A Community-Driven Collection of Approximate Riemann Solvers for Hyperbolic Problems

Mauricio J. Del Razo†, David Ketcheson‡ and Randall J. LeVeque†

†Department of Applied Mathematics, University of Washington; ‡Computer, Electrical, and Mathematical Sciences Division, KAUST

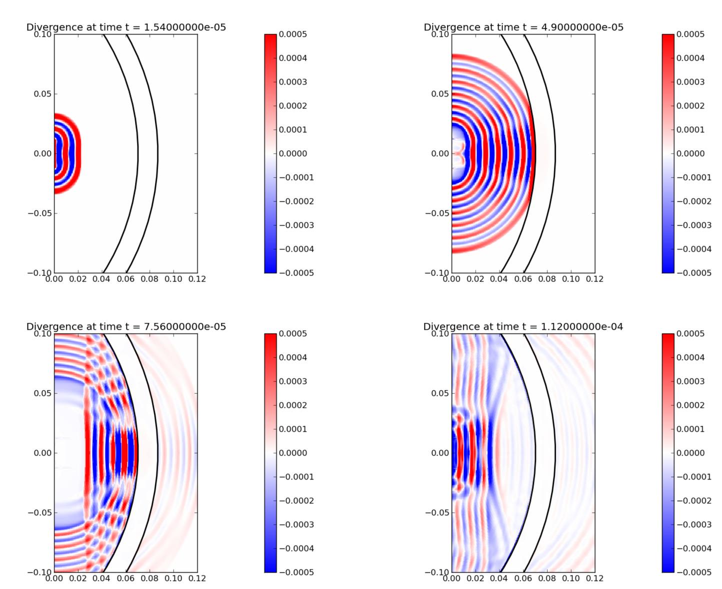
APPLIED MATHEMATICS
UNIVERSITY of WASHINGTON

Motivation

The **key ingredient** in modern numerical methods for **hyperbolic problems** is the **Riemann solver**.

Many of the physical **conservation laws** are framed as hyperbolic problems. Therefore, the same algorithms can be used to solve a wide range of very complex hyperbolic problems in areas like:

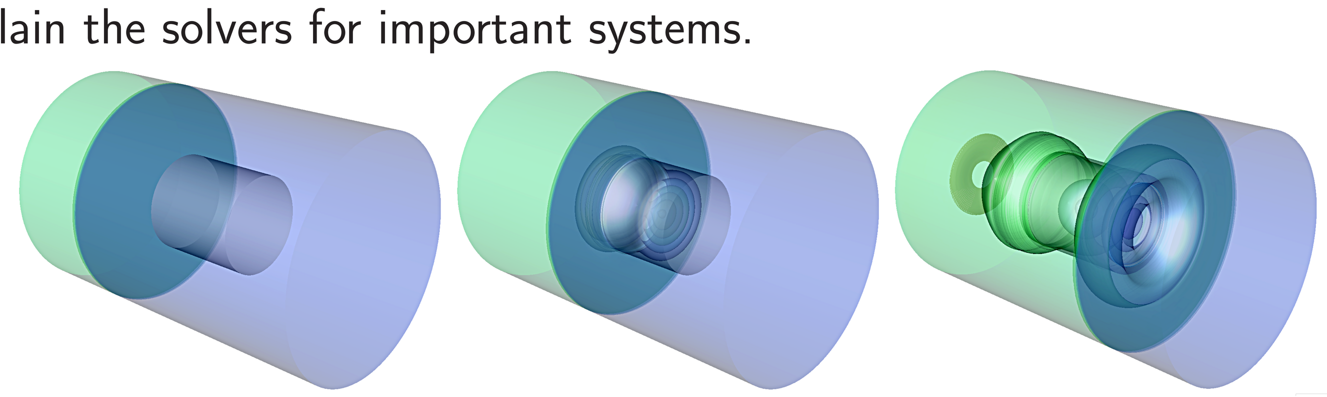
- Water waves,
- Fluid dynamics,
- Elasticity,
- Electromagnetics,
- and many more.



2D evolution of elastic oscillating P-wave hitting a circular interface.

In this work, **we present an effort to**:

- Make available a large **library of Riemann solvers** for general use;
- Develop a set of **IPython notebooks** that describe and **interactively** explain the solvers for important systems.

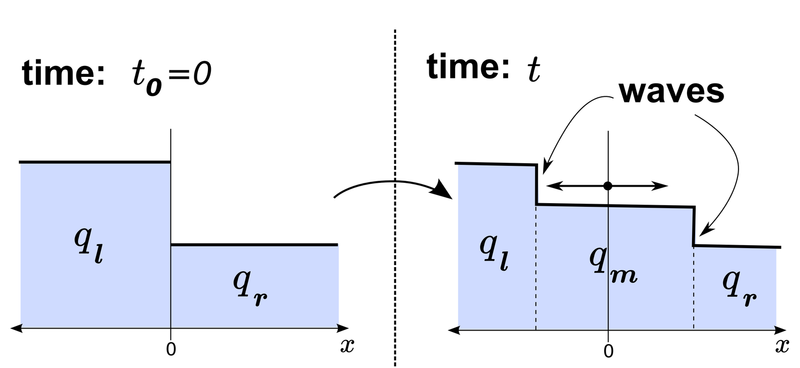


3D evolution of pressure shock wave hitting a cylindrical interface using Euler equations.

What is the Riemann solver?

The Riemann problem

A **one-dimensional Riemann problem** for a system of conservation laws:



A solution of a Riemann problem.

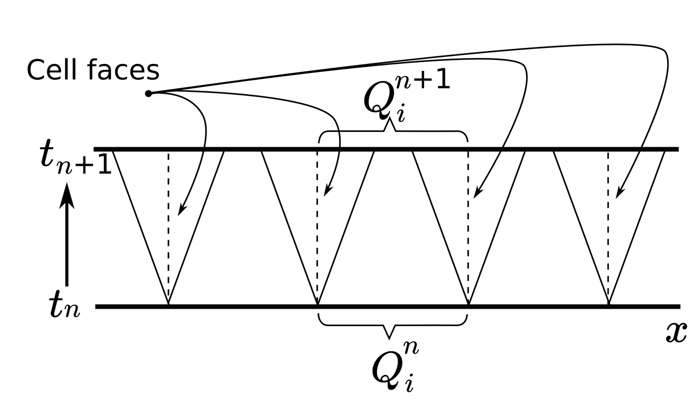
$$q_t + f(q)_x = 0,$$

$$q(x,0) = \begin{cases} q_\ell & \text{if } x < 0, \\ q_r & \text{if } x > 0. \end{cases}$$

Result: waves propagating left and right of discontinuity.

It is fundamental to solve it at each **cell face** of a numerical simulation.

- Discretize space,
- Calculate cells i averages at time t_n: $Q_i^n = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} q(x,t_n)dx$,
- Solve Riemann problem at each face.
- Update cell averages at t_{n+1} with left and right waves ($A^{\pm}\Delta Q$) at each face.

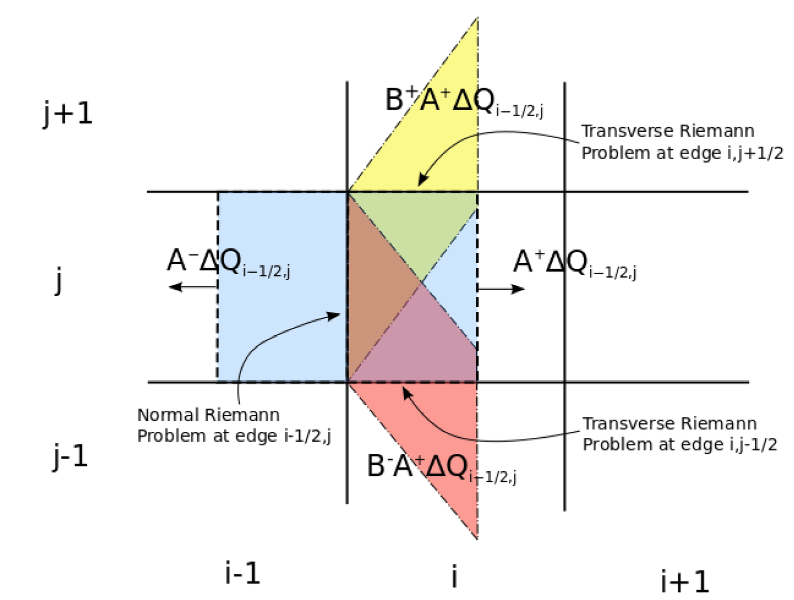


The Riemann problem output is coupled with the **wave propagation algorithms** *–see refs.–* and implemented in Clawpack software.

The Transverse Riemann Problem

What about 2D (and more)?

- Dimensional Splitting (1st order).
  - Uses the normal Riemann solver and swipes in x and y (and z).
  - Higher-order in combination with alternative higher-order methods.
- **Transverse solver** (2nd order).
  - Split the normal waves ($A^{\pm}\Delta Q$) from the original Riemann solver, into transverse waves ($B^{\pm}A^{\pm}\Delta Q$).
  - Employs the **eigen-structure** of the equations for the splitting.



Simple examples:

1D Acoustics Equations

An **exact Riemann solver** for linear acoustic equations in 1D:

$$q_t + \mathrm{A}q_x = 0.$$

Let p be pressure, u velocity ρ_0 density and K_0 the bulk modulus, then

$$q = \begin{bmatrix} p \\ u \end{bmatrix}, \quad \mathrm{A} = \begin{bmatrix} 0 & K_0 \\ 1/\rho_0 & 0 \end{bmatrix},$$

with initial condition

$$q(x,0) = \begin{cases} q_l & \text{if } x \le 0, \\ q_r & \text{if } x > 0. \end{cases}$$

To solve it, we use the eigen-structure to write it as **two uncoupled advection eqs**. The **matrix of eigenvectors** of A is

$$\mathrm{R} = \begin{bmatrix} -Z_0 & Z_0 \\ 1 & 1 \end{bmatrix} = [\bar{r}_l, \bar{r}_r],$$

with impedance $Z_0 = \rho_0 c_0$, **eigenvalues** $s_l = -c_0$ and $s_r = c_0$, and sound speed $c_0 = \sqrt{K_0/\rho_0}$.

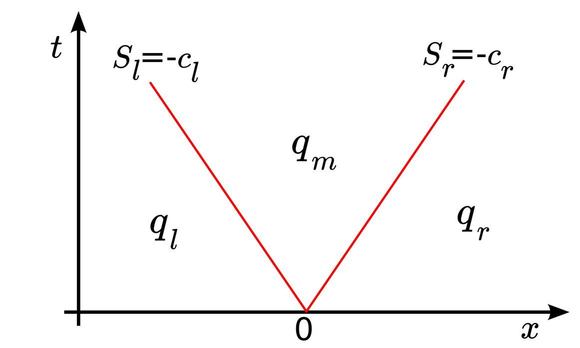
We write the jump in q as a linear combination of the eigenvectors:

$$q_r - q_l = \Delta\bar{q} = \alpha_l \bar{r}_l + \alpha_r \bar{r}_r,$$
$$\Rightarrow \quad \mathrm{R}\bar{\alpha} = \Delta q$$

Solving for α_m, we obtain the solution

$$q_m = q_l + \alpha_l \bar{r}_l = q_r - \alpha_r \bar{r}_r.$$

It has the following structure,

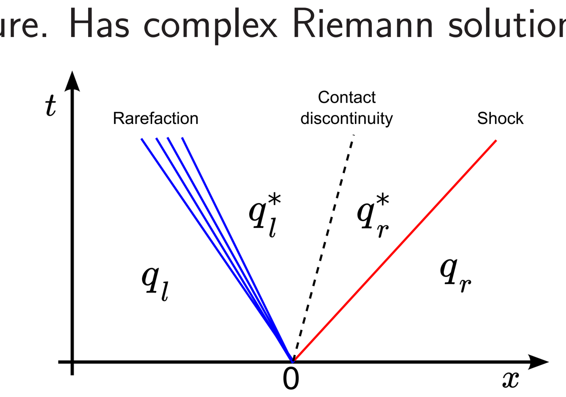


1D Euler Equations

Euler equations are **non-linear** $\Rightarrow$ exact solvers more difficult & expensive.

$$\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{bmatrix} = 0,$$

where ρ is density, u velocity, E energy and p pressure. Has complex Riemann solution:



Two alternative **approximate Riemann-solvers**:

1. **Roe**:
   - Evaluate the Jacobian matrix of the flux at an intermediate state.
   - Proceed as if it was a linear.
2. **HLLC**:
   - Assume two waves and a contact discontinuity.
   - Approximate the wave speeds & calculate the middle states.

Mapped grids? Interfaces?... They get more complicated!

IPython Notebooks & GitHub

How to collect and share them in an educational way?

Collect them & share them with GitHub:

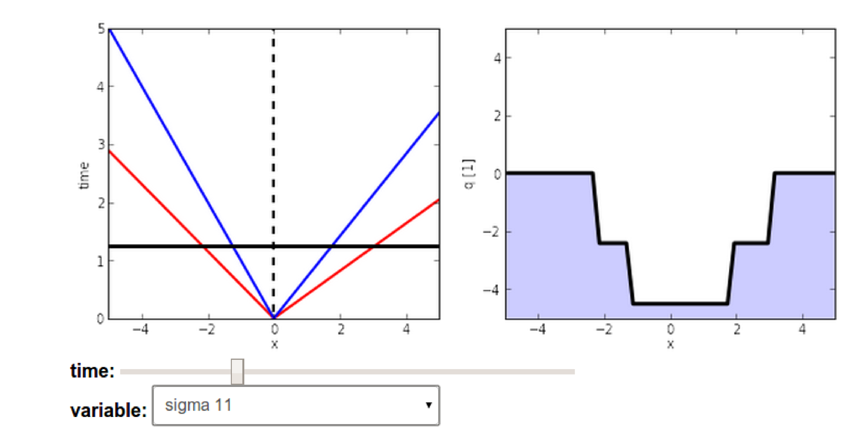
- Repository hosting service
- Version control
- Easy community collaboration
- Open-source available

Current repository: `https://github.com/clawpack/riemann`

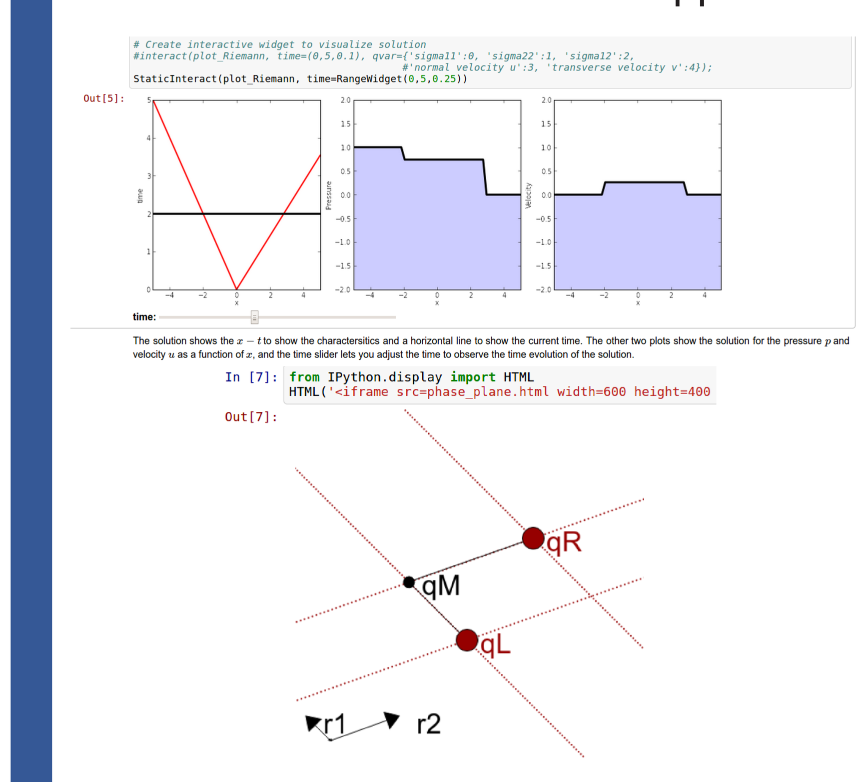
also check: `http://www.clawpack.org/pyclaw/rp.html`

IPython notebooks as an educational tool:

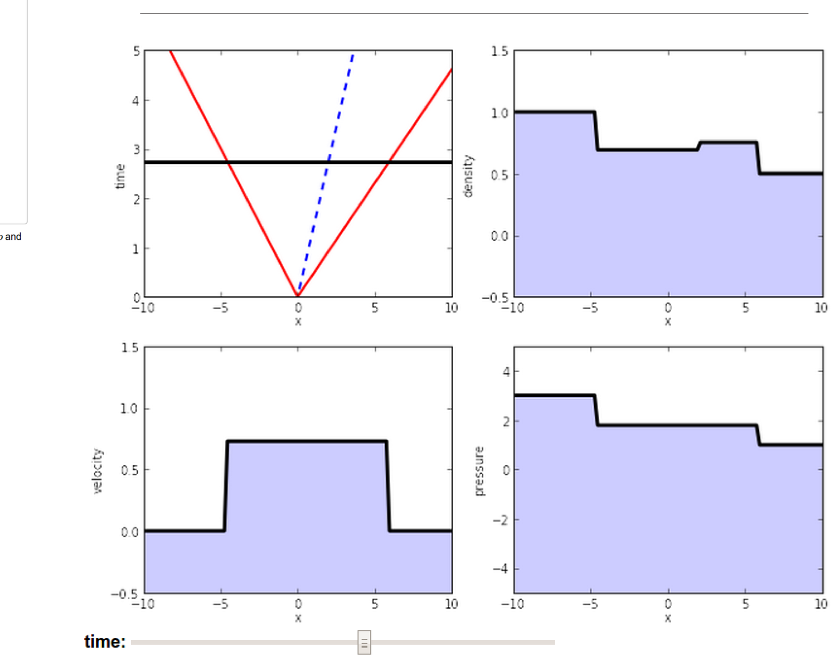
- Web-based interactive interface: **Math + text + code + plots**
- Allows **in-line** execution
- **Interactive** educational apps



Elasticity eqs. iPython notebook.



Acoustic eqs. iPython notebook.



Euler eqs. iPython notebook.

Available online at:

`http://www.clawpack.org/notebooks.html`

Final comments

Riemann solver library

- Made openly available a large library of Riemann solvers.
- Developed for Clawpack software, but scripted for general use.
- Many community members interested in developing it.

iPython notebooks

- Built set of Riemann solvers into iPython notebooks.
- Interactive plotting and coding.
- Math, text and code written educationally.

References:

- LeVeque, Randall J. Finite volume methods for hyperbolic problems. Vol. 31. Cambridge university press, 2002.
- LeVeque, Randall J. "Wave propagation algorithms for multidimensional hyperbolic systems." Journal of Computational Physics 131.2 (1997): 327-353.
- Ketcheson, David I., et al. "PyClaw: Accessible, extensible, scalable tools for wave propagation problems." SIAM Journal on Scientific Computing 34.4 (2012): C210-C231.
- Clawpack Development Team, Clawpack software 5.0, http://www.clawpack.org, 2014

*Supported in part by NSF, the Moore/Sloan Data Science Environment and CONACYT.

CONTACT: Mauricio J. Del Razo <maojrs@uw.edu>