香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Dynamic Discrete Ordinate Method in Solving Boltzmann Equation for Gas Flows

Chin-Tsau HSU

Department of Mechanical and Aerospace Engineering,
The Hong Kong University of Science and Technology

Ka Fai SIN

Management Consulting Division,
Ove Arup & Partners Hong Kong Limited

International Conference on Flow Physics and its Simulation - In Memory of Prof. Jaw-Yen Yang,
3-5 December 2016, Taipei, Taiwan

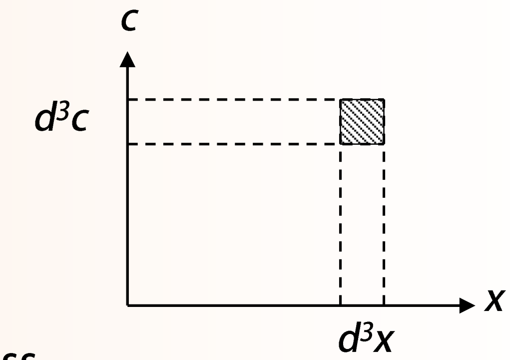
Outline

I. Boltzmann Equation
II. Discrete Ordinate Method (DOM)
III. Dynamic Discrete Ordinate Method (DDOM)
IV. Numerical Results and Validations
V. Massively Parallel Computation
VI. Conclusions

I. Boltzmann Equation

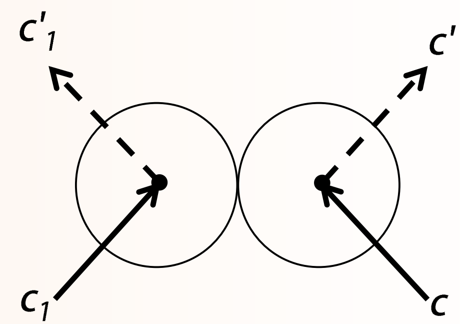
- Transport equation of velocity distribution function (*f*) of number density (*n*)

$$\int_{-\infty}^{+\infty} f\left(\vec{x},\vec{c},t\right) d^3c = n$$



- Assumptions on gas molecules
  - Identical spherical molecules with same mass
  - Dilute gas
  - Elastic collision:

$$\vec{c}+\vec{c}_1=\vec{c}\,'+\vec{c}_1\,' \quad ; \quad \vec{c}^{\,2}+\vec{c}_1^{\,2}=\vec{c}\,'^2+\vec{c}_1\,'^2$$



I. Boltzmann Equation

- Boltzmann equation

$$\frac{\partial f}{\partial t} + \vec{c}\frac{\partial f}{\partial \vec{x}} + \vec{F}\frac{\partial f}{\partial \vec{c}} = \Omega(f)$$

where the collision integral: $\Omega(f) = \int_{-\infty}^{+\infty} \left(f' f_1' - f\, f_1\right) g b\, db\, d\varepsilon\, d^3 c_1$

- Hydrodynamic moments

$$\rho = m\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f d^3 c \qquad \rho\vec{u} = m\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \vec{c} f d^3 c$$

$$\rho e = m\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \frac{1}{2}\left(\vec{c} - \vec{u}\right)^2 f d^3 c$$

($m = 1$ if particle mass is used as a unit scale to measure mass)

Equilibrium State

- Equilibrium velocity distribution function (Maxwellian)

$$f^{eq}(\vec{c}) = \left(\frac{m}{2\pi kT}\right)^{\frac{3}{2}} \exp\left\{-\frac{m}{2kT}\left[(\vec{c}-\vec{u})^2\right]\right\}$$

$$\Omega(f^{eq}) = 0$$

- Therefore, $\Omega = \nu(f^{eq} - f)$ is mathematically exact where the collision frequency is a function of *n* and *c*.

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Boltzmann Equation with BGK Model

- Bhatnagar-Gross-Krook (BGK) collision model:

$$\frac{\partial f}{\partial t}+\vec{c}\,\frac{\partial f}{\partial \vec{x}}+\vec{F}\,\frac{\partial f}{\partial \vec{c}}=\nu\left(f^{eq}-f\right)$$

$$f^{eq}\left(\vec{c}\right)=\left(\frac{m}{2\pi kT}\right)^{\frac{3}{2}}\exp\left[-\frac{m}{2kT}\left(\vec{c}-\vec{u}\right)^{2}\right]$$

- Coupling with hydrodynamic moments

$$Q_i=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\psi_i f d^3c \quad ; \quad Q_i=\left(\rho,\rho\vec{u},e\right) \quad ; \quad \psi_i=\left[1,\vec{c},\frac{1}{2}\left(\vec{c}-\vec{u}\right)^{2}\right]$$

- Leading to incorrect Prandtl number (*Pr*) when the collision frequency is taken as constant: $\mu=\frac{p}{\nu}$ $\quad \kappa=\frac{5}{2}\frac{k}{m}\frac{p}{\nu}$ $\quad \mathrm{Pr}=1$

Boltzmann BGK Equation in 2 and 1 Dimensions

- Defining $f_2(\vec{x},\vec{c}_2,t)=\int_{-\infty}^{+\infty} f(\vec{x},\vec{c}_2,c_3,t)\,dc_3$ and assuming $\frac{\partial f_2}{\partial x_3}=0$
- Integrating the 3-D BGK Boltzmann equation,

$$\frac{\partial f_2}{\partial t}+\vec{c}_2\frac{\partial f_2}{\partial \vec{x}_2}+\vec{F}\frac{\partial f_2}{\partial \vec{c}_2}=\nu\left(f_2^{eq}-f_2\right) \quad ; \quad f_2^{eq}(\vec{c}_2)=\left(\frac{m}{2\pi kT}\right)\exp\left[-\frac{m}{2kT}\left(\vec{c}_2-\vec{u}_2\right)^2\right]$$

$$Q_i=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\psi_i f_2 d^2c \quad ; \quad Q_i=\left(\rho,\rho\vec{u}_2,e\right) \quad ; \quad \psi_i=\left[1,\vec{c}_2,\frac{1}{2}\left(\vec{c}_2-\vec{u}_2\right)^2\right]$$

- Similarly, for 1-D Boltzmann equation,

$$\frac{\partial f_1}{\partial t}+c\frac{\partial f_1}{\partial x}+F\frac{\partial f_1}{\partial c}=\nu\left(f_1^{eq}-f_1\right) \quad ; \quad f_1^{eq}(c)=\left(\frac{m}{2\pi kT}\right)^{\frac{1}{2}}\exp\left[-\frac{m}{2kT}\left(c-u\right)^2\right]$$

$$Q_i=\int_{-\infty}^{+\infty}\psi_i f_1 dc \quad ; \quad Q_i=\left(\rho,\rho u,e\right) \quad ; \quad \psi_i=\left[1,c,\frac{1}{2}\left(c-u\right)^2\right]$$

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Boltzmann-BGK Equation Solvers

- Gas-Kinetic Scheme (GKS)
- Discrete Ordinate Method (DOM)
- Lattice Boltzmann Method (LBM)

II. Discrete Ordinate Method (DOM)

Basic Features of DOM

- Finite discrete velocity points

$$f\left(c_x, c_y, c_z\right) \Rightarrow \left[f_{1,1,1}, \dots, f_{\alpha,\beta,\gamma}, \dots, f_{N_x,N_y,N_z}\right]$$

- Splitting method

$$\frac{\partial f_{\alpha,\beta,\gamma}}{\partial t} + \vec{c}_{\alpha,\beta,\gamma} \frac{\partial f_{\alpha,\beta,\gamma}}{\partial \vec{x}_{\alpha,\beta,\gamma}} = \Omega_{\alpha,\beta,\gamma} \rightarrow \frac{\partial f_{\alpha,\beta,\gamma}}{\partial t} + \vec{c}_{\alpha,\beta,\gamma} \frac{\partial f_{\alpha,\beta,\gamma}}{\partial \vec{x}_{\alpha,\beta,\gamma}} = 0 \text{ and } \frac{\partial f_{\alpha,\beta,\gamma}}{\partial t} = \Omega_{\alpha,\beta,\gamma}$$

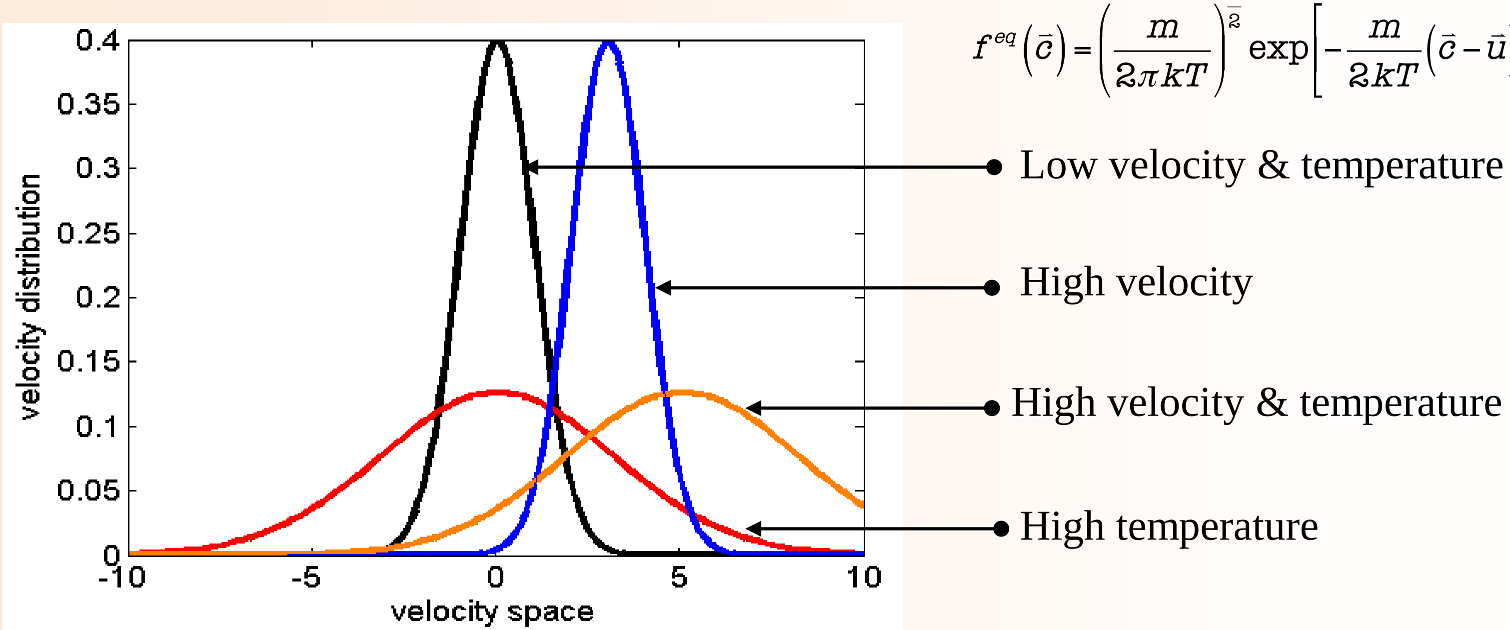
- Quadrature scheme

$$\iiint \psi_i f dc^3 \approx \sum_{\alpha=1}^{N_x} \sum_{\beta=1}^{N_y} \sum_{\gamma=1}^{N_z} W_{\alpha,\beta,\gamma} \left[\psi_i\left(c\right) f\left(c\right)\right]_{\alpha,\beta,\gamma}$$

Difficulties of Conventional DOM

- Difficulties encountered for high Mach number flows.
- High demand on computational resource due to large number of quadrature points needed to recover f^{eq}
- Sensitivity of the discrete point selection

$$f^{eq}(\vec{c}) = \left(\frac{m}{2\pi kT}\right)^{\frac{3}{2}} \exp\left[-\frac{m}{2kT}(\vec{c}-\vec{u})^2\right]$$



Objectives

- To develop a simple, robust, and efficient DOM solver
  - Reformulation: DOM with Dynamic Quadrature Scheme (DQS)
  - Code development
  - Code validation
  - Code speed enhancement
  - Code parallelization

III. Dynamic Discrete Ordinate Method

a. Dynamic Quadrature Scheme (DQS)

b. Transformed Moment Integral

c. Algorithm of Dynamic Discrete Ordinate Method (DDOM)

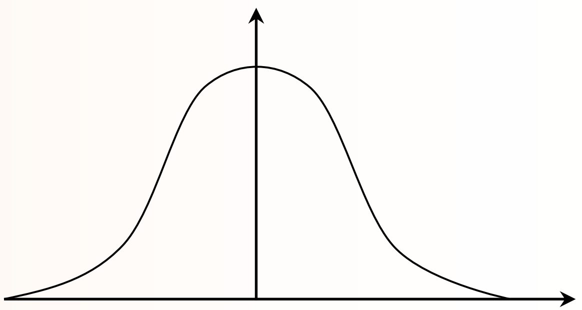
Dynamic Quadrature Scheme (DQS)

- Coordinate transformation of velocity space

$$\vec{\xi} = \frac{\vec{c} - \vec{u}}{\sqrt{2RT}}$$

- Gaussian profile

$$f \approx f^{eq} = G\left(\vec{\xi}\right) = \exp^{-\vec{\xi}^2}$$



- Gaussian-Hermite quadrature
  - Optimizing quadrature point distribution
  - Equivalency to the first-order Chapman-Enskog expansion
  - 3x3x3 quadrature points for recovery NSF equations for 3D case

$$Q = \iiint e^{-\vec{\xi}^2}\left(a_0 + \vec{a}_1\vec{\xi} + +\vec{a}_2\vec{\xi}^2 + \ldots\right)d\xi^3$$

$$Q = \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\sum_{k=1}^{N_z} W_{i,j,k} P^{\alpha}_{i,j,k}\left(\vec{\xi}\right) + O\left(Kn^{\alpha}\right)$$

- Dynamic nature
  - $\vec{c}$ changing with time and space for fixed $\vec{\xi}$

Transformed Moment Integral

- Transformed moment integral

$$Q_i = \iiint \psi_i\left(\vec{\xi}\right) f\left(\vec{\xi}\right) \left\| \vec{\vec{J}} \right\| d\xi^3$$

- Gaussian-like integrand

$$Q = \iiint e^{-\xi^2} \left(b_0 + \vec{b}_1 \vec{\xi} + + \vec{b}_2 \vec{\xi}^2 + \ldots\right) d\xi^3$$

Algorithm of Dynamic Discrete Ordinate Method

- Similar to the algorithm of DOM
  - Solving BGK Boltzmann equation in physical velocity space
    - Collision
    - Advection
  - Calculating hydrodynamic moments in transformed velocity space
    - DQS
    - Error of $O(Kn^2)$ with 3 x 3 x 3 Gaussian-Hermite quadrature
  - Collision modeling
    - Viscous flow (Chapman-Enskog expansion)

$$f = f^{eq}\left(1+\phi\right) \quad \text{where} \quad \phi = -\frac{1}{\nu f^{eq}}\left(\frac{\partial f^{eq}}{\partial t} + \vec{c}\cdot\frac{\partial f^{eq}}{\partial \vec{x}}\right)$$

    - Inviscid flow

$$n \to \infty \setminus \qquad f = f^{eq}$$

IV. Numerical Results and Validations

a. Hardware/software configuration

b. One-dimensional Riemann problems

c. Two-dimensional problems

- Riemann problems
- Backward-step problem
- Shock-reflection problem
- Cavity flow problems

d. Performance comparison with Roe solver

Hardware/software configuration

- Dell T7400 with dual CPU of Intel Xeon 5482 @ 3.20GHz
- No any parallelization and optimization flag
- Fortran PGI 10.0

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

One-Dimensional Riemann Problems

- Shock tube problems

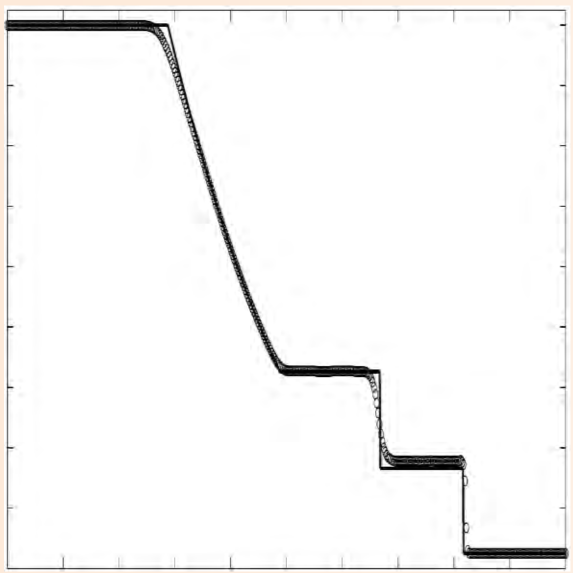
Initial conditions	p_l, ρ_l, u_l	p_r, ρ_r, u_r	t_{final}
Sod (subsonic)	1, 1, 0	0.1, 0.125, 0	0.18
Sjogreen (subsonic)	1.8, 1, -1	1.8, 1, 1	0.14
Sjogreen (supersonic)	0.4, 1, -2	0.4, 1, 2	0.14

Numerical setting	grid nos.	quad. pts.	$\Delta t/\Delta x$
	400	3	0.125

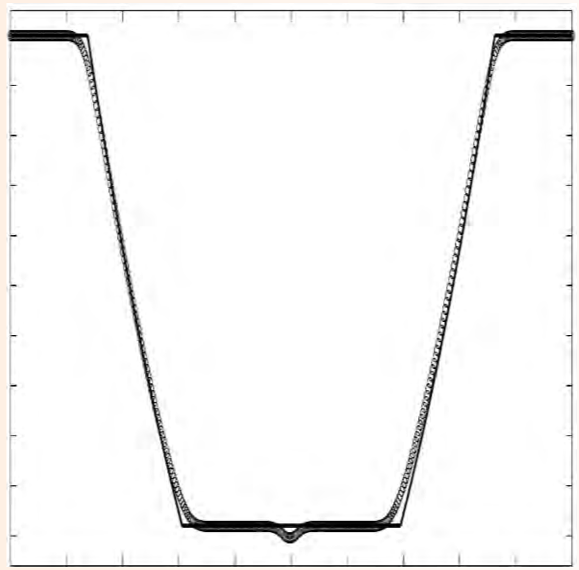
香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

One-Dimensional Riemann Problems

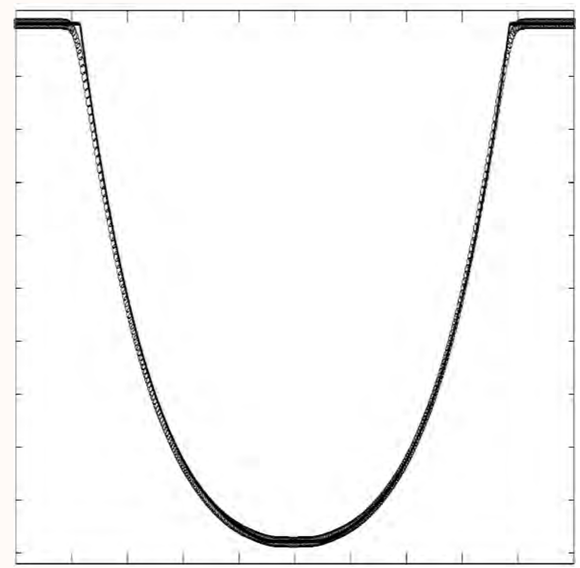
- Numerical result of density profile (reference to solid-line of exact solution)



Sod (subsonic)



Sjogreen (subsonic)



Sjogreen (supersonic)

One-Dimensional Riemann Problems

- 1-D speedup tests (with incomplete transformations)

DOM	DDOM-S	DDOM-P	DDOM
$\vec{\xi} = \vec{c}$	$\vec{\xi} = \vec{c} - \vec{u}$	$\vec{\xi} = \frac{\vec{c}}{\sqrt{2RT}}$	$\vec{\xi} = \frac{\vec{c} - \vec{u}}{\sqrt{2RT}}$

Sod (subsonic)	DOM	DDOM-S	DDOM-P	DDOM
Quad. Pts.	18	14	8	3
Speedup	1.00	0.88	1.45	4.27
Quad. Eff.	1.00	1.46	1.55	1.46

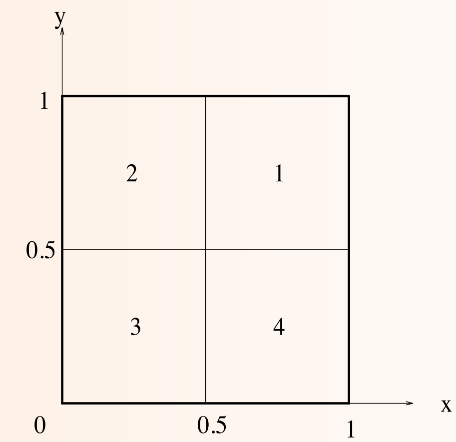
Sjogreen (supersonic)	DOM	DDOM-S	DDOM-P	DDOM
Quad. Pts.	20	24	14	3
Speedup	1.00	0.57	0.99	4.82
Quad. Eff.	1.00	1.46	1.42	1.38

Two-Dimensional Problems

- 2-D Riemann problem
- Backward-step problem
- Shock-reflection problem
- Cavity flow problem
- 2-D speedup tests

Two-Dimensional Problems

- 2-D Riemann problems

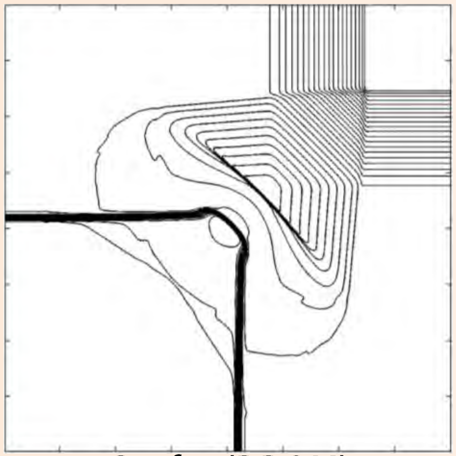


Initial conditions		left	right	t_{final}
Conf. 7 (subsonic) $[p, \rho, u, v]$	upper	0.4, 0.5197, -0.6259, 0.1	1, 1, 0.1, 0.1	0.25
	lower	0.4, 0.8, 0.1, 0.1	0.4, 0.5197, 0.1, -0.6259	
Conf. 15 (supersonic) $[p, \rho, u, v]$	upper	0.4, 0.5197, -0.6259, -0.3	1, 1, 0.1, -0.3	0.20
	lower	0.4, 0.8, 0.1, -0.3	0.4, 0.5313, 0.1, 0.4276	
Conf. 5 (supersonic) $[p, \rho, u, v]$	upper	1, 2, -0.75, 0.5	1, 1, -0.75, -0.5	0.23
	lower	1, 1, 0.75, 0.5	1, 3, 0.75, -0.5	

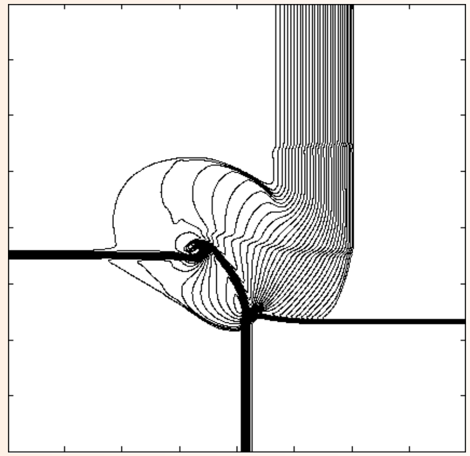
Numerical setting	grid nos.	quad. pts.	$\Delta t / \Delta x$
	400 x 400	2 x 2	0.125

Two-Dimensional Problems

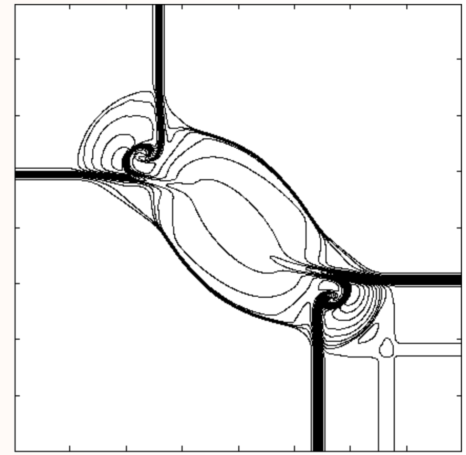
- Numerical result of density contours (reference to Riemann solver)



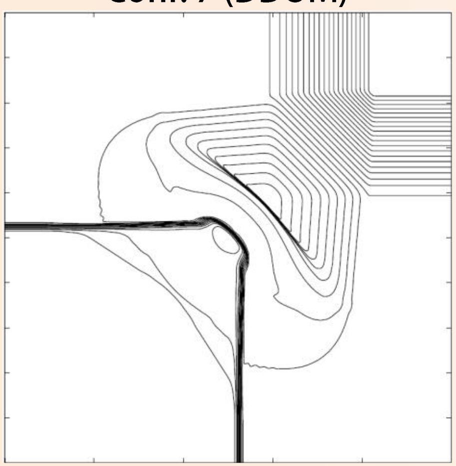
Conf. 7 (DDOM)



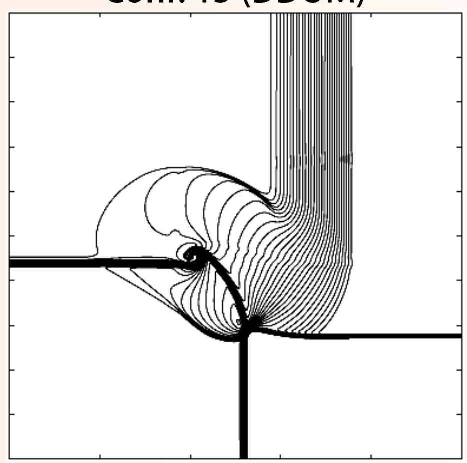
Conf. 15 (DDOM)



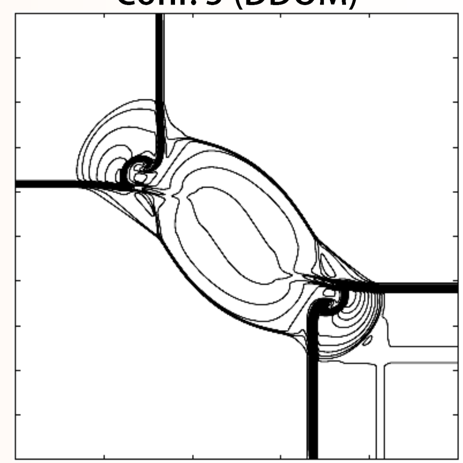
Conf. 5 (DDOM)



Conf. 7 [Lax and Liu (1998)]



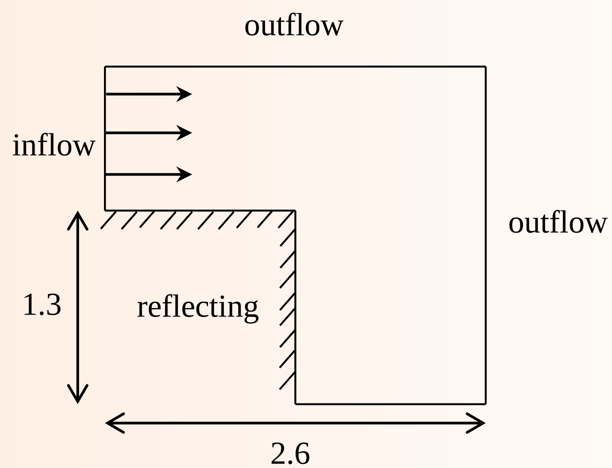
Conf. 15 [Lax and Liu (1998)]



Conf. 5 [Lax and Liu (1998)]

Two-Dimensional Problems

- Backward-step problem

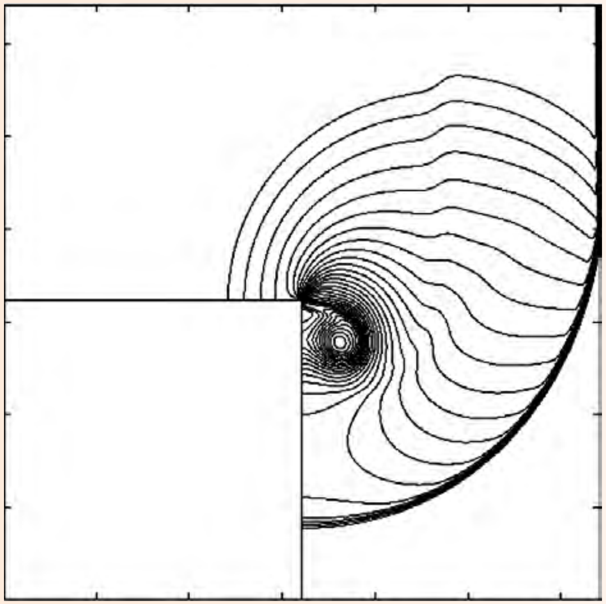


Initial conditions	p	ρ	u	v
inflow	2.4583	1.8620	0.8216	0
outflow	1	1	0	0

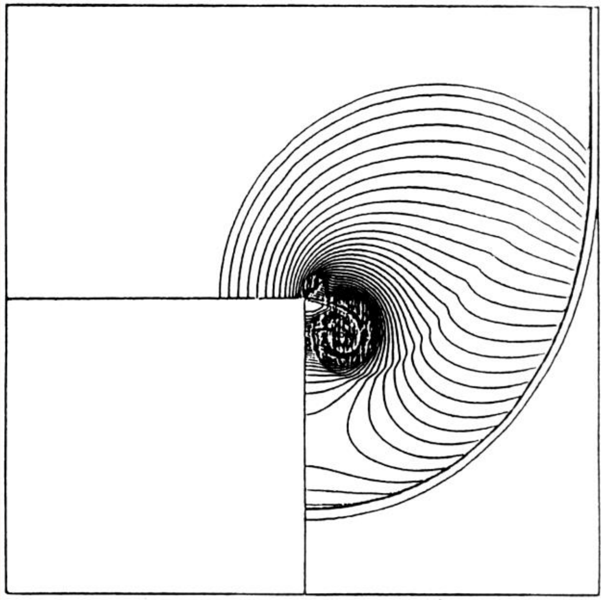
Numerical setting	grid. nos.	quad. pts.	$\Delta t/\Delta x$	t_{final}
	320 x 320	3 x 3	0.125	0.75

Two-Dimensional Problems

- Numerical result of density contours (reference to Riemann solver)



Backward-step (DDOM)

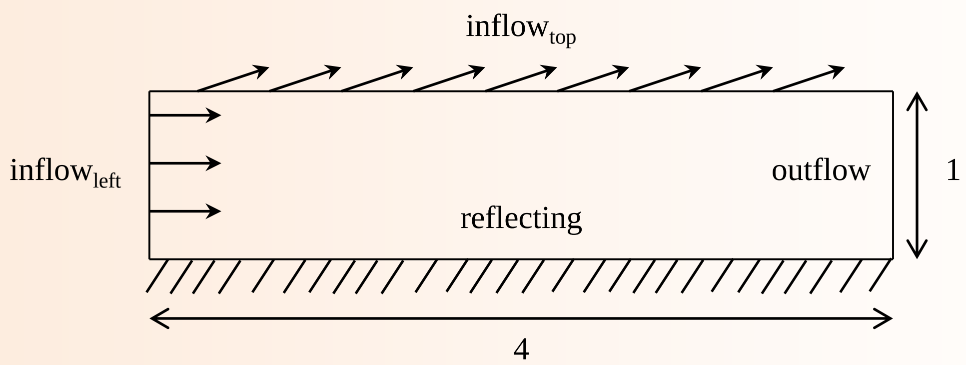


Backward-step [Takayama and Inoue (1991)]

香港科技大學 THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Two-Dimensional Problems

- Shock-reflection problem

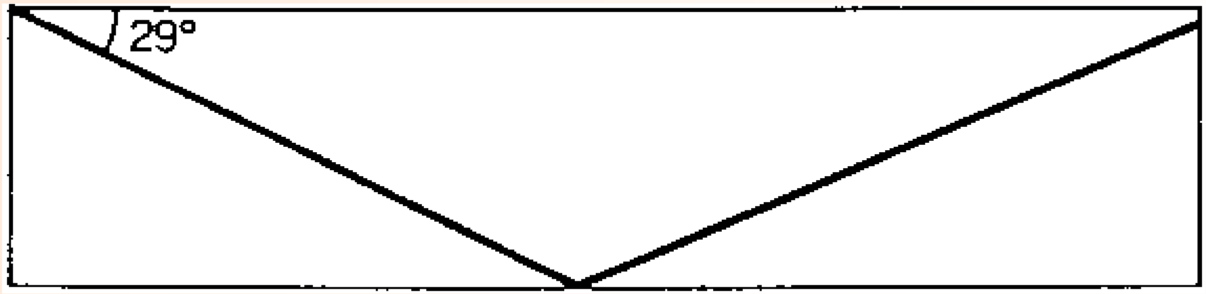


Boundary conditions	p	ρ	u	v
top	1.5282	1.69997	2.61934	0.50632
left	0.7143	1	2.9	0

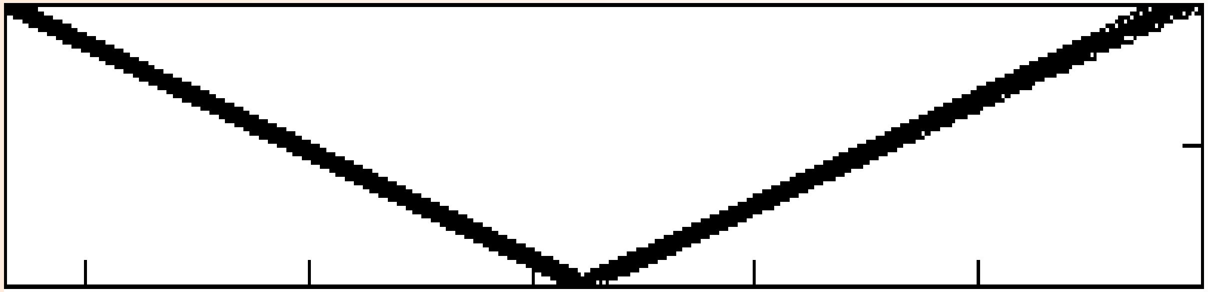
Numerical setting	grid. nos.	quad. pts.	$\Delta t/\Delta x$	t_{final}
	800 x 200	3 x 3	0.125	2.0

Two-Dimensional Problems

- Numerical result of pressure contours (reference to theoretical prediction)



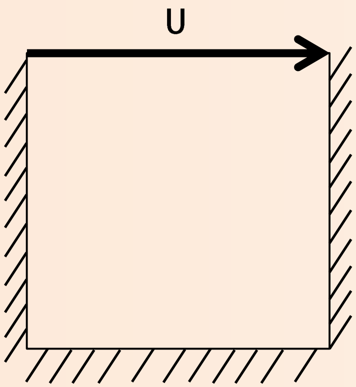
Shock-reflection [Colella (1987)]



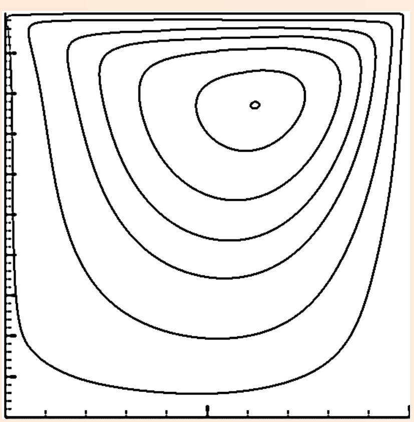
Shock-reflection (DDOM)

Two-Dimensional Problems

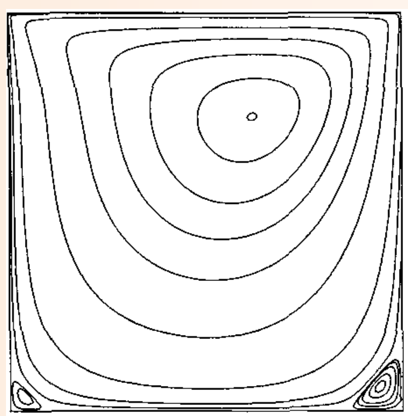
- Steady state solution of cavity flow (reference to LBM solver)



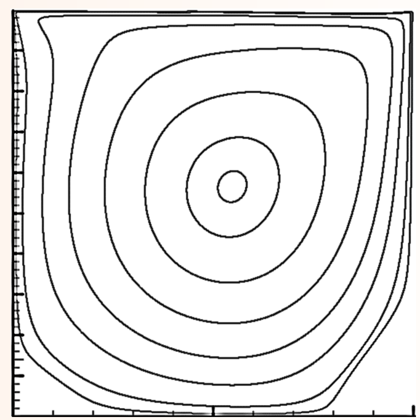
Re	Mach nos.	U_{top}	U_{side}	grid nos.	quad. pts.	$\Delta t/\Delta x$	t_{final}
100	0.15	0.1	0	502 x 502	3x3	0.3	¥
1,000	0.15	1	0	502 x 502	3x3	0.3	¥



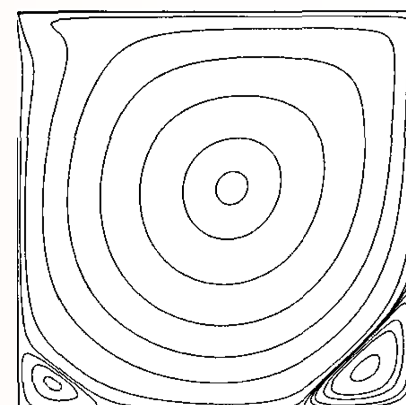
Re = 100 (DDOM)



Re = 100 [Hou et al.(1995)]



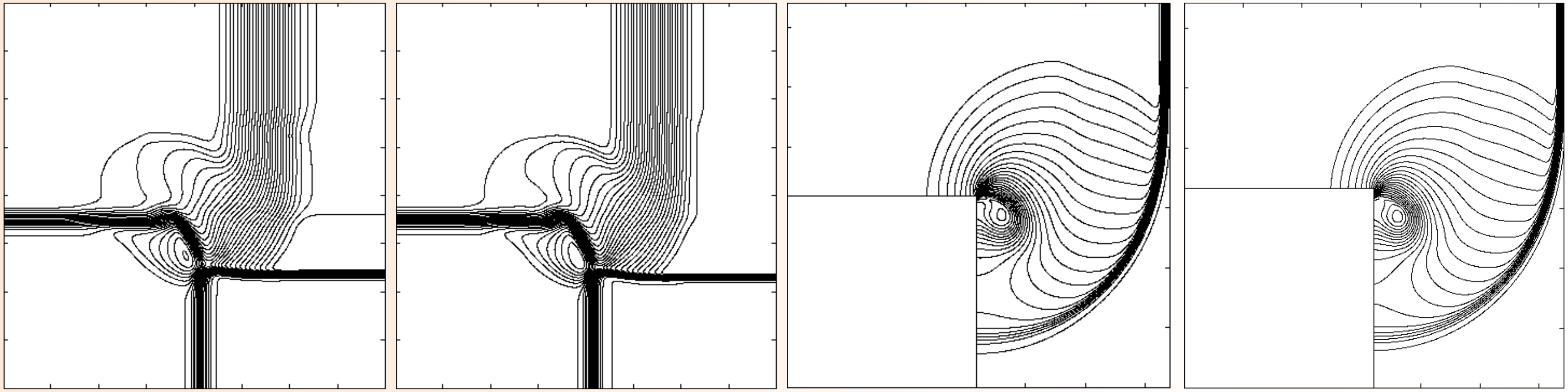
Re = 1,000 (DDOM)



Re = 1,000 [Hou et al.(1995)]

Two-Dimensional Problems

- 2-D speedup tests (Riemann and Backward-step problems)



Conf. 15 (DOM) | Conf. 15 (DDOM) | Backward-step (DOM) | Backward-step (DDOM)

Convergence rate	Quad. Pts.		Speedup	
	DOM	DDOM	DOM	DDOM
Riemann	256	4	1.00	20.09
Backward-Step	324	9	1.00	14.98

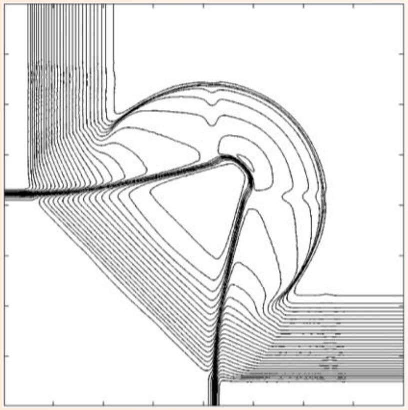
Performance Comparison with Roe Solver

- Roe solver by *CLAWPACK* (approximated Riemann solver)
  - Stable
  - Efficient
  - Low-numerical viscosity
- Enhanced DDOM solver
  - Inviscid flow model
  - Fast-scheme + Adaptive quadrature method

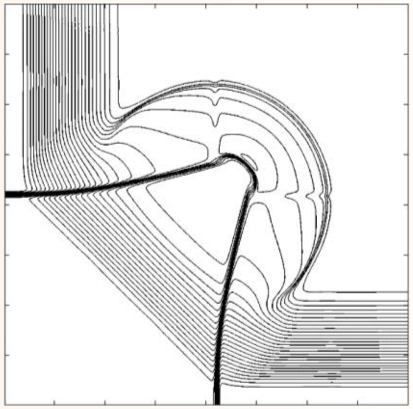
香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Performance Comparison with Roe Solver

- **Speedup tests** (Conf. 8 of 2D Riemann problem with 400 x 400 grids)



Enhanced DDOM

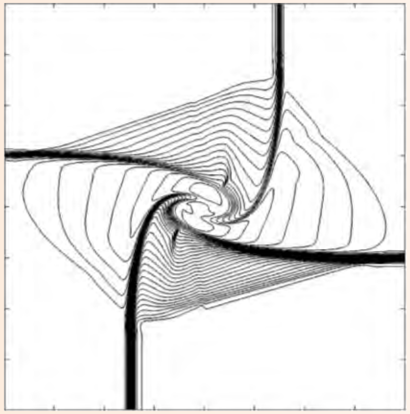


Splitting (Roe)

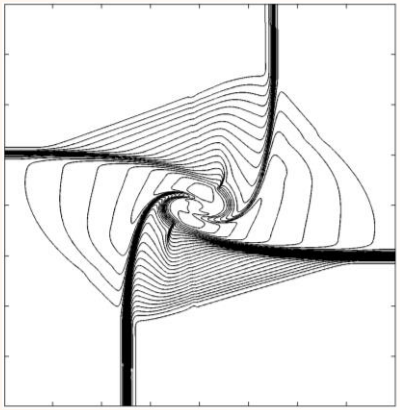
- DDOM is 14% faster than Roe Solver

Performance Comparison with Roe Solver

- **Speedup tests** (Conf. 6 of 2D Riemann problem with 400 x 400 grids)



Enhanced DDOM



Splitting (Roe)

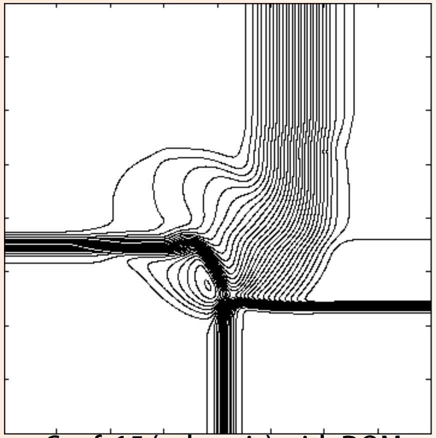
- DDOM is 14% faster than Roe Solver

Concluding Remarks on 2-D DDOM

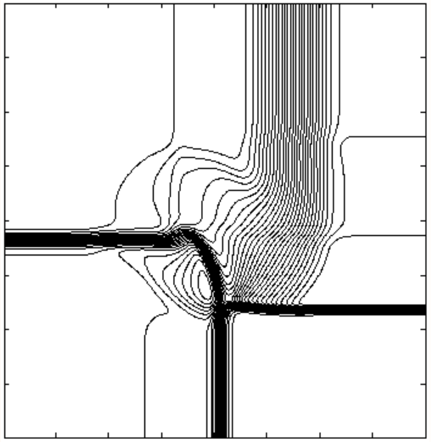
1. 2D-DDOM code is 20 times faster than conventional DOM.
2. Enhanced 2D-DDOM code is 14% faster than Roe solver.
3. With single CPU, real running time about 100 seconds for the 2-D Riemann problems of very short time (0.3 second).
4. Estimated time, for 3-D Riemann problem with grid 400 x 400 x 400 and quadrature 3 x 3 x 3, will be about 33 hours.
5. Parallel computing is inevitable for practical 3-D DDOM simulations.

Preliminary 3-D Test Results

- Riemann problem with 400 x 400 x 10 grids



Conf. 15 (subsonic) with DOM



Conf. 15 (subsonic) with DDOM

- 200 times speedup

Conf. 15 (subsonic)	Quad. pts.	Speedup	Cost/step/grid(ms)
DDOM	2 x 2 x 2	232X	1.31
DOM	16 x 16 x 16	1X	304.

V. Massively Parallel Computation

a. Features of DDOM for parallelization

b. Methods of parallel computing

c. Hardware configuration

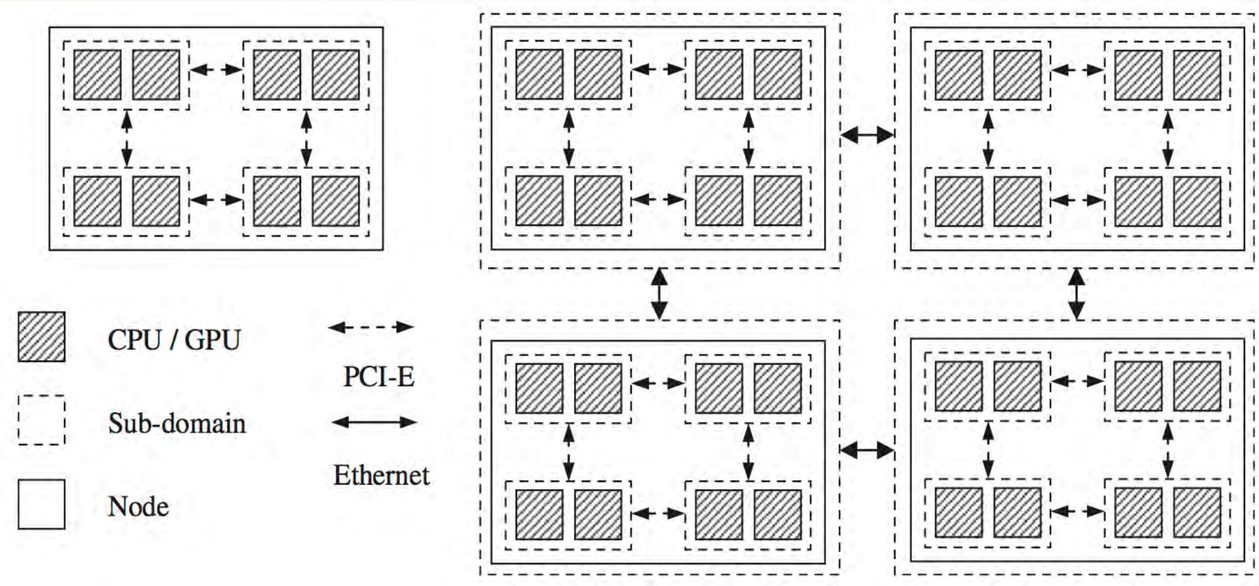
d. Performance tests

Features of DDOM for Parallelization

- Uncoupled features
  - Spatial space (*x*, *y*, and *z* grid system)
  - Velocity space (c_α, c_β and c_γ quadrature points)
- Data-localization
  - Nearby spatial and velocity space data
  - Low data accessing time
- Simple algebraic algorithm
  - No iteration
  - Only algebraic operation

Methods of Parallel Computing

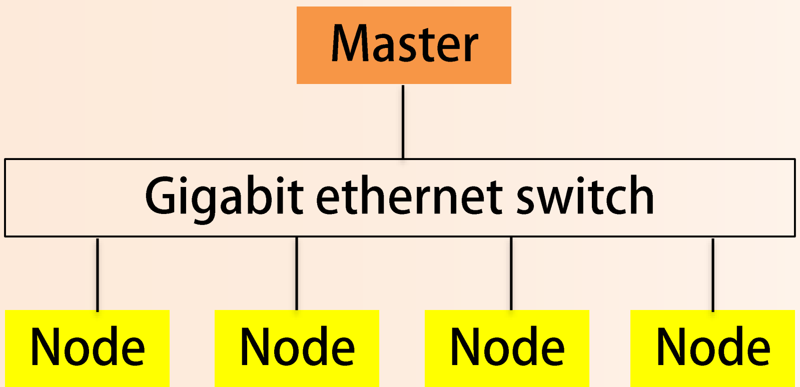
- Parallel programming methods
  - Domain decomposition
  - Parallel programs (OpenMP, MPI, CUDA)



Hybrid parallelism model

Hardware Configurations

- Beowulf cluster



Schematic of Beowulf cluster

Setup of Beowulf cluster

Platform	Device	Node	Core/Node	Processor/Core	Processor
Cluster	Intel W5590	8	2	4	64
Cluster	Nvidia GTX 460	2	2	336	1,344
	Intel W5590				

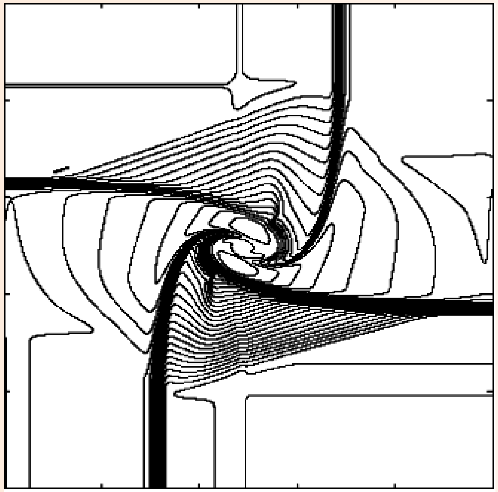
Test Problems and Settings

Test problems		Grid nos.	Quad. pts.	Scheme	Parallel-level
A	2-D Riemann	2,500 x 2,500	3 x 3	Standard	Thread
B	2-D Riemann*	640 x 640 x 44	3 x 3 x 3	Fast	Data
C	2-D Cavity flow	1,024 x 1,024	3 x 3	Standard	Data

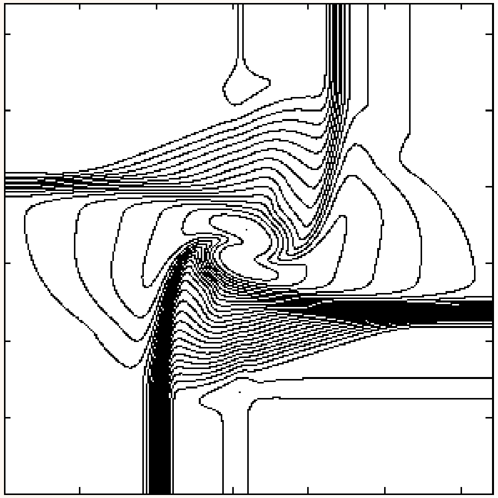
Test nos.	Program	Memory	Node	Core / GPU
A	OpenMP/MPI	Shared/Distributed	8	16 / 32 / 64
B	OpenMP/MPI/CUDA	Shared/Distributed	2	1 / 2 / 4
C	OpenMP/MPI/CUDA	Shared/Distributed	2	1 / 2 / 4

* 2-D Riemann problem calculated with 3D-DDOM code, where the gradient of the macroscopic variables in z-direction are given as zero.

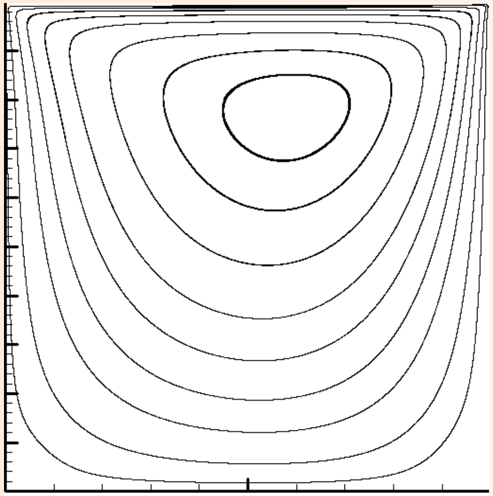
Preliminary Results of Parallel-DDOM



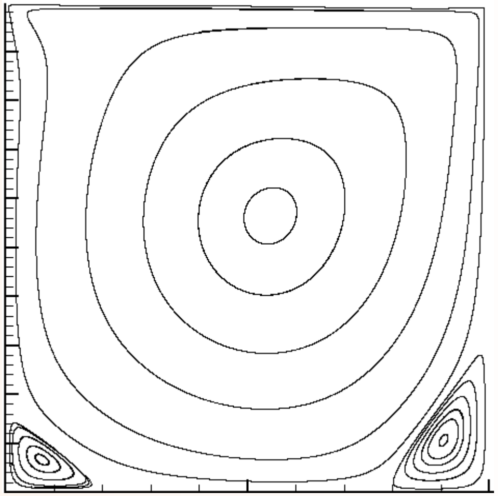
Test A, 2D DDOM (CPU)



Test B, 3D DDOM (GPU)



Test C, Cavity Flow at Re=100 by (GPU)

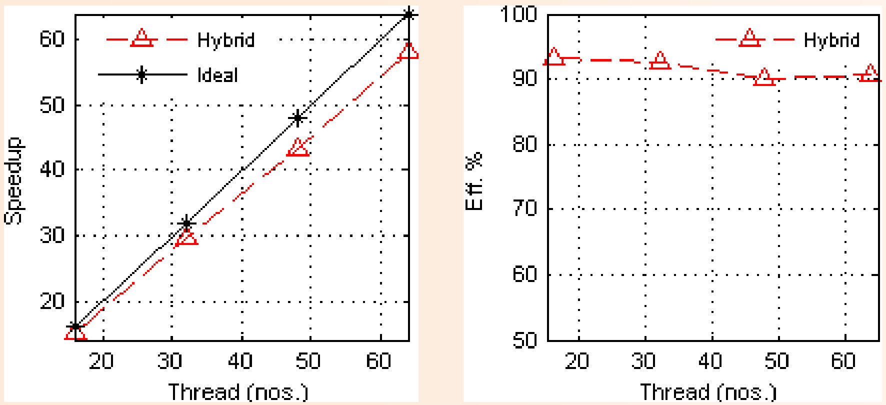


Test C, Cavity Flow at Re=1,000 by (GPU)

Parallel-DDOM Performance

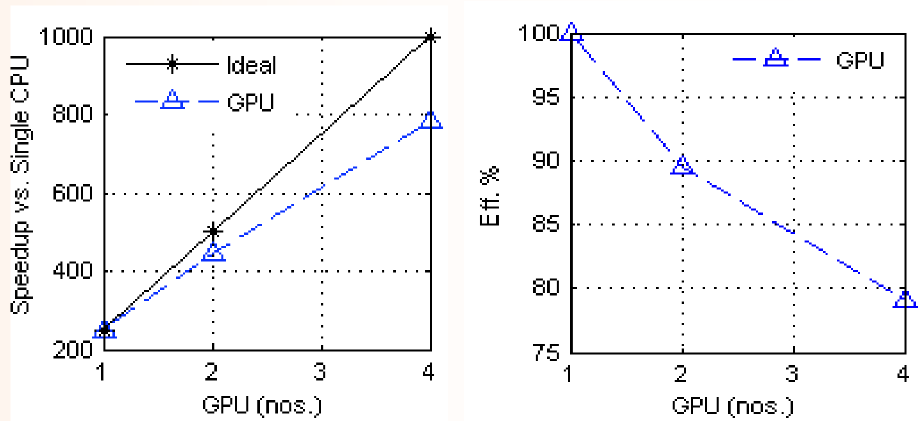
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY
香港科技大學

Test A with CPU-Cluster



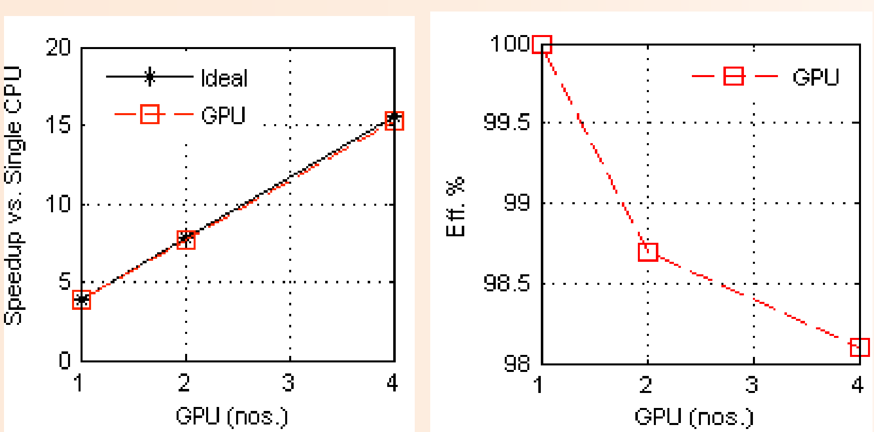
(2D Riemann | Δ OpenMP/MPI)

Test B with GPU-Cluster



(3D Riemann | Δ OpenMP/MPI/CUDA)

Test C with GPU-Cluster



(2D Cavity Flow | □ OpenMP/MPI/CUDA)

- Over 90% parallel efficiency of parallel-DDOM solver is achieved for CPU-cluster.
- Around 80 to 90 % parallel efficiency of parallel-DDOM solver is achieved for GPU-cluster.
- Parallel DDOM is a good candidate for parallel computing.

- **DDOM Code Summary**

Robust	Inviscid flow	✔
	Viscous flow	✔
	Incompressible flow	✔
	Compressible flow	✔

Efficient	Quadrature points	2x2x2 — 3x3x3 for 3D case
	Speedup over DOM	Around 20X for 2D case
	Performance	Efficiency as Roe solver

Scalable	Speedup by CPU cluster	Over 57X @ 90% parallel efficiency
	Speedup by GPU cluster	Over 780X @ 80% parallel efficiency

VI. Conclusions

1. DOM was reviewed to reveal its low-efficiency deficiency at high Mach number.
2. DQS was proposed and implemented to DOM, termed as DDOM, to resolve the deficiency of DOM.
3. One and two-dimensional DDOM codes were developed and tested to show the codes are robust applicable to different type of flows.

VI. Conclusions

4. The 1-D DDOM code is 5-times faster than conventional 1-D DOM code, and 20-times faster for 2-D DDOM code.
5. Further enhancement of the 2-D DDOM code achieved 14% faster than Roe solver.
6. Parallel 2-D and 3-D DDOM codes have been developed and preliminarily tested.
7. Over 90% parallel efficiency has been achieved with 64-CPUs cluster, and about 80 to 90% parallel efficiency with 4-GPUs cluster (780 times speedup).

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

THANK YOU