GPU Acceleration of Riemann Solver

YuHsuan Shih[1]

[1]Department of Applied Mathematics
Columbia University

E4302 Methods in Computational Science, 2016

Outline

Outline

Why Riemann solver?

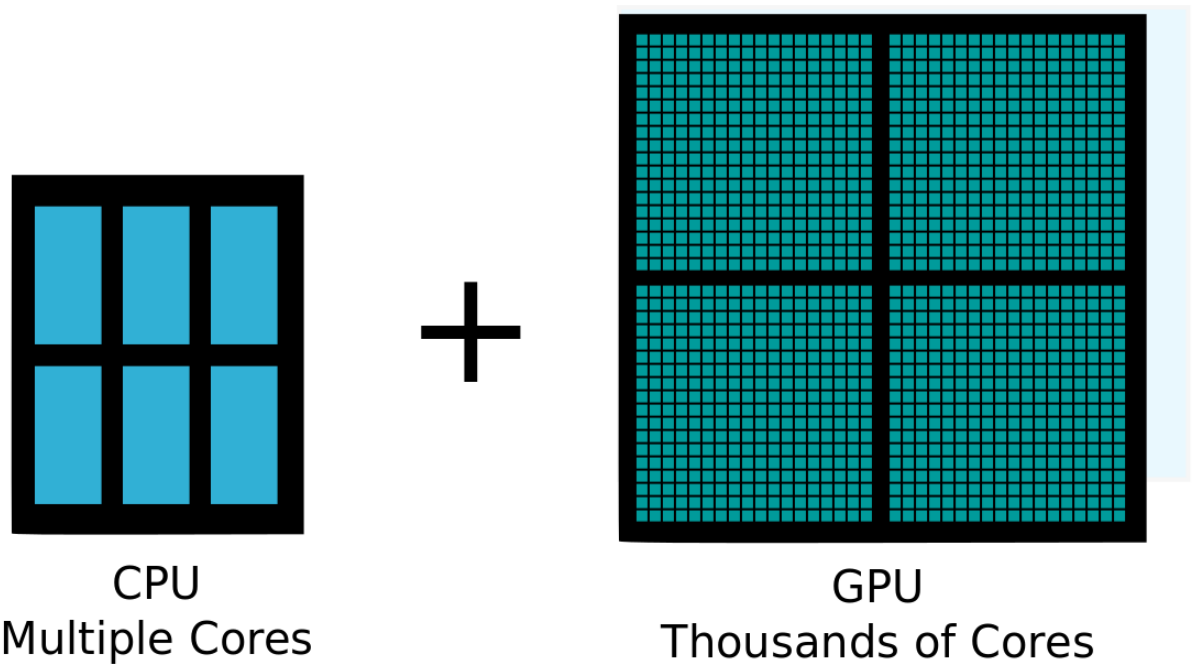
Riemann solver is expensive:

- The high-resolution Godunov methods, widely used finite volume methods for solving conservation law, require solution of **Riemann problem at every cell boundary** for each time step.
- If we discrestize a 2D domain into 100×100 cells, we need to solve 10000 riemann problems per time step!

Outline

Why Graphics Processing Unit (GPU)?

- GPU is a manycore accelerator. It has thousands of cores to process parallel workloads.
- Riemann problems from each cell boundary can be run in parallel.



Source: https://svi.nl/HuygensGPU

Figure: CPU v.s. GPU

Outline

Roe riemann solver for 1D Euler equation on GPU

- 1D Euler equations

$$\begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}_t + \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (E + p)u \end{bmatrix}_x = 0$$

- Numerical Method:
  Godunov type finite volume method with **roe's approximate riemann solver** for 1d euler riemann problem.
- Device: Intel(R) Iris(TM) Graphics 6100
- Programming Language: OpenCL
- Visualization: Visclaw

Outline

Linear acoustic equation

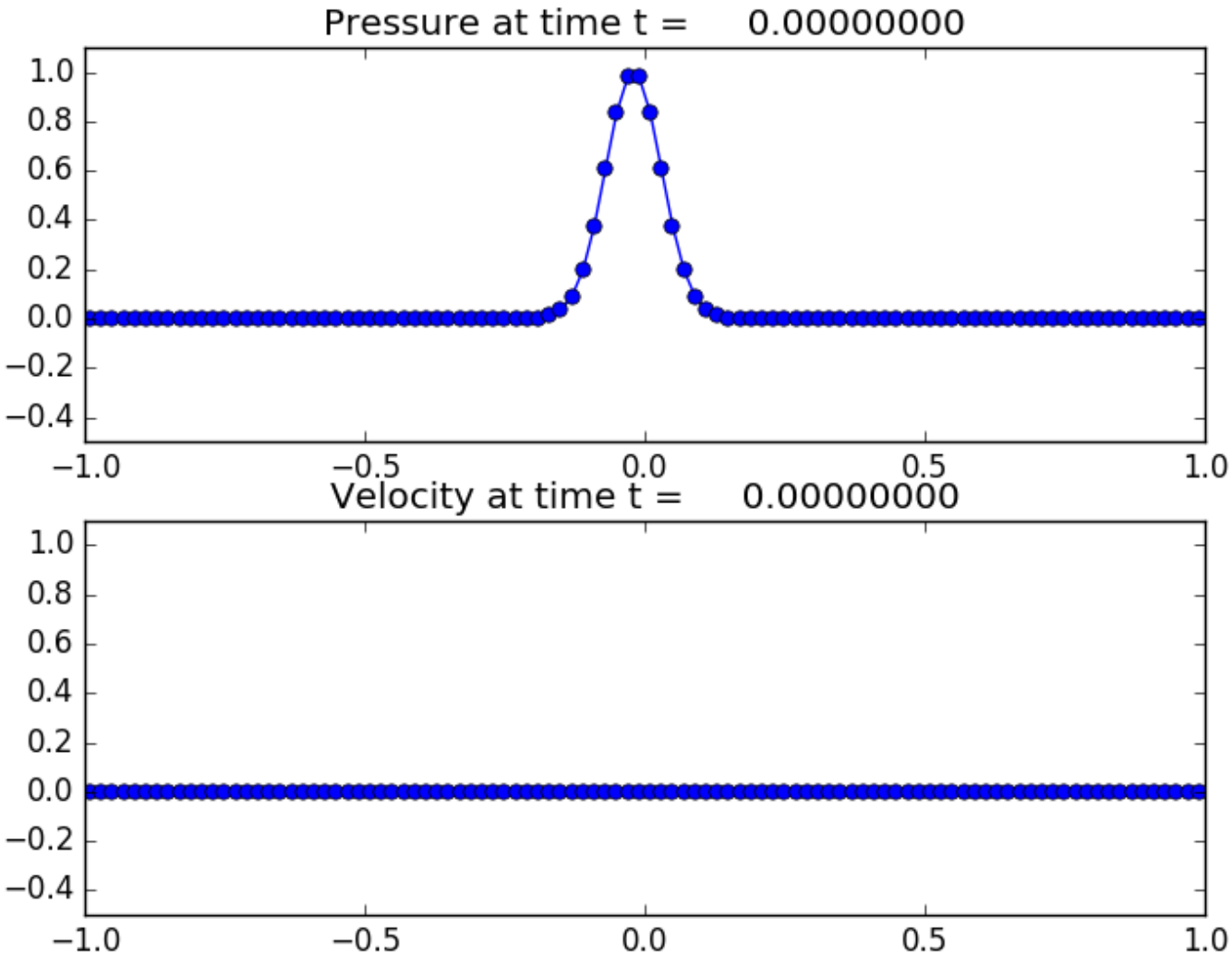
- Equation -

$$\begin{bmatrix} p \\ u \end{bmatrix}_t + \begin{bmatrix} u_0 & K_0 \\ 1/\rho_0 & u_0 \end{bmatrix} \begin{bmatrix} p \\ u \end{bmatrix}_x = 0$$

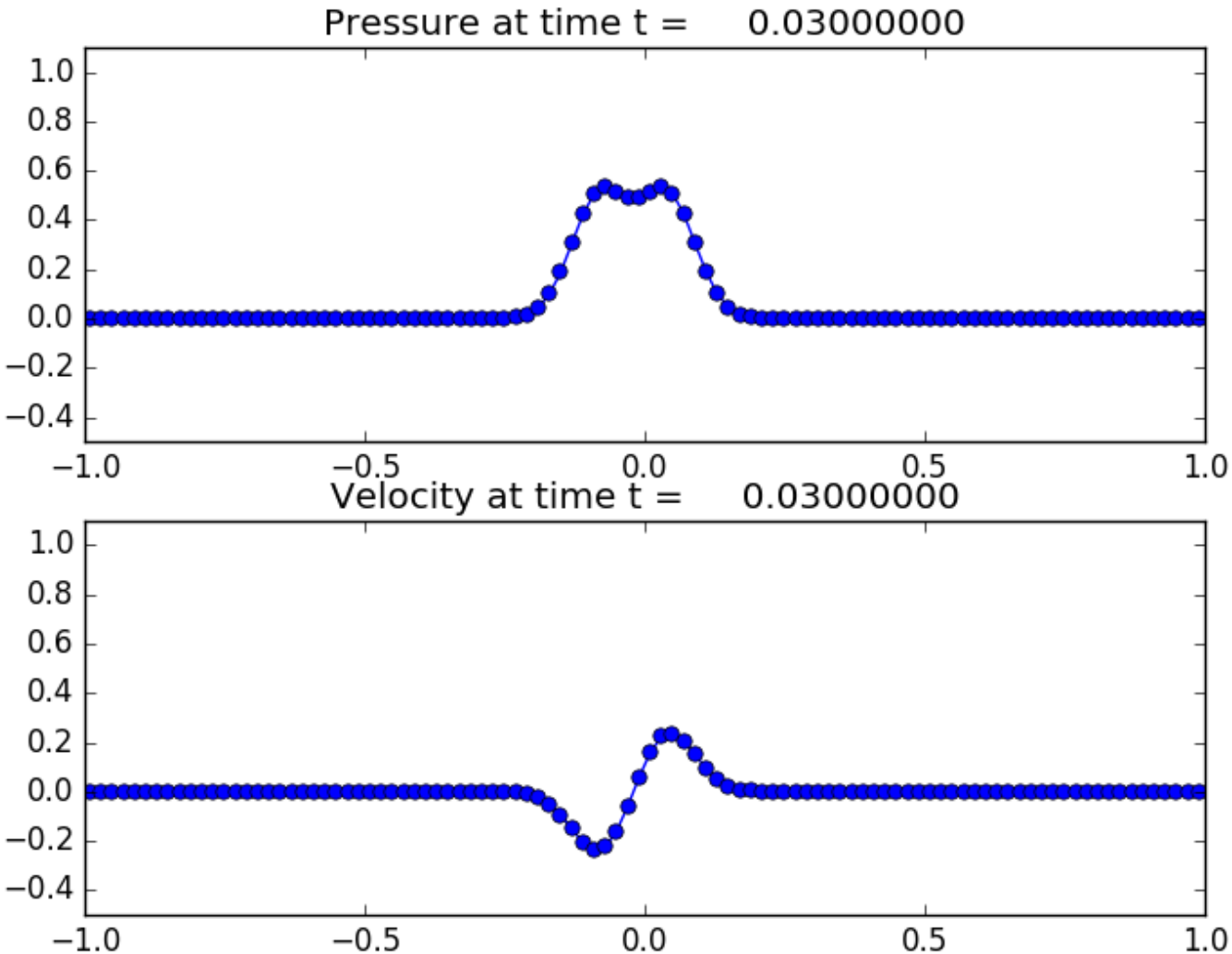
$\longrightarrow$ One right going wave and one leftgoing wave with velocity $\pm\sqrt{K_0/\rho_0}$.

- OpenCL Kernels:
  qinit.cl - Initialize the q array.
  bc1.cl - Fill in ghost cells value for boundary condition.
  acoustic_1d.cl - solve riemann problems, calculate flux between cells and update the cell value.

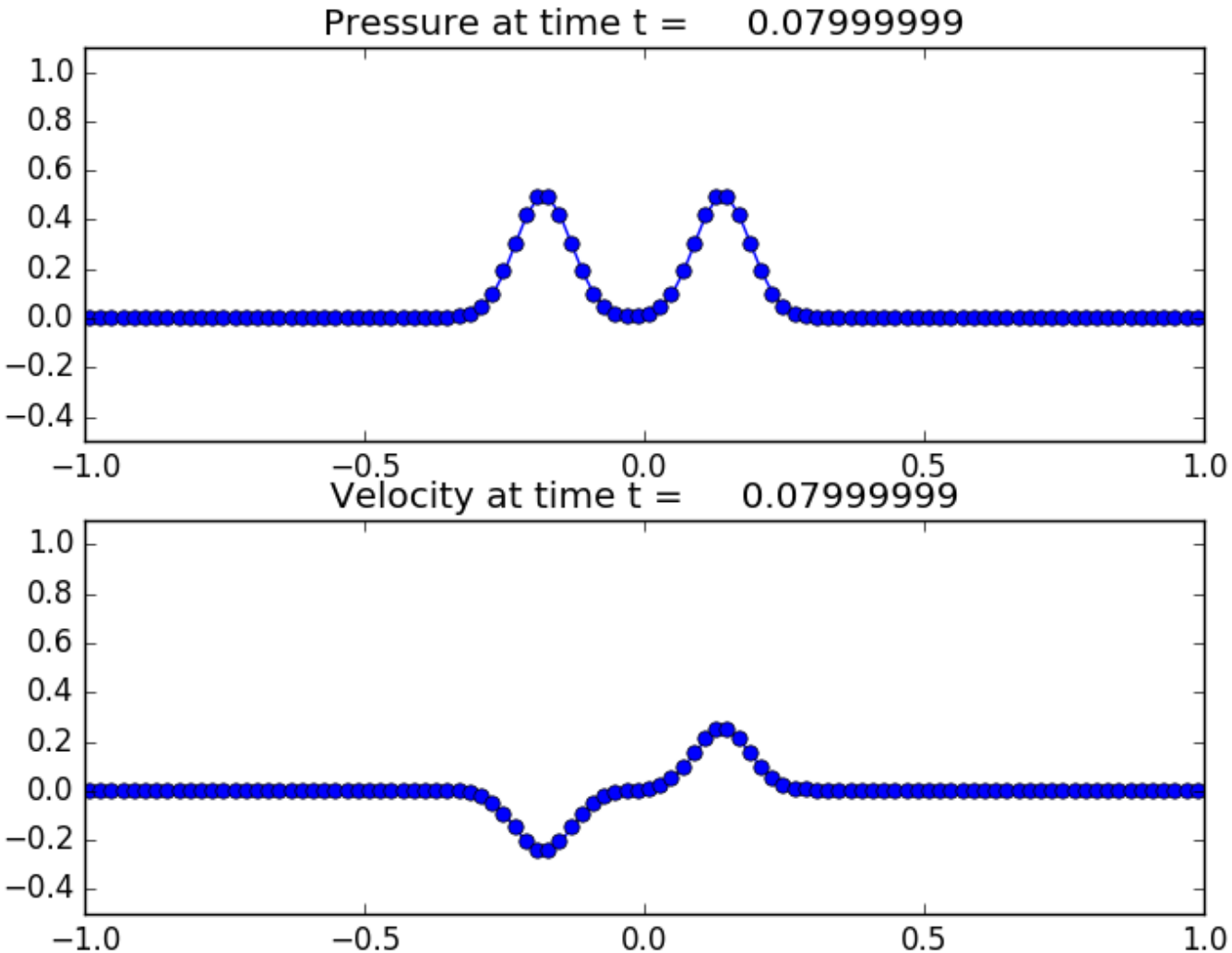
Linear acoustic equation



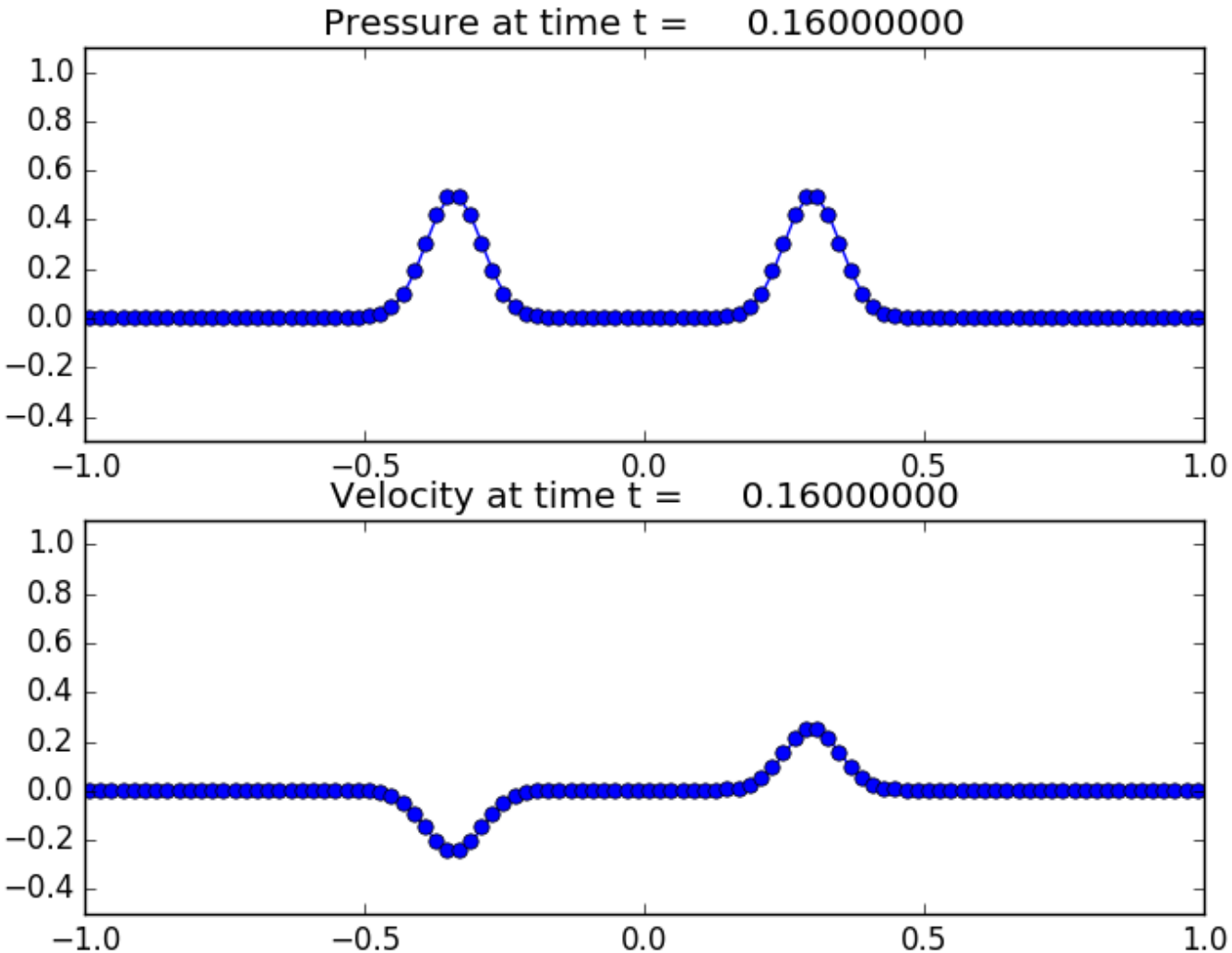
Linear acoustic equation



Linear acoustic equation



Linear acoustic equation



Reference I

Ideas come from [1].


Amal Alghamdi, Aron Ahmadia, David I. Ketcheson, Matthew G. Knepley, Kyle T. Mandli, and Lisandro Dalcin.
Petclaw: A scalable parallel nonlinear wave propagation solver for python.
pages 96–103, 2011.