Kokkos, Modern C++, performance portability, ...

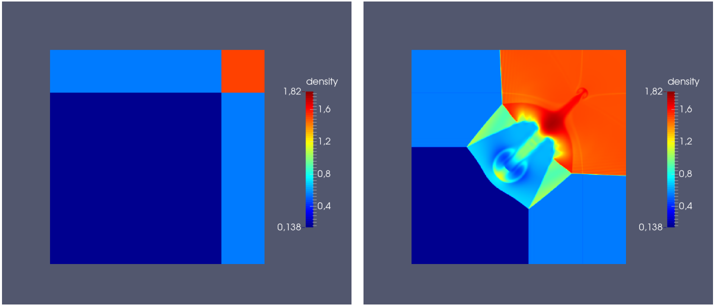
Pierre Kestener[1]

[1]CEA Saclay, DSM, Maison de la Simulation

PATC, January, 16-18th, 2017

Finite volume CFD example in Kokkos

- **Use Kokkos to parallelize a finite volumes solver for CFD**: solver Euler system in 2D/3D, compressible fluid flow, hyperbolique problem
- Example graphics output; temporal evolution of fluid density (initial condition is a *4 quadrants* Riemann problem):



- **Euler equations** (conservation of **mater**, **momentum** and **totla energy**)

$$\frac{\partial \rho}{\partial t} + \nabla.(\rho \mathbf{v}) = 0 \tag{1}$$

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v}.\nabla)\mathbf{v} = -\nabla P \tag{2}$$

$$\frac{\partial E}{\partial t} + \nabla.\left[(E + P_{tot})\mathbf{v}\right] = 0 \tag{3}$$

- **Conservative form** :

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y + \mathbf{H}(\mathbf{U})_z = \mathbf{0}$$

- Conservatives variables and flux:

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}, \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ u(E+p) \end{bmatrix}, \mathbf{G} = \begin{bmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ \rho v w \\ v(E+p) \end{bmatrix}, \mathbf{H} = \begin{bmatrix} \rho w \\ \rho w u \\ \rho w v \\ \rho w^2 + p \\ w(E+p) \end{bmatrix}$$

- **Conservative form** :

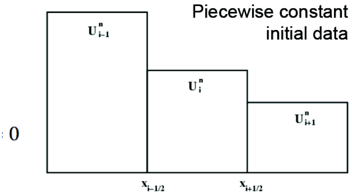
$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + ... = \mathbf{0} \Rightarrow \int_{t_n}^{t_{n+1}} \iiint_{C_{i,j,k}} dt\,\mathbf{dv}\,(\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + ...) = \mathbf{0}$$

- After **time integration** between t_n and t_{n+1} and over a cell volume :

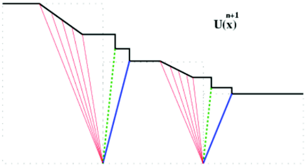
$$\frac{\mathbf{U}^{n+1}_{i,j,k} - \mathbf{U}^{n}_{i,j,k}}{\Delta t} + \frac{\mathbf{F}^{n+1/2}_{i+1/2,j,k} - \mathbf{F}^{n+1/2}_{i-1/2,j,k}}{\Delta x} + ... = 0$$

- **$\mathbf{U}^n_{i,j,k}$ is a volume-averaged quantity at t_n**
- **$\mathbf{F}^{n+1/2}_{i+1/2,j,k}$ is a time-averaged quantity (between t_n and t_{n+1}, explaining index $1/2$) of the surface-average flux at $x = i + 1/2$**
- $E = \rho(\frac{1}{2}\mathbf{V}^2 + e)$ Total volumic energy
- Perfect gas law give the internal energy: $e = \frac{p}{\rho(\gamma-1)}$, with $\gamma = 1.4$ (ex. air at $T = 20^o C$)
- change to non-conservatives variables: $U \Rightarrow W$ with ${}^T W = [\rho, u, v, w, p]$

- $\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{\Delta t}{\Delta x}(\mathbf{F}^{n+1/2}_{i-\frac{1}{2}} - \mathbf{F}^{n+1/2}_{i+\frac{1}{2}})$
- **How to compute/approximate flux $\mathbf{F}^{n+1/2}_{i-\frac{1}{2}}$?**
- **1er ordre Godunov method :**
- anear $x = i + 1/2$, just solve a **Riemann** problem (Euler system with init conditions defined a piecewise constants $\mathbf{U}_i^n$ et $\mathbf{U}_{i+1}^n$:
$\mathbf{U}^*_{i+1/2} = RP(\mathbf{U}_i^n, \mathbf{U}_{i+1}^n)$
- then use flux $\mathbf{F}^{n+1/2}_{i+\frac{1}{2}} = F(\mathbf{U}^*_{i+1/2}(0))$
- Many type of Riemann solvers : Roe, HLL, etc ...



Godunov, S. K. (1959), A Difference Scheme for Numerical Solution of Discontinuos Solution of Hydrodynamic Equations, *Math. Sbornik*, *47*, *271-306*, translated US Joint Publ. Res. Service, JPRS 7226, 1969.

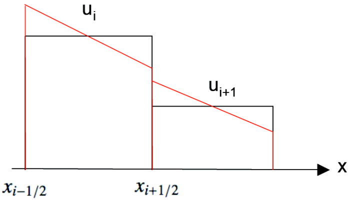


Advection: 1 wave, Euler: 3 waves, MHD: 7 waves

Source: R. Teyssier, 5th JETSET School, amr_lecture1.pdf

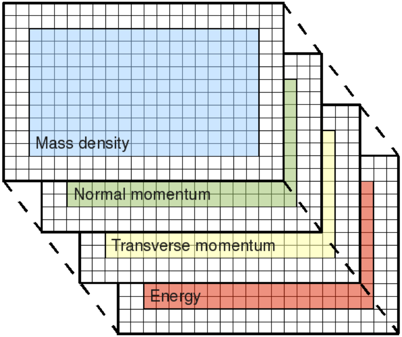
book: *Riemann Solvers And Numerical Methods for Fluid Dynamics: A Practical Introduction*, by E. F. Toro, Springer

MUSCL-Hancock (Monotone Upstream-centered Schemes for Conservation Laws) scheme implemented



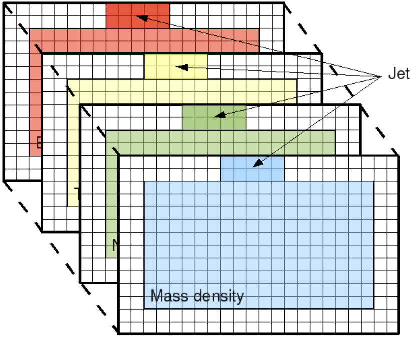
- 2nd order Godunov scheme : $\mathbf{U}_i^n$ replaced by linear picewise functions(predictor-corrector scheme).
- MUSCL-Hancock done in 3 steps :
  - compute slopes Δ_i (using a TVD limiter) and extrapolate $\mathbf{U_i}$ values to cell edges
  - perform 1/2 time step integration of edge values
  - solve Riemann problems at cell interfaces to get fluxes $\mathbf{F_{i+\frac{1}{2}}}$, then update $\mathbf{U_i}$ at next time step

- Four 2D grids:
  - 1 per conservative variable : ρ, ρv_x, ρv_y, e
  - 2d space discretization
  - limit conditions : 2 ghost cells, multiple types
    - reflective
    - absorbing
    - periodic



Simulation

- jet simulation
- parameters
  - run parameters (total time, output rate)
  - geometric parameters (NX, NY, Δx)
  - border types
  - schmeme parameters
  - jet parameters



- How to run ? `./euler2d.cuda ./test.ini`
- Quick output visualization: paraview

Visualisation

- File format is VTK: more precisely VTI (for regular cartesian grids) [1]; see website vtk_formats.simple.html which describes VTK files format variants
- Visualization GUI:
  - paraview
  - Can also use standalone python script `plot_data.py`

[1] vtr format is also possible here.

Additional reference

- See C.P. Dullemond document on CFD numerical methods: http://www.mpia-hd.mpg.de/~dullemon/lectures/fluiddynamics08/