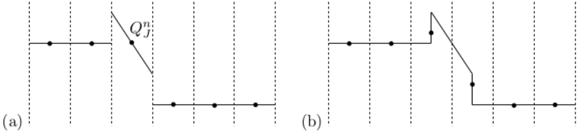
Polynomial reconstruction. Picture taken from Toro, Riemann solvers and numerical methods for fluid dynamics,Springer 2009

Some Other Methods

- K-reconstruction with the successive correction method
- ENO and WENO methods
- The Weighted Average Flux, WAF Schemes
- Centered TVD Schemes: The SLIC and FLIC Method
- ADER scheme

Oscillations

The slopes proposed before were based on the assumption that the solution is smooth. Near a discontinuity there is no reason to believe that introducing this slope will improve the accuracy.



Grid values using Lax-Wendroff slopes. Picture taken from LeVeque, Finite Volume Methods for Hyperbolic Problems, 2001

We want a second order accuracy in smooth regions and less near discontinuities! Slope limiters method

Total Variation

How much should we limit the slope? A way measure oscillations in the solution is the *total variation* of a function.

$$\mathrm{TVD(Q)} = \sum_{i=-\inf}^{\inf} |Q_i - Q_{i-1}|$$

For a continuous function $q(x)$:

$$TV(q) = \sup \sum_{j=1}^{N} |q(\xi_j) - q(\xi_{j-1})|,$$

where the supremum is taken over all subdivisions of the real line. If $q(x)$ is differentiable:

$$TV(q) = \int_{-\infty}^{\infty} |q'(x)| dx.$$

Why Does Total Variation Matter?

The exact solution to the advection equation preserves the total variation:

$$TV(q(\cdot, t)) = \text{constant}.$$

Numerical methods may introduce oscillations, increasing total variation. To avoid this, we require:

$$TV(Q^{n+1}) \leq TV(Q^n).$$

A method satisfying this condition is called **Total Variation Diminishing (TVD)**.

Definition of a TVD Method

Definition: A two-level method is **TVD** if it satisfies:

$$TV(Q^{n+1}) \leq TV(Q^n),$$

for any initial data Q^n.

Properties of TVD Methods:

- Prevents the formation of new extrema (oscillations).
- Ensures stability in the presence of shocks.
- Improves numerical robustness without excessive diffusion.

Monotonicity-Preserving Schemes

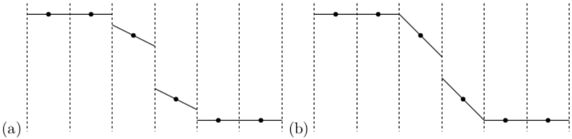
A scheme is **monotonicity-preserving** if:

$$Q_i^n \geq Q_{i+1}^n \quad \forall i \Rightarrow Q_i^{n+1} \geq Q_{i+1}^{n+1} \quad \forall i.$$

Key Observation: Every TVD scheme is monotonicity-preserving, but not every monotonicity-preserving scheme is TVD.
This property ensures that a numerical solution does not develop oscillations near discontinuities.

MUSCL Schemes: Limited Slopes

- Unrestricted linear slopes may lead to spurious oscillations.
- Slope limiters selectively reduce slopes to prevent oscillations.
- A good limiter should:
  - Maintain second-order accuracy in smooth regions.
  - Ensure Total Variation Diminishing (TVD) properties.
  - Prevent nonphysical oscillations.



Grid values using limited slopes. Picture taken from LeVeque, Finite Volume Methods for Hyperbolic Problems, 2001

Minmod Limiter: A Simple TVD Approach

The minmod limiter chooses the least steep slope to avoid oscillations:

$$\sigma_i^n = \text{minmod}\left(\frac{Q_{i+1}^n - Q_i^n}{\Delta x}, \frac{Q_i^n - Q_{i-1}^n}{\Delta x}\right).$$

The **minmod function** is defined as:

$$\text{minmod}(a, b) = \begin{cases} a, & \text{if } |a| < |b| \text{ and } ab > 0, \\ b, & \text{if } |b| < |a| \text{ and } ab > 0, \\ 0, & \text{if } ab \leq 0. \end{cases}$$

Key Properties:

- Simple and strictly TVD.
- Eliminates oscillations completely.
- Can be overly diffusive near sharp gradients.

Superbee Limiter: Sharp Shock Capturing

The Superbee limiter enhances resolution of steep gradients by selecting the steepest allowable slope:

$$
\begin{aligned}
\sigma_i^n &= \operatorname{maxmod}(s_1, s_2) \\
s_1 &= \operatorname{minmod}\left(2\frac{Q_{i+1}^n - Q_i^n}{\Delta x}, \frac{Q_i^n - Q_{i-1}^n}{\Delta x}\right), \\
s_2 &= \operatorname{minmod}\left(\frac{Q_{i+1}^n - Q_i^n}{\Delta x}, 2\frac{Q_i^n - Q_{i-1}^n}{\Delta x}\right)
\end{aligned}
$$

Key Properties:

- Provides sharper resolution of discontinuities.
- Tends to steepen smooth regions excessively.
- May introduce slight numerical instability in some cases.

Monotonized Central (MC) Limiter: Balanced Performance

The Monotonized Central (MC) limiter balances smoothness and sharpness:

$$\sigma_i^n = \text{minmod}\left(\frac{Q_{i+1}^n - Q_{i-1}^n}{2\Delta x}, 2\frac{Q_{i+1}^n - Q_i^n}{\Delta x}, 2\frac{Q_i^n - Q_{i-1}^n}{\Delta x}\right).$$

Key Properties:

- Less diffusive than minmod.
- Avoids excessive steepening like Superbee.
- Good overall choice for many applications.

MUSCL Schemes: Slope limiters (a second approach)

We will use slopes to mean differences such as the following

$$\sigma^n_{i-1/2} = Q^n_i - Q^n_{i-1}, \quad \sigma^n_{i+1/2} = Q^n_{i+1} - Q^n_i$$

with an average

$$\sigma^n_i = 0.5 * (\sigma_{i+1/2} + \sigma_{i-1/2}) = Q^n_{i+1} - Q^n_{i-1}$$

An other approach is to find a slope limiter ξ_i such that

$$\bar{\sigma}_i = \xi_i \sigma_i$$

This approach leads to a TVD region of $\xi(r)$

$$\xi(r) = 0 \text{ for } r \leq 0, 0 \leq \xi(r) \leq \min\left(\xi_L(r), \xi_R(r)\right) \text{ for } r > 0 \text{ and } r = \frac{\sigma_{i-1/2}}{\sigma_{i+1/2}}$$

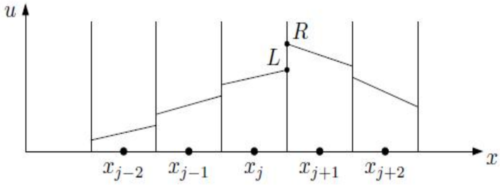
MUSCL Schemes: Slope limiters (a second approach)

A minmode type slope limiter is:

$$\xi(r) = \begin{cases} 0, & \text{if } r \leq 0 \\ r, & \text{if } 0 \leq r \leq 1 \\ \min(1, 1/r) & \text{if } r \geq 1 \end{cases}$$

A third order scheme

- Third order reconstruction using the MUSCL scheme
  If we want more than first order accuracy we have to reconstruct the values across each face



$$
\begin{aligned}
u^L_{i+1/2} &= u_i + \frac{\phi(r_i)}{4}\left[(1-k)\delta u_{i-1/2} + (1+k)\delta u_{i+1/2}\right] \\
u^R_{i+1/2} &= u_{i+1} - \frac{\phi(r_i+1)}{4}\left[(1-k)\delta u_{i+3/2} + (1+k)\delta u_{i+1/2}\right]
\end{aligned}
$$

where $\phi(r)$ is a limiter function.

Problem Statement

- **Explicit methods**
- The Runge-Kutta 2nd-order method, also known as the Midpoint Method, is a numerical technique to solve first-order ordinary differential equations (ODEs).
- It improves upon Euler's method by considering the slope at the midpoint of the interval, leading to better accuracy.

Problem Statement

- **Explicit methods**
- The Runge-Kutta 2nd-order method, also known as the Midpoint Method, is a numerical technique to solve first-order ordinary differential equations (ODEs).
- It improves upon Euler's method by considering the slope at the midpoint of the interval, leading to better accuracy.
- Consider the first-order ODE:
$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0$$
- Our goal is to approximate the solution $y(t)$ at discrete points using the Runge-Kutta 2nd-order method.

Euler's Method Recap

- Euler's method estimates y at $t_{i+1} = t_i + h$ using:

  $$y_{i+1} = y_i + hf(t_i, y_i)$$

  where $y(t_0) = y_0$ and $h = t_{i+1} - t_i$

Euler's Method Recap

- Euler's method estimates y at $t_{i+1} = t_i + h$ using:

  $$y_{i+1} = y_i + hf(t_i, y_i)$$

  where $y(t_0) = y_0$ and $h = t_{i+1} - t_i$

Euler's method from Taylor series:

$$y_{t+1} = y_i + \frac{dy}{dt}|_{t_i,y_i}(t_{i+1} - t_i) + \frac{1}{2!}\frac{d^2y}{dt^2}|_{t_i,y_i}(t_{i+1} - t_i)^2 + \frac{1}{3!}\frac{d^3y}{dt^3}|_{t_i,y_i}(t_{i+1} - t_i)^3 + ...$$

since

$$\frac{dy}{dt} = f(t, y)$$

so we write:

$$y_{i+1} = y_i + f(t_i, y_i)h + O(h^2)$$

Improving Accuracy: RK2

We include more terms from the Taylor expansion:

$$y_{i+1} = y_i + \frac{dy}{dt}|_{t_i,y_i}(t_{i+1} - t_i) + \frac{1}{2!}\frac{d^2y}{dt^2}|_{t_i,y_i}(t_{i+1} - t_i)^2 + O(h^3)$$

or

$$y_{i+1} = y_i + f(t_i, y_i)h + \boxed{f'(t_i, y_i)}h^2$$

We can write the above as:

$$y_{i+1} = y_i + (\beta_1 k_1 + \beta_2 k_2)h$$

- k_1 first slope estimate , $k_1 = f(t_i, y_i)$
- second slope estimate, $k_2 = f(t_i + \alpha h, y_i + \alpha h k_1)$

Improving Accuracy: RK2

Find the parameters β_1, β_2, α through the Taylor expansions. We can write:

$$f(t_i + \alpha h + y_i + \alpha h k_1) \approx f(t_i, y_i) + \alpha h \frac{d}{dt} f(t_i, y_i)$$

and substituting in

$$y_{i+1} = y_i + (\beta_1 k_1 + \beta_2 k_2) h$$

We get:

$$y_{i+1} = y_i + h(\beta_1 f_i + \beta_2 f_i) + h^2 \beta_2 \alpha f_i' + O(h^3) \tag{3}$$

And we want to much the exact Taylor expansion so it has to: $\beta_1 + \beta_2 = 1$ and $\beta_2 \alpha = \frac{1}{2}$
For $\alpha = 0.5$ and $\beta_1 = 0, \quad \beta_2 = 1$: Midpoint Method

Improving Accuracy: RK2

Following our notation:

$$
\begin{aligned}
k^1 &= \mathcal{L}(t^n, Qy^n) \\
k^2 &= \mathcal{L}(t^n + 0.5h, Q^n + 0.5hk_1) \\
Q^{n+1} &= Q^n + hk^2
\end{aligned}
$$

- Stability region: a disk of radius 1 centered at 0.
- Maximum CFL number c=1.

SSP RK2

The idea behind SSP methods is to assume that the first order forward Euler time discretization of the method of lines ODE is strongly stable under a certain norm, when the time step Δt is suitably restricted, and then try to find a higher order time discretization (Runge–Kutta or multi step) that maintains strong stability for the same norm, perhaps under a different time step restriction[1]

[1] Gottlieb, On High Order Strong Stability Preserving Runge-Kutta and Multi Step Time Discretizations, Journal of Scientific Computing,25,2005

SSP RK2

SSP Runge Kutta (RK2)

1. First predictor

$$\mathbf{Q}^1 = \mathbf{Q}^n + \Delta t \mathcal{L}(\mathbf{Q}^n)$$

2. Final Stage

$$\mathbf{Q}^{n+1} = \frac{1}{2}\mathbf{Q}^n + \frac{1}{2}\mathbf{Q}^1 + \frac{\Delta t}{2}\mathcal{L}(\mathbf{Q}^1)$$

- Maximum CFL number c=1.

SSP RK3

SSP Runge Kutta (RK3)

1. First predictor

$$\mathbf{Q}^1 = \mathbf{Q}^n + \Delta t \mathcal{L}(\mathbf{Q}^n)$$

2. Second Predictor

$$\mathbf{Q}^2 = \frac{3}{4}\mathbf{Q}^n + \frac{1}{4}\mathbf{Q}^1 + \frac{\Delta t}{4}\mathcal{L}(\mathbf{Q}^n)$$

3. Final Stage

$$\mathbf{Q}^{n+1} = \frac{1}{3}\mathbf{Q}^n + \frac{2}{3}\mathbf{Q}^2 + \frac{2\Delta t}{3}\mathcal{L}(\mathbf{Q}^n)$$

- Maximum CFL number c=1.