# Some Applications for Extreme Computing

**Jin XU**

**5/20/2012**

**Institute of Computing**
**Chinese Academy of Science**
**Beijing, China**

**First Workshop of the MCS-ISCAS Joint Laboratory for Parallel Processing and Computing Technologies (PPCT)**

# Few points before the talk



- As you already know that I have worked in ANL before, and now I am still collaborating with researchers in ANL/MCS. So I am qualified in this joint laboratory;
- This talk is about scientific computing and numerical methods which is a little bit different from the research interests of most participants in this workshop, the simulations presented in this talk needs lots of optimizations in computer science and hope to collaborate with experts attending this workshop;
- The talk will include some introduction on the physics of the simulations, and I hope they are interesting to you.

# Overview



- Introduce some large scale simulations that have been done and numerical methods for them, due to the time limit this talk only focus in fluid simulations.
- Extreme computing provides more capability for scientific computing than before. Now it is meaningful to revise and improve the numerical methods and parallel models used before so that extreme computing can be efficiently used for some typical computing challenges such as turbulence, plasma, and combustion, etc.
- High-order numerical methods have been used in these simulations.
- This talk will discuss following questions:
  - Why these simulations need extreme computing?
  - What are the benefits of extreme computing for these simulations?
  - Which numerical methods are scalable for extreme computing and how to improve them?
  - What are the benefits and difficulties of using high-order numerical methods in extreme computing?
- These simulations need more improvements on the side of computer science such as parallel models, parallel I/O, etc.

# Content



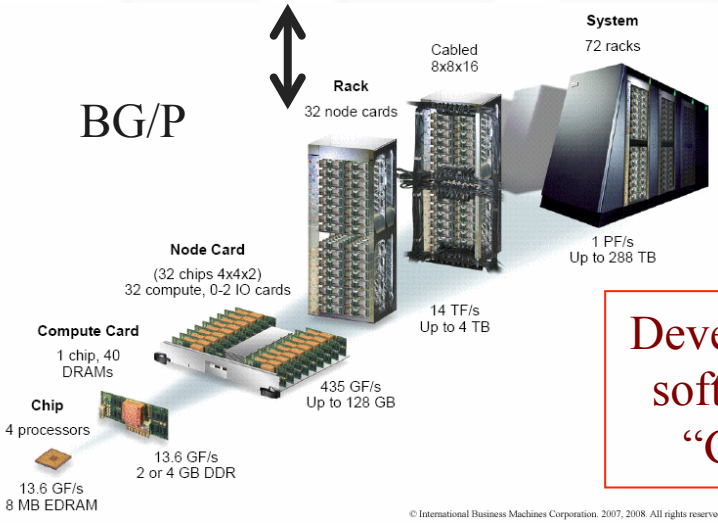
- Fluid Mechanics
  - Incompressible flow, compressible flow, inviscid flow, viscous flow, multiphase flow, laminar flow, turbulent flow, ……
  - Finite volume, Finite Difference, Finite Element, Spectral Fourier, Spectral Element, Lattice, Mesh free, *hp*-Finite Element, ……
  - Direct Numerical Simulation of turbulent flow
  - Channel turbulence DNS
  - Pipe turbulence DNS
  - DNS of flow pass cylinder with one homogeneous direction
  - DNS of flow in arbitrary 3D domain
- Parallel Poisson Solver
  - Structured and unstructured grids
  - Cartesian and cylindrical coordinate systems
  - Continuous and Discontinuous Galerkin methods

# High-order numerical methods

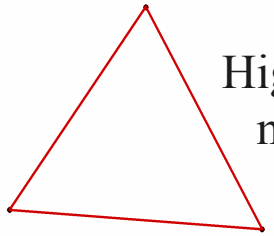


Software package build with low-order methods for extreme computing

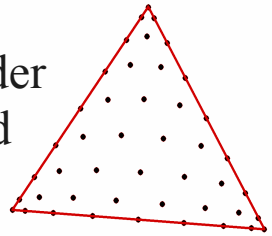
BG/P




Low-order method

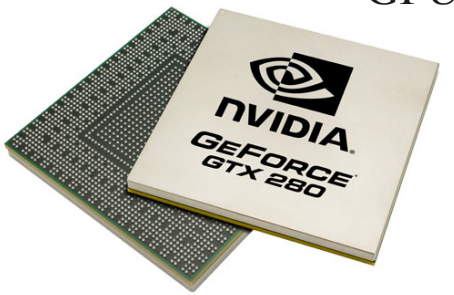


High-order method



CPU

GPU




Developing extreme software with soft "GPU"-HOMs

- Extreme computing software usually use low-order methods
- HOMs can achieve high accuracy than LOMs in some situations
- HOMs can not replace LOMs completely
- It needs more efforts to use HOMs

- Supercomputers usually are built with CPUs
- GPU can achieve faster speed than CPU in some situations
- GPU can not replace CPU completely
- It needs more efforts to use GPU

# Why Turbulence DNS needs extreme computing?



- Most materials in nature are in the state of fluids
- Most fluids are in turbulent state
- Three ways to investigate turbulence: (Experiment, theory and simulation)

$$\frac{\partial \vec{V}}{\partial t} + \vec{V}\cdot\nabla\vec{V} = -\nabla p + \frac{1}{\mathrm{Re}_m}\nabla^2\vec{V}, \quad \nabla\cdot\vec{V} = 0$$

| |
|---|
| Direct Numerical Simulation (DNS) |
| Large Eddy Simulation (LES) |
| Reynolds Average Navier-Stokes (RANS) |

Kolmogorov's Theory of 1941

- Turbulent flow is composed by "eddies" of different sizes.
- The large eddies are unstable and eventually break up into smaller eddies, and the kinetic energy of the initial large eddy is divided into the smaller eddies that generated from it.

Kolmogorov's length, velocity and time scales

$$\eta = \left(\frac{\nu^3}{\varepsilon}\right)^{1/4} \quad u = \left(\frac{\nu}{\varepsilon}\right)^{1/4} \quad \tau = \left(\frac{\nu}{\varepsilon}\right)^{1/2}$$

Energy spectrum function

$$E(k) = C\varepsilon^{2/3}k^{-5/3}$$

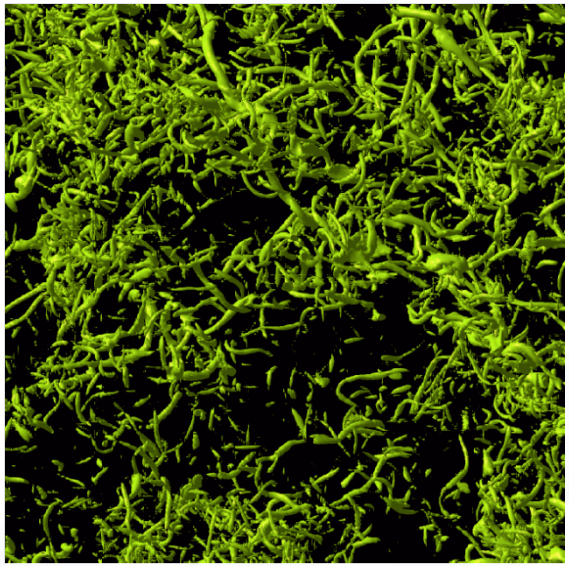
Total cost of DNS $\propto \mathrm{Re}_m^3$

$$\mathrm{Re}_m = \frac{U_m H}{\nu} \qquad \mathrm{Re}_\tau = \frac{U_\tau H}{\nu} \qquad \rho U_\tau^2 = \mu \left.\frac{\partial \bar{u}}{\partial y}\right|_{y=0} \qquad \mathrm{Re}_m \approx 2000$$

# Homogeneous isotropic turbulence



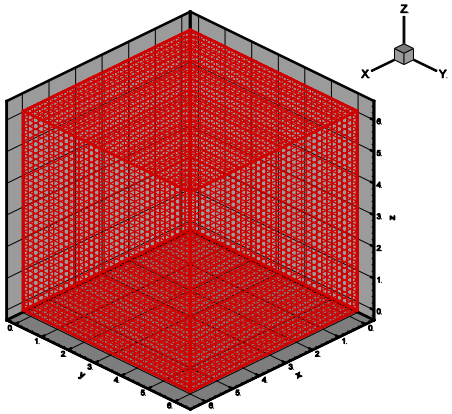
Borrowed $4096^3$



$$\mathrm{Re}_\lambda \propto \mathrm{Re}^{1/2}, \mathrm{Re} = \frac{UL}{\nu}$$

16.4-Tflops Direct Numerical Simulation of turbulence by a Fourier Spectral Method on the Earth Simulator

Mitsuo Yokokawa, etc.



| $N$ | $R_\lambda$ | $\nu(\times 10^{-4})$ |
|---|---|---|
| 512 | 257 | 2.8 |
| 1024 | 471 | 1.1 |
| 2048 | 732 | 0.44 |
| 4096 | 1217 | 0.173 |

$$\varphi(x,y,z,t) = \sum_{m=-M/2}^{M/2-1} \sum_{n=-N/2}^{N/2-1} \sum_{p=-P/2}^{P/2-1} \varphi(m,p,n,t) e^{-i\alpha m x} e^{-i\beta n y} e^{-i\gamma p z}$$

$$\varphi(0,y,z,t) = \varphi(2\pi,y,z,t)$$

$$\varphi(x,0,z,t) = \varphi(x,2\pi,z,t)$$

$$\varphi(x,y,0,t) = \varphi(x,y,2\pi,t)$$

For channel flow:

Laminar flow: $\mathrm{Re}_D < 2300$

Transition flow: $2300 < \mathrm{Re}_D < 4000$

Turbulent flow: $\mathrm{Re}_D > 4000$

$$\mathrm{Re}_\tau = 180 \Leftrightarrow \mathrm{Re}_D = 4300$$

In 2005:

$$\mathrm{Re}_\tau = 2320 \Leftrightarrow \mathrm{Re}_D = 51500$$

Supercomputer now in 2012 can simulate

$$\mathrm{Re}_\tau = 10^4 \Leftrightarrow \mathrm{Re}_D = 5\times 10^5$$

Typical Reynolds number $\sim 10^7$

# Turbulent Channel Flow

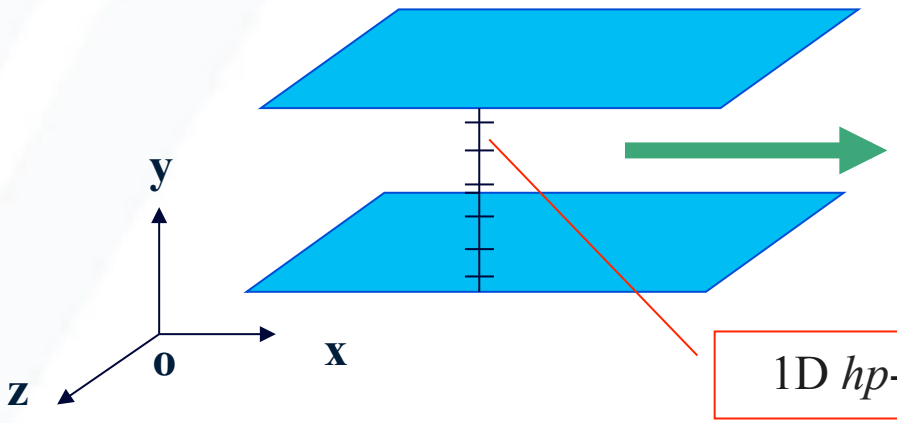


*Navier-Stokes Equation*

$$\frac{\partial \vec{V}}{\partial t} + \vec{V}\cdot\nabla\vec{V} = -\nabla p + \frac{1}{\mathrm{Re}_m}\nabla^2\vec{V}, \quad \nabla\cdot\vec{V} = 0$$

*hp*-Finite Element Method in 1D

- Kim, Moin, Moser 1987, JFM




$$\mathrm{Re}_m = \frac{U_m H}{\nu}$$

Flow

1D *hp*-Finite Elements

$$u(x,y,z,t) = \sum_{m=-M/2}^{M/2-1}\sum_{n=-N/2}^{N/2-1}\sum_{0}^{P} u(m,p,n,t)e^{-i\alpha m x}e^{-i\beta n z}P_p(y)$$

$$P_p(y) = \begin{cases} (\frac{1-y}{2}) & p=0 \\ (\frac{1-y}{2})(\frac{1+y}{2})P_{p-1}^{1,1}(y) & 0<p<P \\ (\frac{1+y}{2}) & p=P \end{cases}$$

Multi Elements in y
Faster Speed
Stability Increased

# Numerical Method



- The first DNS of homogeneous turbulence by Steven Orszag (1973)
- The first channel turbulence DNS was done by Kim, Moin and Moser in 1987.
- 3rd Order Splitting Scheme
- De-aliasing in stream-wise and span-wise directions
- Static Condensation Technique
- FFTW, BLAS, LAPACK Libraries
- MPI parallelization
- Iterative and Direct Solver (1D)

***Time Splitting Scheme***

1. Nonlinear step:

$$\frac{\vec{V}^{s+1/3} - \sum_{q=0}^{J_i-1} \alpha_q \vec{V}^{s-q}}{\Delta t} = \sum_{q=0}^{J_e-1} \beta_q N(\vec{V}^{s-q})$$

2. Pressure step:

$$\frac{\vec{V}^{s+2/3} - \vec{V}^{s+1/3}}{\Delta t} = -\nabla \Pi^{s+1}$$

$$\nabla \cdot \vec{V}^{s+2/3} = 0$$

$$\frac{\partial \Pi_{s+1}}{\partial n} = \vec{n} \cdot [\sum_{q=0}^{J_e-1} \beta_q N(\vec{V}^{s-q}) + \upsilon \sum_{q=0}^{J_e-1} \beta_q (-\nabla \times (\nabla \times \vec{V}^{s-q}))]$$

3. Viscous step:

$$\frac{\vec{V}^{s+1} - \vec{V}^{s+1/3}}{\Delta t} = \upsilon \nabla^2 \vec{V}^{s+1}$$

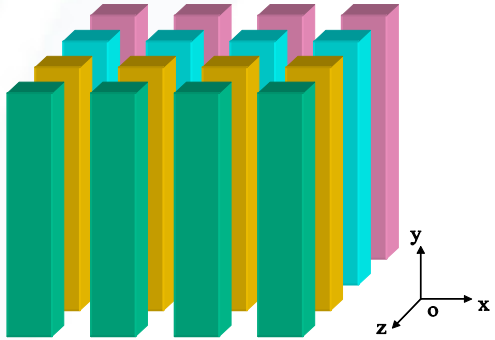
# Parallel Models



A

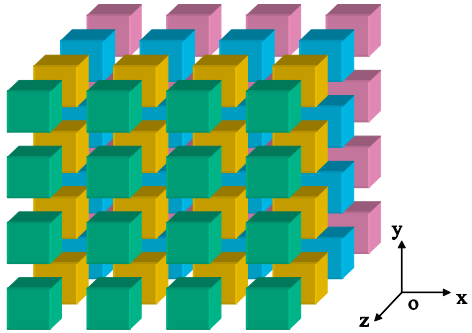
Grid = $2048^3$

B




$1000 \times 1000 = 10^6$

C




$100 \times 100 \times 100 = 10^6$

- Model A is hard to use one million cores, Model B can use but not easily, only model C can easily use.

- Fourier expansion and Chebyshev polynomial are global functions, which are not suitable for using large number of processors.

- We changed the method to Fourier-Fourier-1D *hp* Finite Element Method, which can easily fits elements to processors in y direction. This means that we have used method with local expansion to substitute the method with global expansion method.

# De-aliasing in x and z directions



$$\frac{\vec{V}^{n+1/3} - \sum_{q=0}^{J_e-1} \alpha_q \vec{V}^{n-q}}{\Delta t} = \sum_{q=0}^{J_e-1} \beta_q N(\vec{V}^{n-q})$$

$$N(\vec{V}) = u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}$$

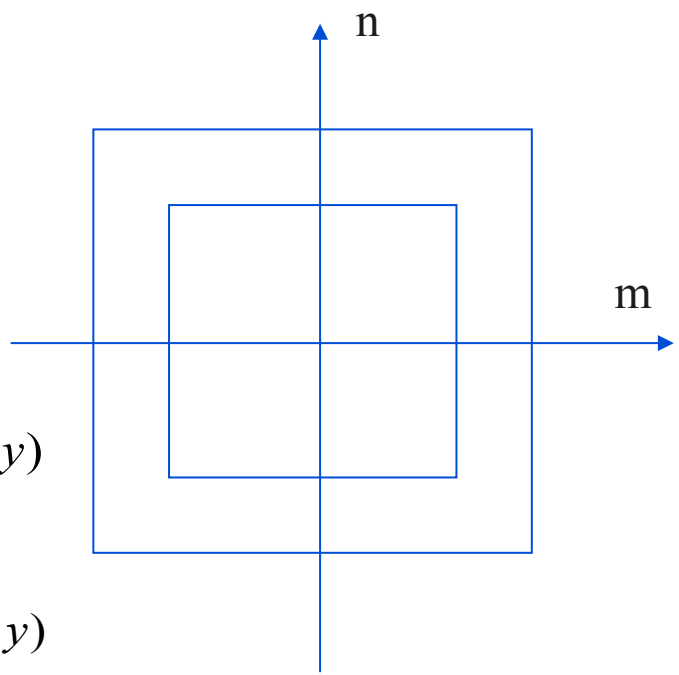
$$u_0(x,y,z,t) = \sum_{m=-3M/2}^{3M/2-1} \sum_{n=-3N/2}^{3N/2-1} \sum_{0}^{P} \hat{u}_0(m,p,n,t) e^{-i\alpha m x} e^{-i\beta n z} P_p(y)$$

$$\frac{\partial u_0}{\partial x}(x,y,z,t) = \sum_{m=-M/2}^{M/2-1} \sum_{n=-N/2}^{N/2-1} \sum_{0}^{P} \frac{\partial \hat{u}_0}{\partial x}(m,p,n,t) e^{-i\alpha m x} e^{-i\beta n z} P_p(y)$$

De-aliasing

$$F(x,y,z) = u_0(x,y,z)\frac{\partial u_0}{\partial x}(x,y,z)$$

$$\hat{F}(m,y,n,t) = \sum_{m=-M/2}^{M/2-1} \sum_{n=-N/2}^{N/2-1} F(x,y,z,t) e^{i\alpha m x} e^{i\beta n z}$$




Fourier space

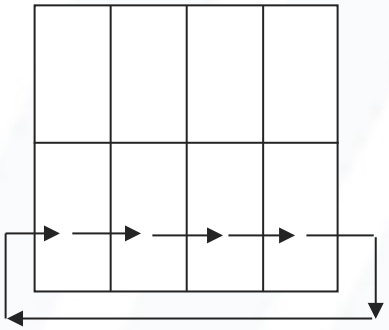
# De-aliasing for non-linear step



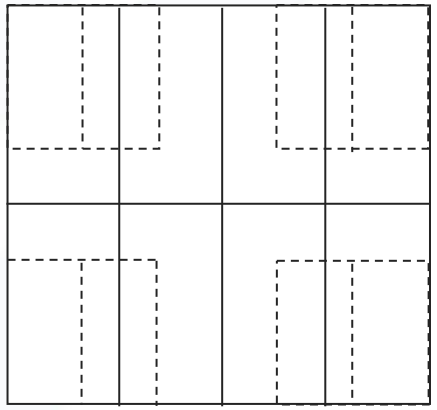
(Model A)

Interpolation

Model A



Fourier plane (m, n)

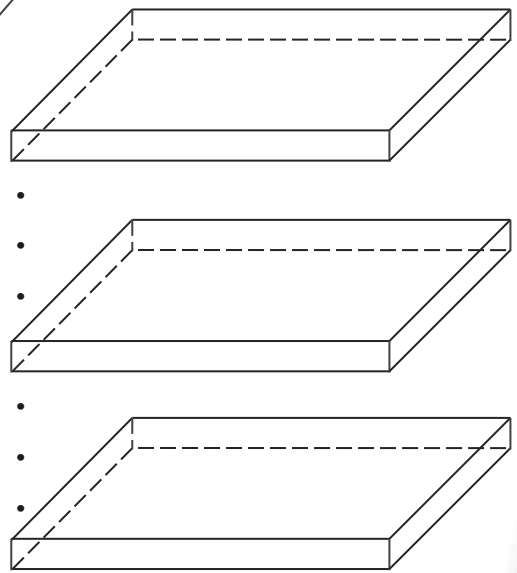


Extrapolation

Shift data

(Model B1)

Redistribute data
to planes on each processor



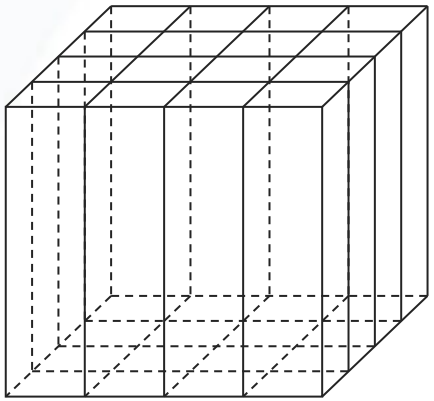
CPU I

CPU 10

CPU 16

Limitation

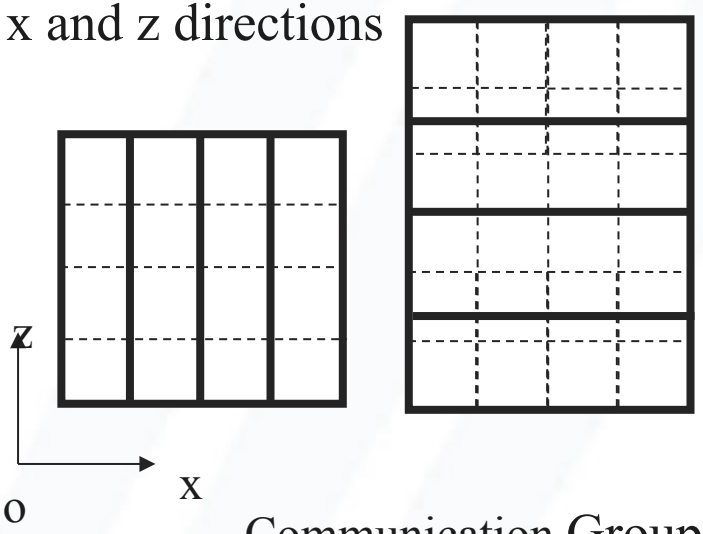
2D FFT



# De-aliasing for Non-linear Step



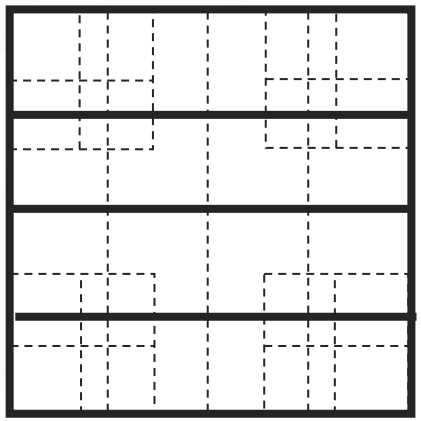
De-aliasing separately in x and z directions

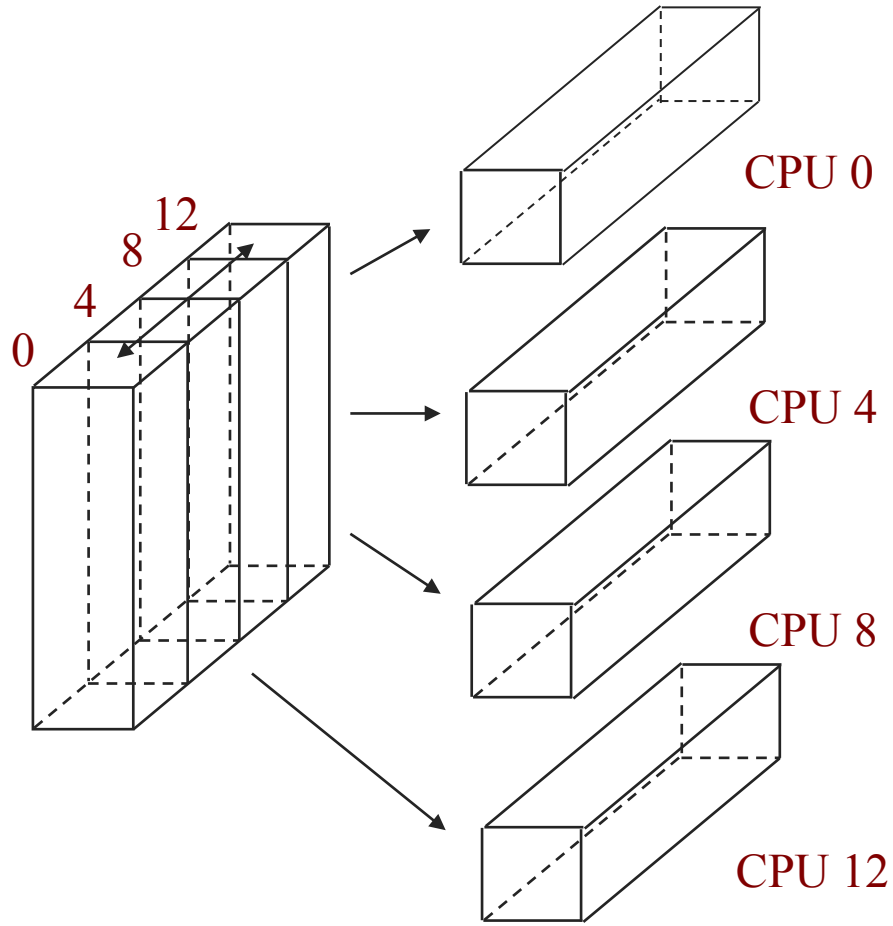



Communication Group x

(Model B2)

Communication Group z





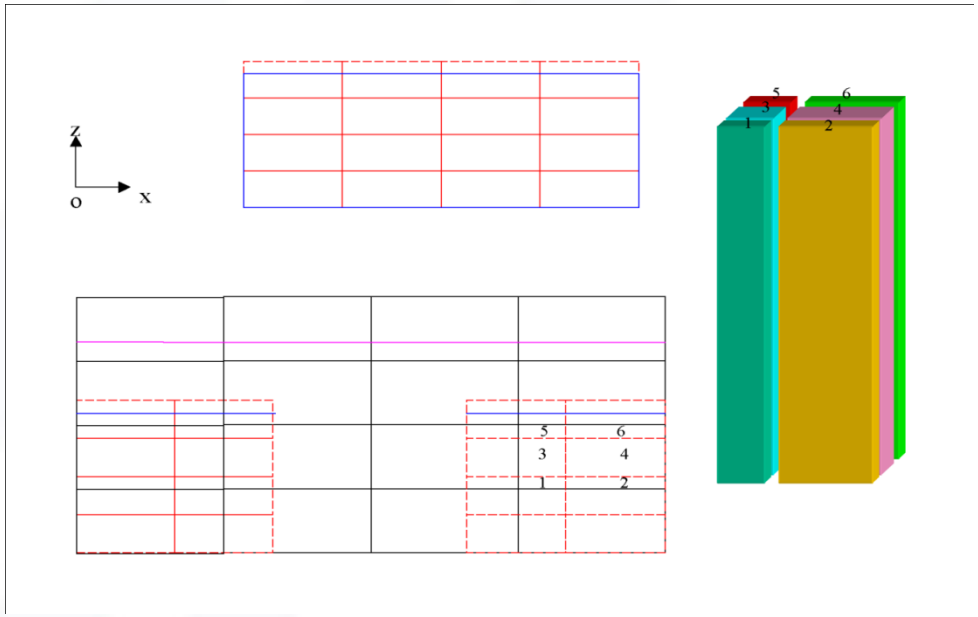

Redistribute data

# De-aliasing for Non-linear Step



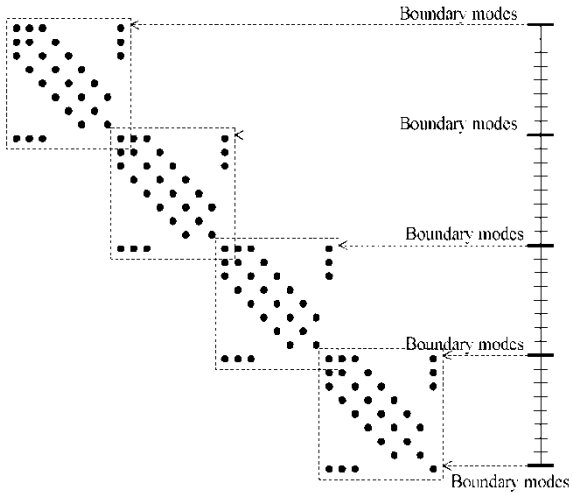
Complex FFT: from 2N real numbers to (N+1) complex numbers

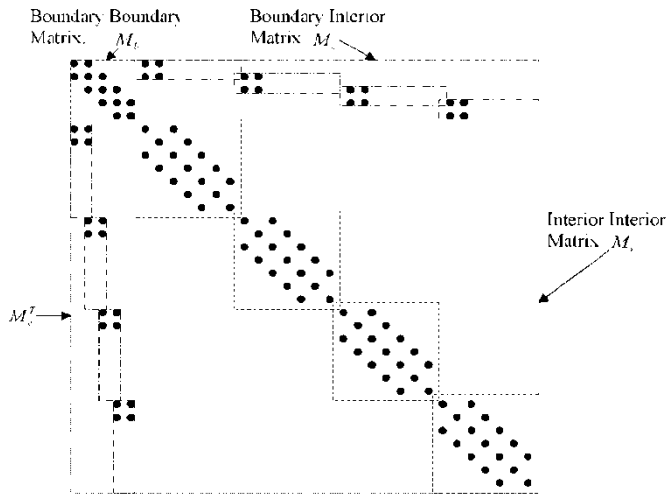
Sending data from all processors to their 6 neighbor processors




(Model B3)

Fig. 1D Helmholtz matrix



# Benchmarks for Model B and C



Table. Time of nonlinear step for model B2 with same number of points in the y direction

| Mesh | Procs. # | T(NL) | T(PR) | T(VIS) | T(tot) | T(NL)/T(tot) |
|---|---|---|---|---|---|---|
| $512\times513\times512$ | 512 | 40.61 | 11.53 | 2.64 | 54.78 | 74.1% |
| $768\times513\times768$ | 1024 | 40.64 | 12.51 | 3.06 | 56.21 | 72.3% |
| $1024\times513\times1024$ | 2048 | 52.89 | 13.26 | 2.65 | 68.8 | 76.9% |

Table. Time of nonlinear step for model C with same number of points in the y direction

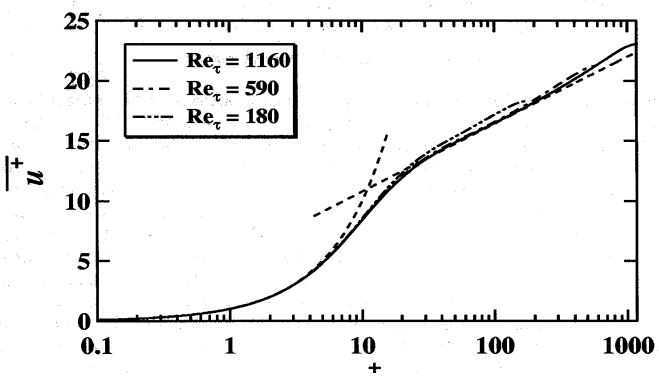
| Mesh | Procs. # | T(NL) | T(PR) | T(VIS) | T(tot) | T(NL)/T(tot) |
|---|---|---|---|---|---|---|
| $512\times513\times512$ | 512 | 20.56 | 6.26 | 8.15 | 34.97 | 58.8% |
| $768\times513\times768$ | 1024 | 18.68 | 6.62 | 8.92 | 35.22 | 55.9% |
| $1024\times513\times1024$ | 2048 | 18.19 | 6.18 | 8.38 | 32.75 | 55.5% |

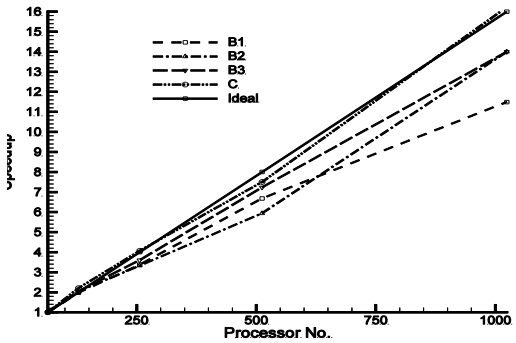
| $Re_\tau$ | Re | Domain | Mesh | Data |
|---|---|---|---|---|
| 180 | 4300 | $2\pi\times2\times2\pi$ | $128\times130\times128$ | 64M |
| 600 | 18000 | $2\pi\times2\times\pi$ | $384\times361\times384$ | 1.5G |
| 1000 | 27500 | $6\pi\times2\times1.5\pi$ | $768\times521\times768$ | 8.64G |

Table. Simulation parameters for DNS at different Reynolds number

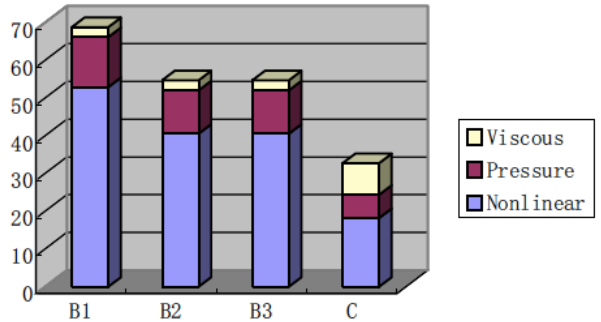
# Simulation Results

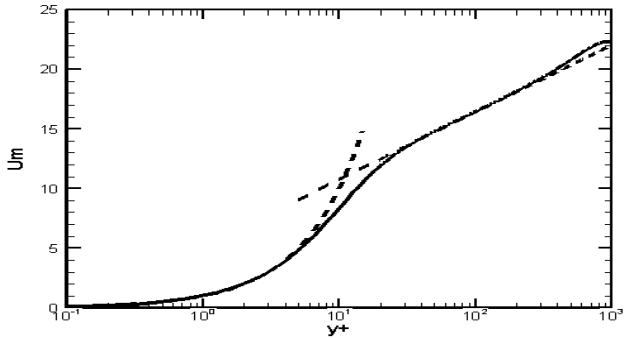


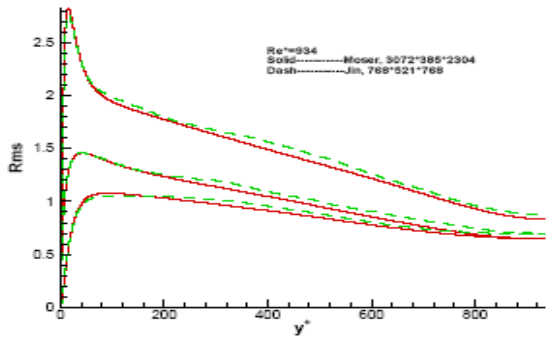



Comparison of model B and C




10510/136.8
～76

| Country | Computer | Cores | Year | Rmax | Rpeak |
|---|---|---|---|---|---|
| United States | BlueGene/L | 65536 | 2005 | 136.80 | 183.50 |
| RIKEN, Japan | K computer | 705024 | 2011 | 10510.00 | 11280.38 |

# Benchmark Results



| Case | Mesh | Racks | Processors | Elements | Pts/Elet | Time |
|---|---|---|---|---|---|---|
| A | $1024\times513\times1024$ | 1 | 2048 | 16 | 33 | 35 |
| B | $1024\times513\times1024$ | 2 | 2048 | 16 | 33 | 25.9 |
| C | $1024\times513\times1024$ | 4 | 8192 | 16 | 33 | 11.5 |
| D | $1024\times1025\times1024$ | 4 | 8192 | 32 | 33 | 24.92 |
| E | $1280\times1025\times1280$ | 4 | 8192 | 32 | 33 | 47.04 |
| F | $1536\times769\times1536$ | 4 | 8192 | 32 | 33 | 19.6 |
| G | $2048\times513\times2048$ | 4 | 8192 | 16 | 33 | 46.5 |
| H | $2048\times513\times2048$ | 8 | 16384 | 16 | 33 | 26.1 |
| I | $2048\times513\times2048$ | 16 | 16384 | 16 | 33 | 17.0 |
| J | $2048\times513\times2048$ | 16 | 32768 | 16 | 33 | 14.8 |
| K | $2048\times769\times2048$ | 8 | 16384 | 24 | 33 | 39.6 |
| L | $2048\times769\times2048$ | 16 | 16384 | 24 | 33 | 29.0 |
| M | $2048\times769\times2048$ | 16 | 32768 | 24 | 33 | 22.6 |
| N | $2048\times1025\times2048$ | 16 | 32768 | 16 | 65 | 33.9 |
| O | $2048\times1025\times2048$ | 16 | 32768 | 32 | 33 | 30.0 |

# Simulation Results



- *"Benchmarks on Tera-scalable Models for DNS of Channel Turbulent Flow"*, Parallel Computing. Vol. 33/12, 780-794, 2007.

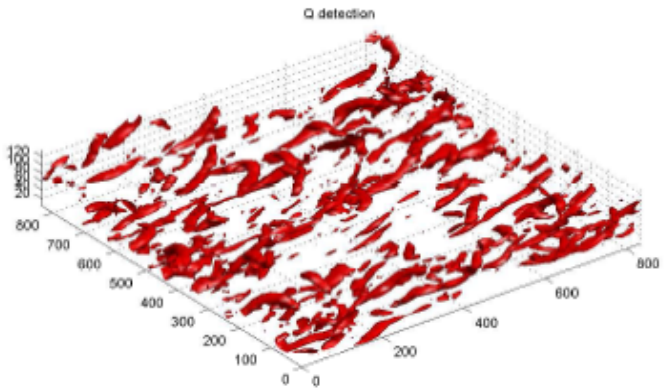
$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right),$$

$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right)$$

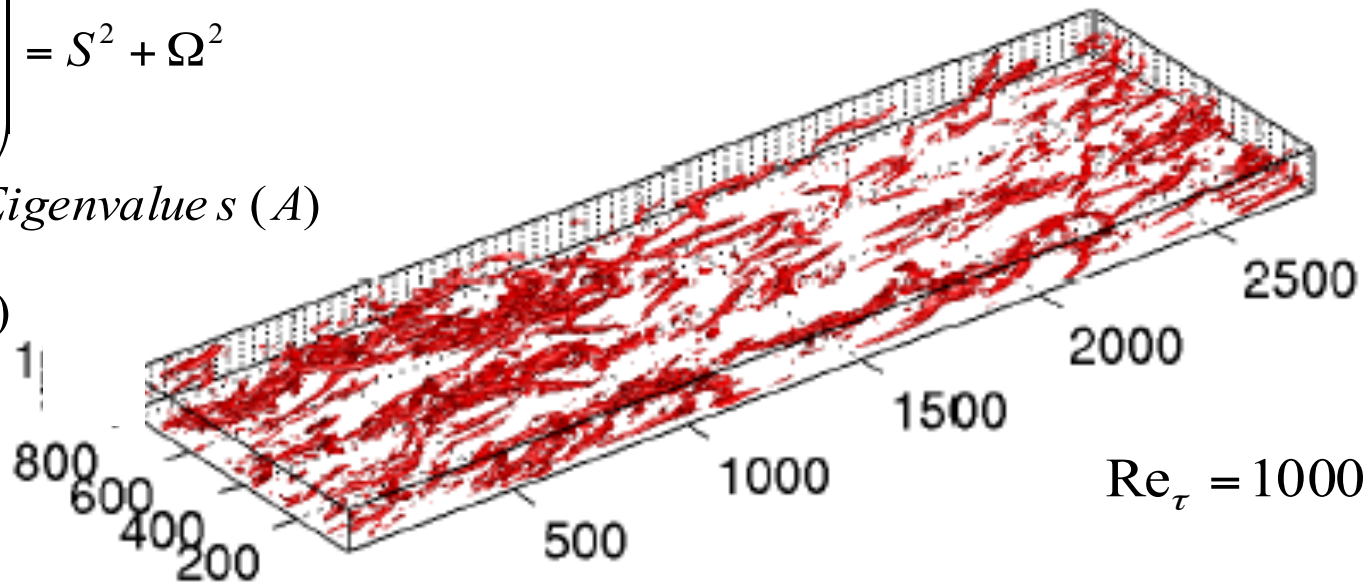
$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} = S^2 + \Omega^2$$

$$\lambda_3 < \lambda_2 < \lambda_1, \quad \lambda_i = Eigenvalues\,(A)$$

$$Q = -\frac{1}{2}(\lambda_1 + \lambda_2 + \lambda_3)$$




$\mathrm{Re}_\tau = 180$




$\mathrm{Re}_\tau = 1000$

Near Wall
Vortex Structure

# Simulation Results



Comparison of parameters for DNS at $Re_\tau = 1000$ or higher

| Case | $N_x \times N_y \times N_z$ | $Re_\tau$ | Element | $L_x$ | $L_z$ | $\Delta x^+$ | $\Delta y^+$ | $\Delta z^+$ |
|---|---|---|---|---|---|---|---|---|
| Moser [9] | $768 \times 769 \times 768$ | 1901 | 1 | $\pi$ | $\pi/2$ | 7.8 | 7.8 | 3.9 |
| Moser [9] | $3072 \times 385 \times 2304$ | 934 | 1 | $8\pi$ | $3\pi$ | 7.6 | 3.8 | 3.8 |
| Iwamoto [7] | $1152 \times 513 \times 1024$ | 1160 | 1 | $6\pi$ | $2\pi$ | 19 | – | 7.1 |
| Iwamoto [8] | $2304 \times 1025 \times 2048$ | 2320 | 1 | $6\pi$ | $2\pi$ | 9.5 | – | 3.6 |
| Hoyas [6] | $6144 \times 633 \times 4608$ | 2003 | 1 | $8\pi$ | $3\pi$ | 9.2 | 3.8 | 7.6 |
| 2005 ~~Present~~ | $768 \times 521 \times 768$ | 934 | 20 | $6\pi$ | $1.5\pi$ | 22.9 | 7.66 | 5.7 |

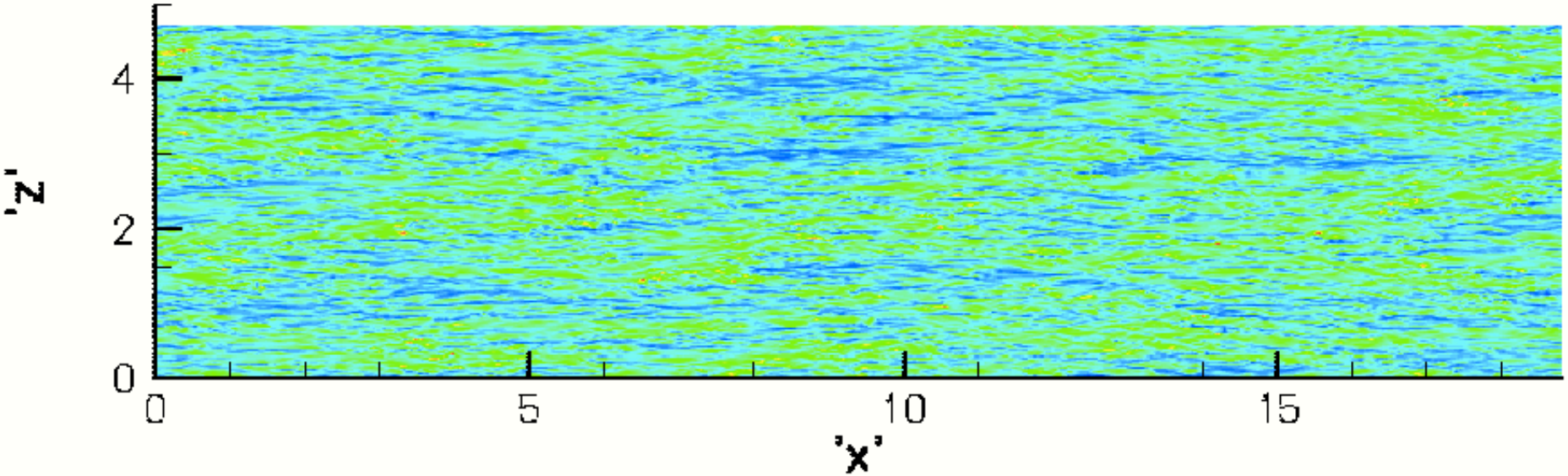



Fig. 9. Streaks at $y^+ = 5$ for $Re_\tau = 1000$, aixes are all in wall units

# Benefits of Extreme Computing



- Micro-bubbles:
  Force Coupling Method (FCM)

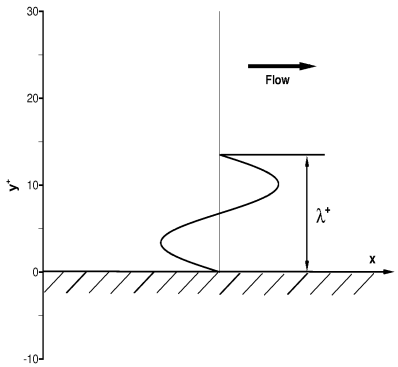
- *"Numerical Simulation of Turbulent Drag Reduction Using Micro-Bubbles"*, Journal of Fluid Mechanics, 468:271-281, 2002.

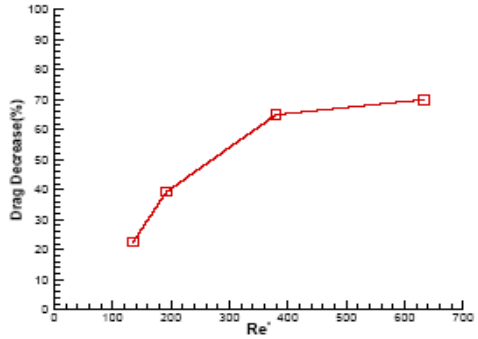
Force Monopole

$$\rho_F\left(\frac{\partial u_i}{\partial t}+u_j\frac{\partial u_i}{\partial x_j}\right)=-\frac{\partial p}{\partial x_i}+\mu\nabla^2 u_i+\sum_n F_i^n\Delta(\vec{x}-\vec{Y}^n)+\sum_n G_{ij}^n\frac{\partial\Theta}{\partial x_j}(\vec{x}-\vec{Y}^n)$$

Force Dipole

$$\Delta(\vec{x}-\vec{y})=(2\pi\sigma^2)^{-3/2}\exp((\vec{x}-\vec{y})^2/2\sigma^2)$$




- Constant Forcing

$$F_x(y)=-I\sin(\frac{2\pi}{\lambda}y),\quad 0\le y\le\lambda$$

$$F_x(y)=0,\quad y>\lambda$$

- *"Turbulent Drag Reduction by Constant Near-wall Forcing"*, Journal of Fluid Mechanics. 582, 79-101, 2007.

- Slip boundary condition

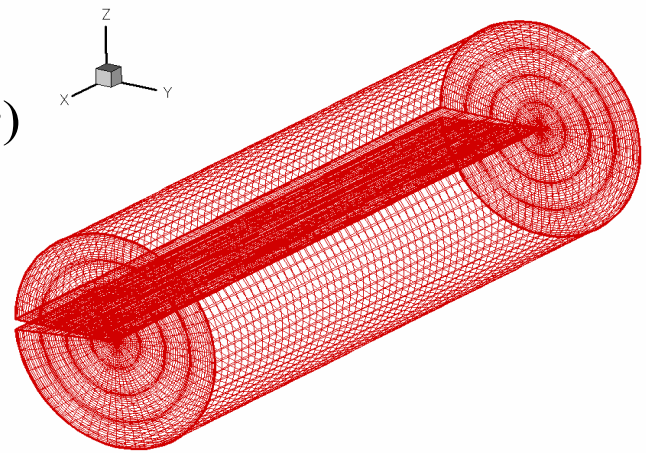
U = b * dU/dy

# DNS for Pipe Turbulence



- Numerical discretization

$$\phi(x,y,z,t)=\sum_{m=-M/2}^{M/2-1}\sum_{n=-N/2}^{N/2-1}\sum_{0}^{P}\phi(m,p,n,t)e^{-i\alpha mx}e^{-i\beta nz}P_p(y)$$




- Same time splitting scheme as in the channel

- Pressure step

$$\frac{d^2\hat{P}}{dr^2}+\frac{1}{r}\frac{d\hat{P}}{dr}-n^2\alpha^2\hat{P}-\frac{m^2}{r^2}\hat{P}=\frac{1}{\Delta t}\left(i\alpha n\hat{u}+\frac{im}{r}\hat{v}+\frac{d\hat{w}}{dr}+\frac{\hat{w}}{r}\right)^{n+1/3}$$

- Viscous step: Helmholtz equation

$$\frac{d^2\bar{u}}{dr^2}+\frac{1}{r}\frac{d\bar{u}}{dr}-n^2\alpha^2\bar{u}-\frac{m^2}{r^2}\bar{u}-\frac{\mathrm{Re}\,\gamma_0}{\Delta t}\bar{u}=\bar{f}_1$$

$$\frac{d^2\bar{v}}{dr^2}+\frac{1}{r}\frac{d\bar{v}}{dr}-n^2\alpha^2\bar{v}-\frac{(m+1)^2}{r^2}\bar{v}-\frac{\mathrm{Re}\,\gamma_0}{\Delta t}\bar{v}=\bar{f}_2$$

$$\frac{d^2\bar{w}}{dr^2}+\frac{1}{r}\frac{d\bar{w}}{dr}-n^2\alpha^2\bar{w}-\frac{(m-1)^2}{r^2}\bar{w}-\frac{\mathrm{Re}\,\gamma_0}{\Delta t}\bar{w}=\bar{f}_3$$

$$\bar{u}=\hat{u}\qquad \bar{v}=\hat{v}-i\hat{w}\qquad \bar{w}=\hat{v}+i\hat{w}$$

$$\hat{u}=\bar{u}\qquad \hat{v}=\frac{\bar{v}+\bar{w}}{2}\qquad \hat{w}=\frac{\bar{v}-\bar{w}}{2}i$$

# Numerical and Parallel Methods



$$\phi(x,y,z,t)=\sum_{m=-M/2}^{M/2-1}\sum_{n=-N/2}^{N/2-1}\sum_{0}^{P}\phi(m,p,n,t)e^{-i\alpha mx}e^{-i\beta nz}P_p(y)$$

Space Charge Effect:

$$\Delta\phi=\frac{\partial^2\phi}{\partial x^2}+\frac{\partial^2\phi}{\partial y^2}+\frac{\partial^2\phi}{\partial z^2}=-\frac{\rho}{\varepsilon_0}$$

$$=\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial\phi}{\partial r})+\frac{1}{r^2}\frac{\partial^2\phi}{\partial\theta^2}+\frac{\partial^2\phi}{\partial z^2}$$

$$A\cdot\tilde{\varphi}=\hat{f},\quad A=K+M+N$$

$$K_{pq}=\int_0^1 r\frac{\partial\varphi_p(y)}{\partial r}\frac{\partial\varphi_q(y)}{\partial r}dr$$

$$M_{pq}=m^2\int_0^1\frac{1}{r}\varphi_p(y)\varphi_q(y)dr$$

$$N_{pq}=n^2\int_0^1 r\varphi_p(y)\varphi_q(y)dr$$

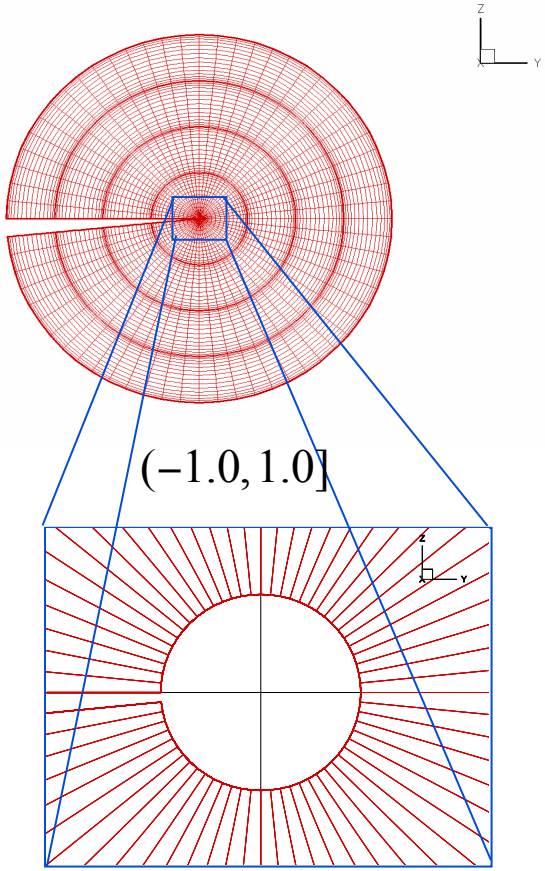
$$\phi(x,0,z)<\infty$$

$$\left.\frac{\partial\phi}{\partial r}\right|_{r=0}=0,\quad \phi(x,1,z)=0$$

$$\phi(0,y,z)=\phi(2\pi,y,z)$$

$$\phi(x,y,0)=\phi(x,y,2\pi)$$

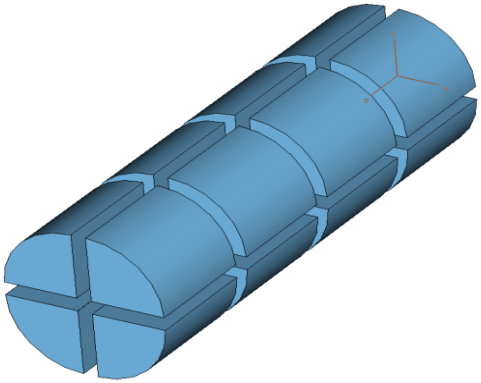
$$P_p(y)=\begin{cases}(\frac{1-y}{2}) & p=0\\ (\frac{1-y}{2})(\frac{1+y}{2})P_{p-1}^{1,1}(y) & 0<p<P\\ (\frac{1+y}{2}) & p=P\end{cases}$$

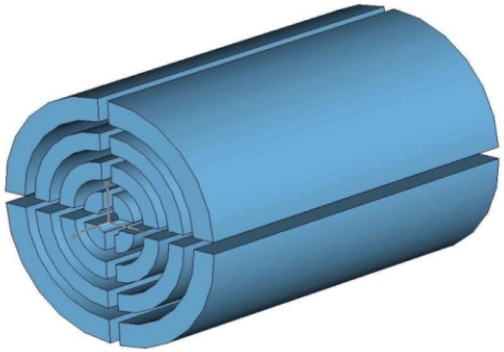


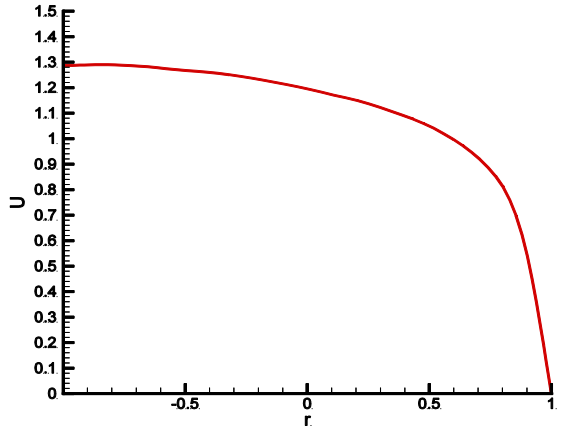

Gauss-Lobatto-Radau

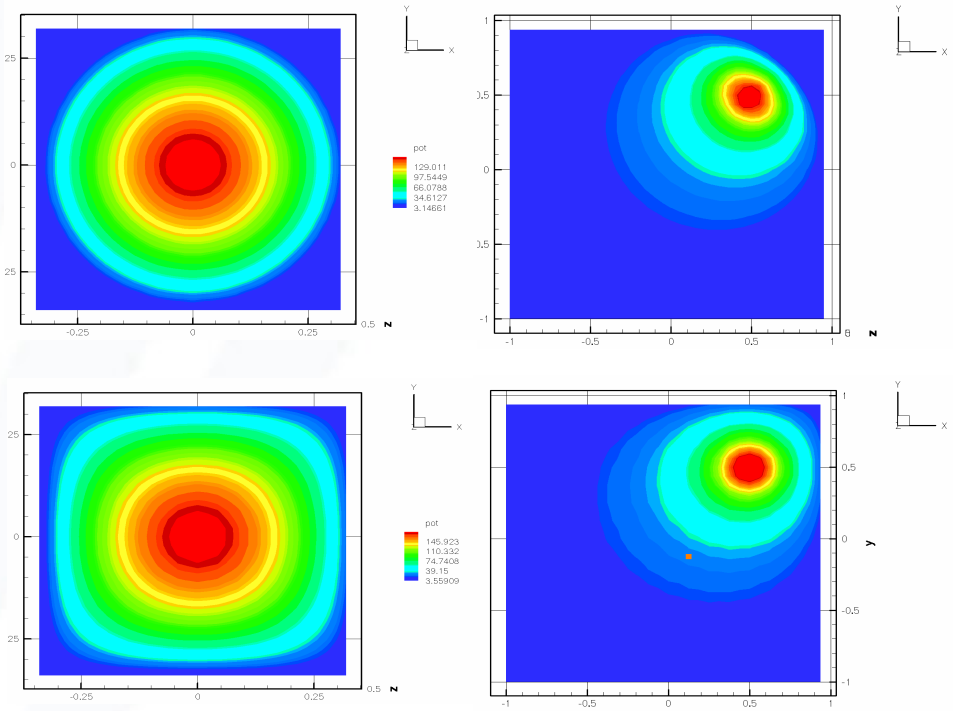
# Cylindrical Coordinate System

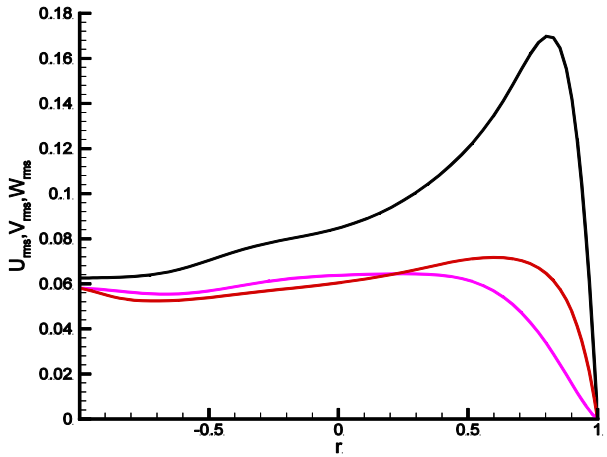






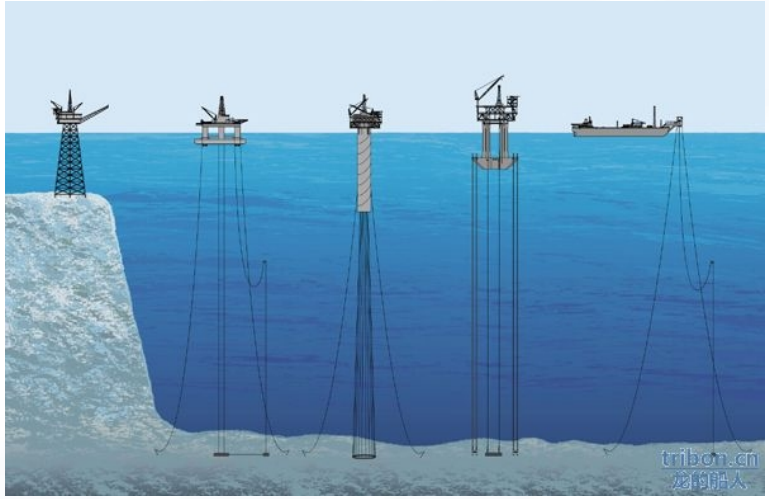


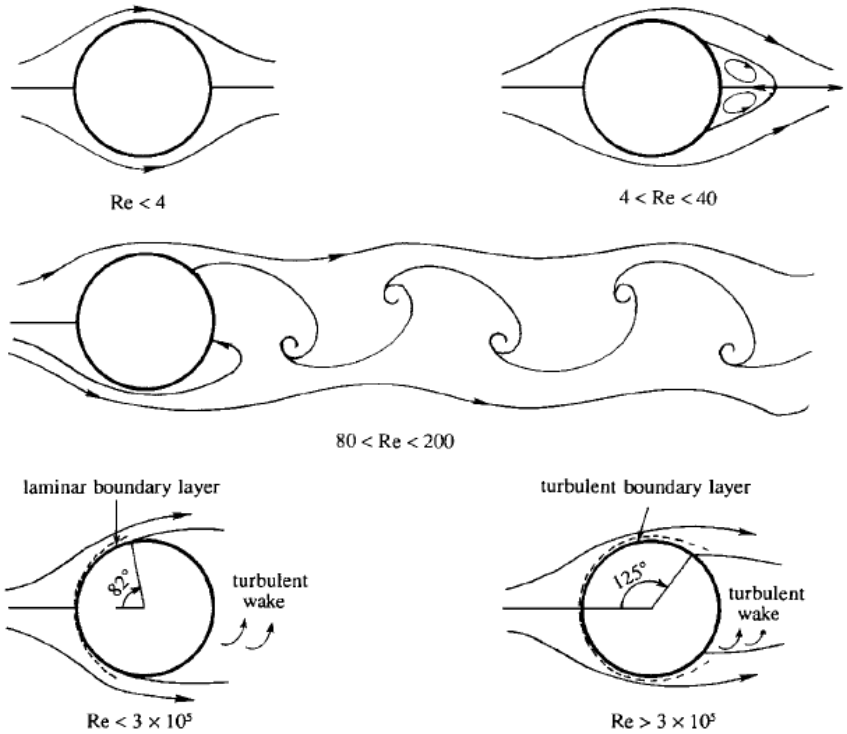





# Flow Past Cylinder







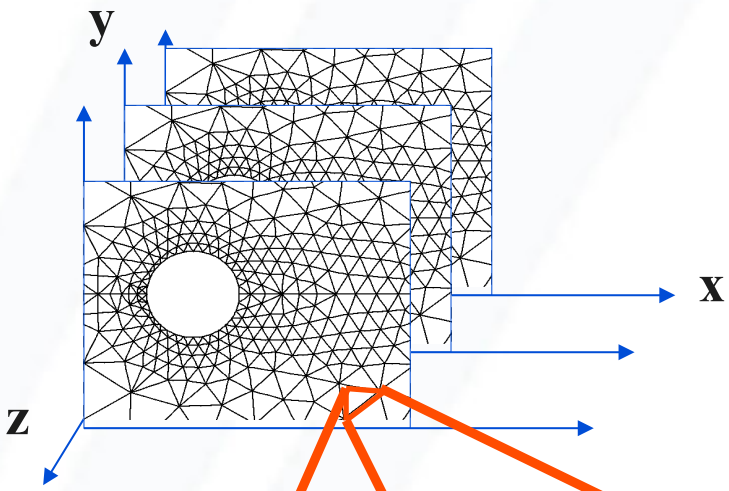

Flow past cylinder has many applications in industry, such as oil pipe in ocean and cable, chimney in the wind.

# Fourier *hp*-Finite Element 2D Code



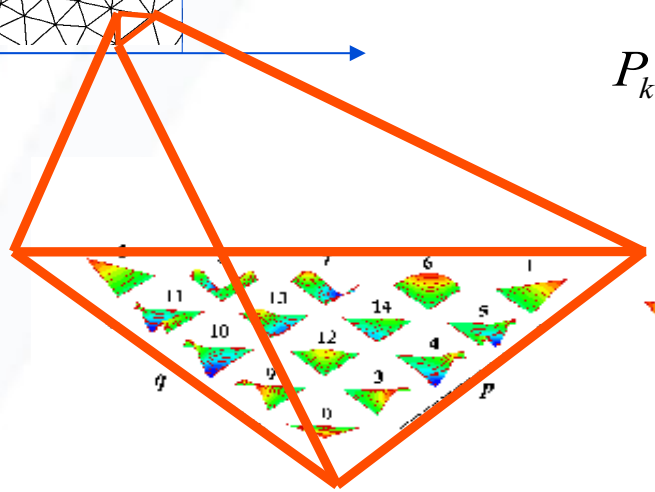
$$P_{l,k}(x,y) = P_l(x) * P_k(y)$$

$$u(x,y,z,t) = \sum_{m=-M/2}^{M/2-1} \sum_{l} \sum_{k} u(m,l,k,t) e^{-i\alpha m x} P_{l.k}(x,y)$$

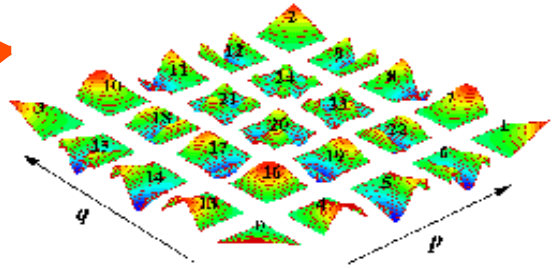



$$P_l(x) = \begin{cases} (\frac{1-x}{2}) & l = 0 \\ (\frac{1-x}{2})(\frac{1+x}{2}) P_{l-1}^{1,1}(x) & 0 < l < P_1 \\ (\frac{1+x}{2}) & l = P_1 \end{cases}$$

$$P_k(y) = \begin{cases} (\frac{1-y}{2}) & k = 0 \\ (\frac{1-y}{2})(\frac{1+y}{2}) P_{k-1}^{1,1}(y) & 0 < k < P_2 \\ (\frac{1+y}{2}) & k = P_2 \end{cases}$$



Triangular

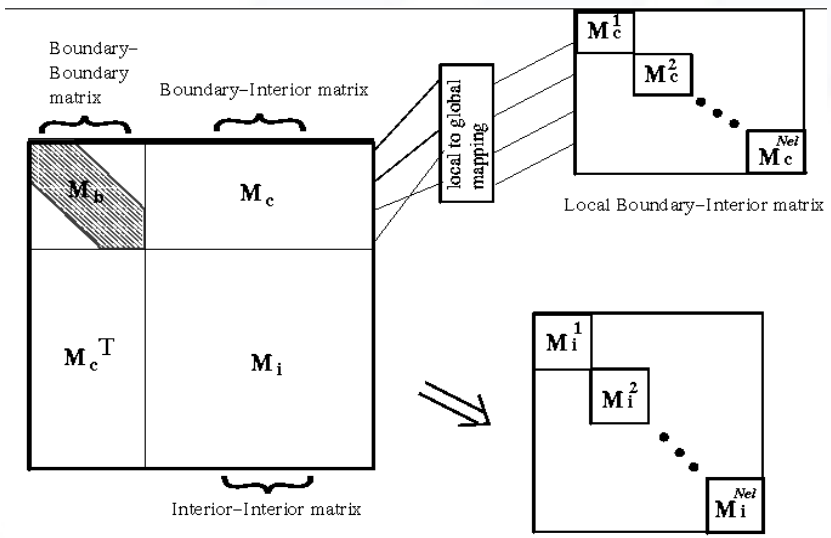


Quadrilateral

(Originally started at Brown Univ. )

# Static Condensation






$Du = f \qquad Lu = f$

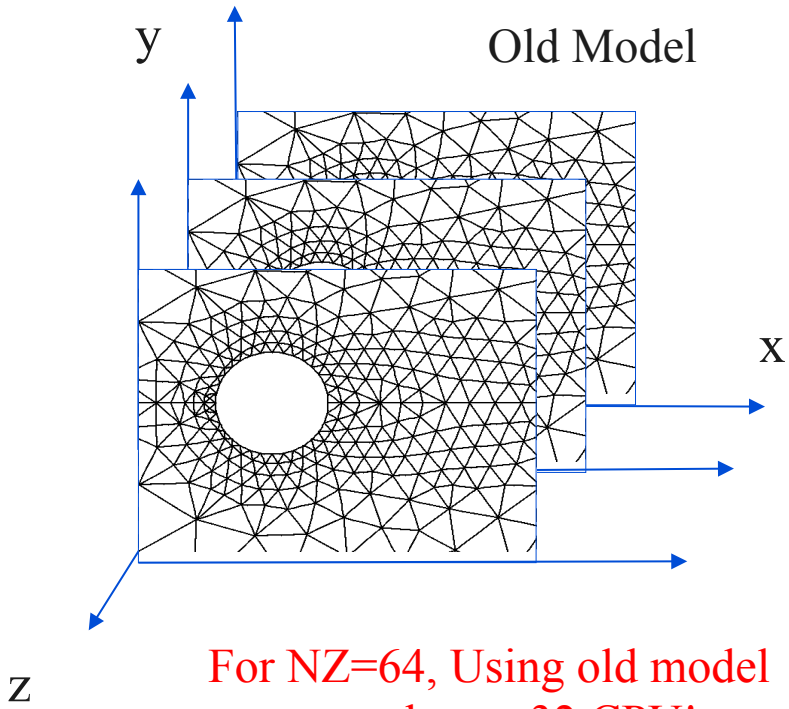
within every elemental domain

$$\begin{pmatrix} L_{bb} & L_{ib} \\ L_{bi} & L_{ii} \end{pmatrix} \begin{pmatrix} u_b \\ u_i \end{pmatrix} = \begin{pmatrix} f_b \\ f_i \end{pmatrix}$$

$L_{ii}$ are decoupled and so easy to invert and construct the Schur complement:

$$S\, u_b = [L_{bb} - L_{bi}\, L^{-1}_{ii}\, L_{ib}]\, u_b = f_b - L_{bi}\, L^{-1}_{ii}\, f_i$$

Schur complement is constructed at an elemental level then use PCG




For NZ=64, Using old model can only use 32 CPU!

Advantages:

- Easy implementation,
- High parallel efficiency

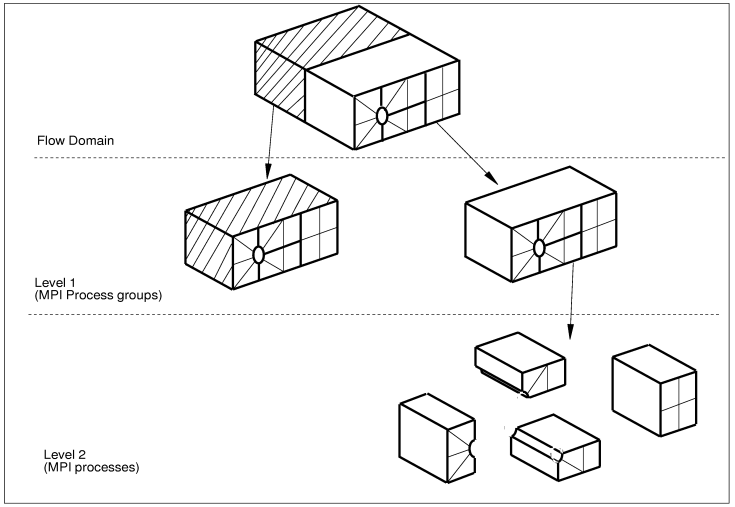
Disadvantages:

- Maximum CPU is limited Since each processor must have 2 planes!

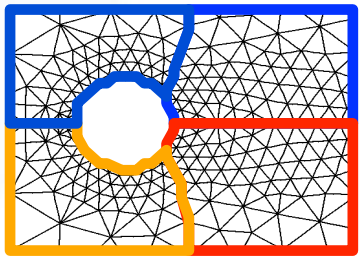
# MPI/MPI Two-Level Parallelism



New Model




Mesh partition



## Parallel Iterative Solver(PCG)

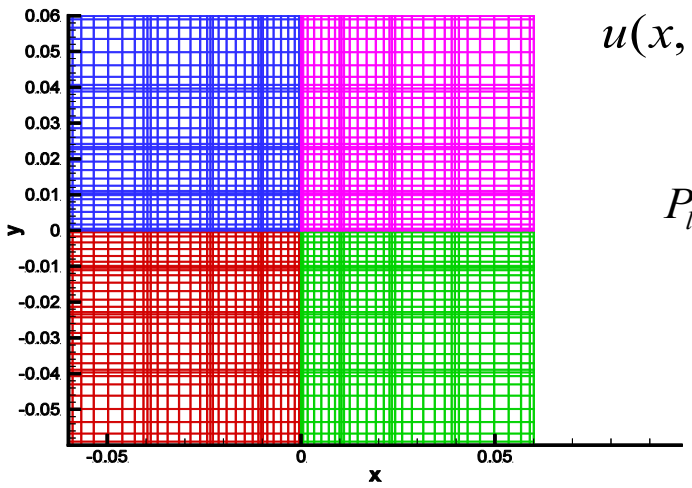
$$
\begin{aligned}
&For\ j = 1 \rightarrow n \\
&\quad \alpha_j \leftarrow r_{j-1}^T r_{j-1} / q_j^T A q_j \\
&\quad x_j \leftarrow x_{j-1} + \alpha_j q_j \\
&\quad r_j \leftarrow r_{j-1} - \alpha_j A q_j \\
&\quad If\ \|r_j\| < \varepsilon \\
&\qquad set\ x = x_j \\
&\quad End\ If \\
&\quad \beta_j \leftarrow r_j^T r_j / r_{j-1}^T r_{j-1} \\
&\quad q_{j+1} \leftarrow r_j + \beta_j q_j \\
&End\ For
\end{aligned}
$$

Parallel

Serial

# Parallel Poisson solver on structured grid

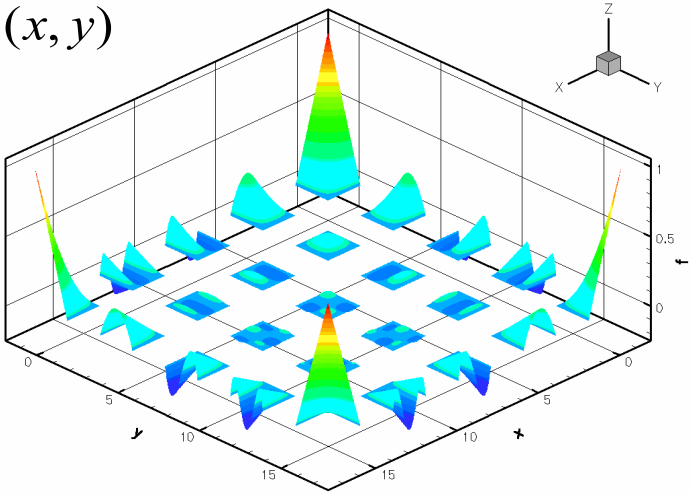





$$u(x,y,t) = \sum_l \sum_k u(l,k,t) P_{l.k}(x,y)$$

$$P_{l,k}(x,y) = P_l(x) * P_k(y)$$

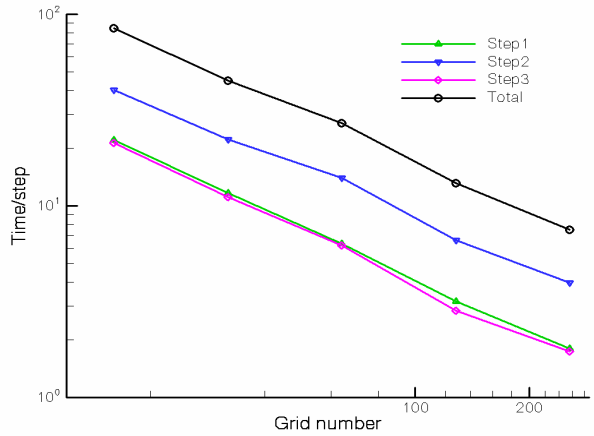
Parallel 2D
Poisson Solver




$$P_l(x) = P_l(x) = \begin{cases} \left(\frac{1-x}{2}\right) & l = 0 \\ \left(\frac{1-x}{2}\right)\left(\frac{1+x}{2}\right) P_{l-1}^{1,1}(x) & 0 < l < P_1 \\ \left(\frac{1+x}{2}\right) & l = P_1 \end{cases}$$

Table 2: Strong scaling of 2D iterative Poisson solver (E=64, P=4)

| CPU | 16 | 64 | 256 | 1024 | 4096 |
|---|---|---|---|---|---|
| Time (s) | 286 | 68 | 17.2 | 4.08 | 1.66 |
| Parallel Efficiency | 1.0 | 1.0 | 1.0 | 1.0 | 0.673 |



# Internal Penalty Method



Let $\vec{n}^+$ and $\vec{n}^-$ be the corresponding normal vectors on triangle edges. Let $\phi$ be a smooth function inside each element k and let us denote by $\phi^{\pm}$ the trace of $\phi$ on edges from the interior of $k^{\pm}$.

$$-\Delta u = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega,$$

We seek the discontinuous approximation $\phi_h$ to the exact solution $\phi$ in $P^N_{\Omega_k}$ such that

$$\sigma = \nabla u, \quad -\nabla\cdot\sigma = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega,$$

$$V_h = \{v \in L^2(\Omega) \quad | \quad v \in P(K) \quad \forall K \in T_h\}$$

$$\Sigma_h = \{v \in (L^2(\Omega))^2 \quad | v \in \Sigma(K) \quad \forall K \in T_h\}$$

Find $u_h \in V_h$ and $\sigma_h \in \Sigma_h$ such that $\forall K \in T_h$,

$$\int_K \sigma_h \cdot \tau dx = -\int_K u_h \nabla\cdot\tau dx + \sum_{e\subset\partial K}\int_e h_u^{e,K} n_K \cdot \tau ds \qquad \forall \tau \in \Sigma(K)$$

$$\int_K \sigma_h \cdot \nabla v dx = \int_K f v dx + \sum_{e\subset\partial K}\int_e h_\sigma^{e,K} n_K v ds \qquad \forall v \in P(K)$$

Equivalent

$$a(\phi_h,\varphi_h) = F(\varphi_h) \quad \forall \varphi_h \in P^N_{\Omega_k}; \qquad \varphi_h = 0 \ on\ \partial\Omega.$$

$$a(\phi_h,\varphi_h) = \sum_k \int_{\Omega_k} \nabla\phi\cdot\nabla\varphi d\vec{x} - \int_\Gamma ([[\varphi]]\cdot\nabla\phi + [[\phi]]\cdot\nabla\varphi)ds + \int_\Gamma \mu[[\phi]]\cdot[[\varphi]]ds$$

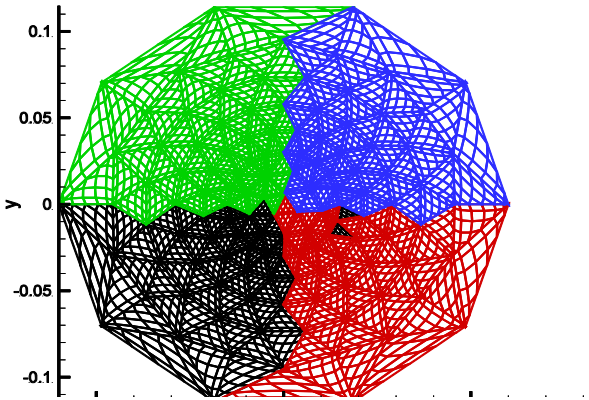
$$F(\varphi) = \sum_k \int_{\Omega_k} f\varphi d\vec{x}$$

$$[[\phi]] = \phi^+\vec{n}^+ + \phi^-\vec{n}^-,$$

$$\{\phi\} = (\phi^+ + \phi^-)/2$$

# Parallel Poisson solver on unstructured mesh






$$u_N(x_j) = \sum_{i=1}^{N} \hat{u}_i \Phi_i(x_j) = \sum_{i=1}^{N} u_i L_i(x_j)$$

$$L_j(x) = \sum_{k=1}^{N} V_{kj}^{-1} \Phi_k(x)$$

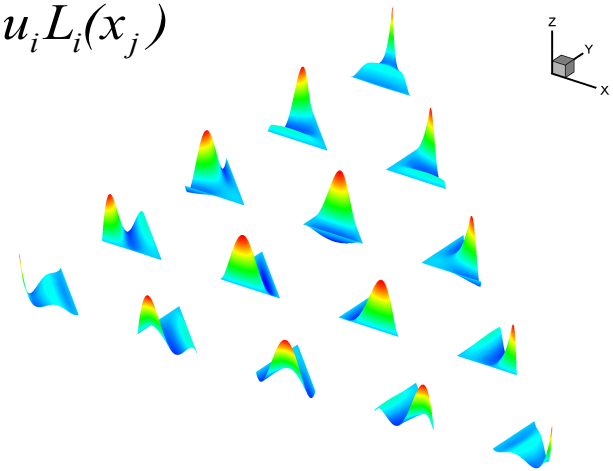
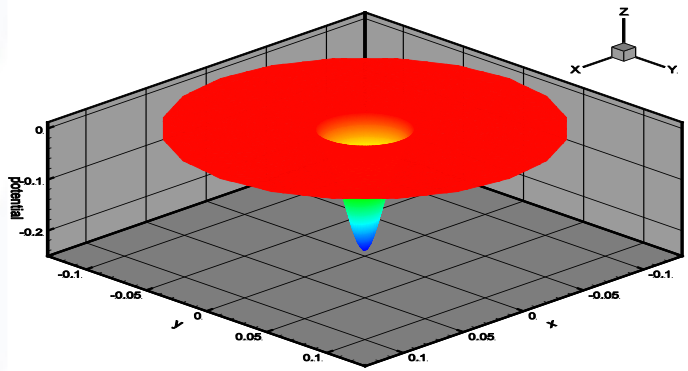
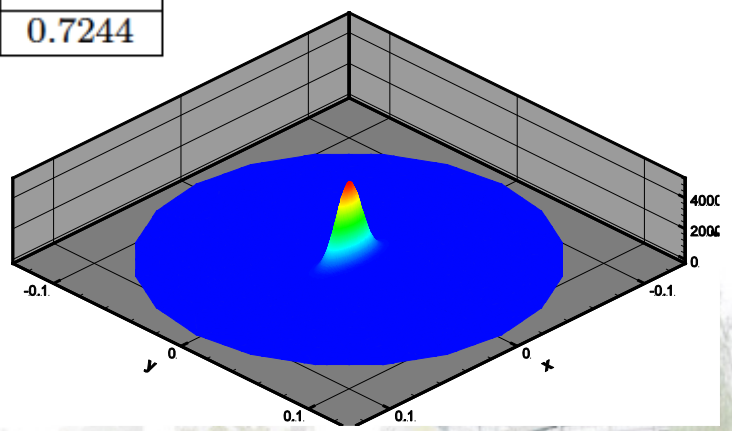



TABLE 1

*Strong scaling of 2D iterative Poisson solver* ($E = 1586$, $P$ :

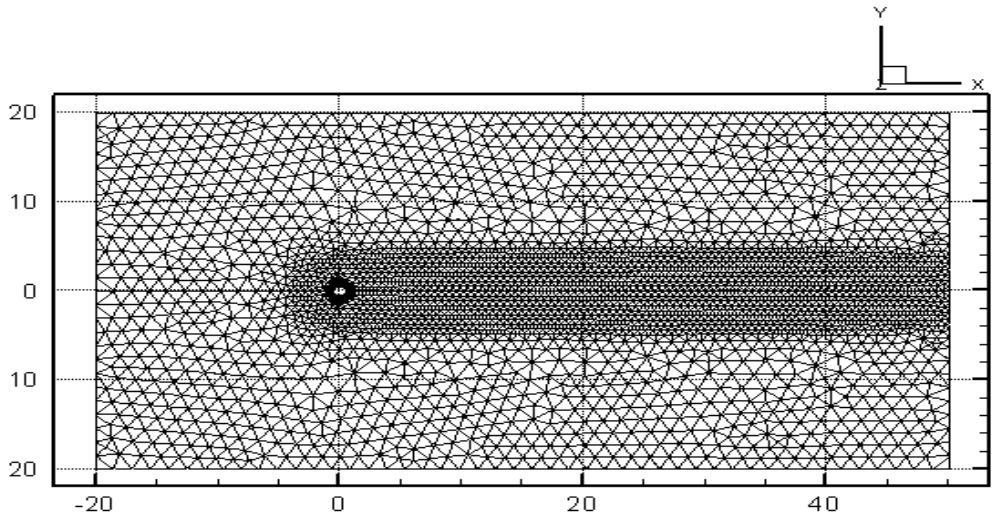
| CPU | 4 | 8 | 16 | 32 | 64 |
|---|---|---|---|---|---|
| CG solver time (s) | 1783.4 | 397.6 | 172.5 | 88.4 | 41.2 |
| DG solver time (s) | 1213.7 | 610.3 | 321.5 | 197.2 | 104.7 |
| Speedup | 1.0 | 1.989 | 3.775 | 6.155 | 11.59 |
| Parallel efficiency | 1.0 | 0.9945 | 0.94375 | 0.769375 | 0.7244 |



# Simulation Results






## Flow Pass Cylinder at Re=10,000

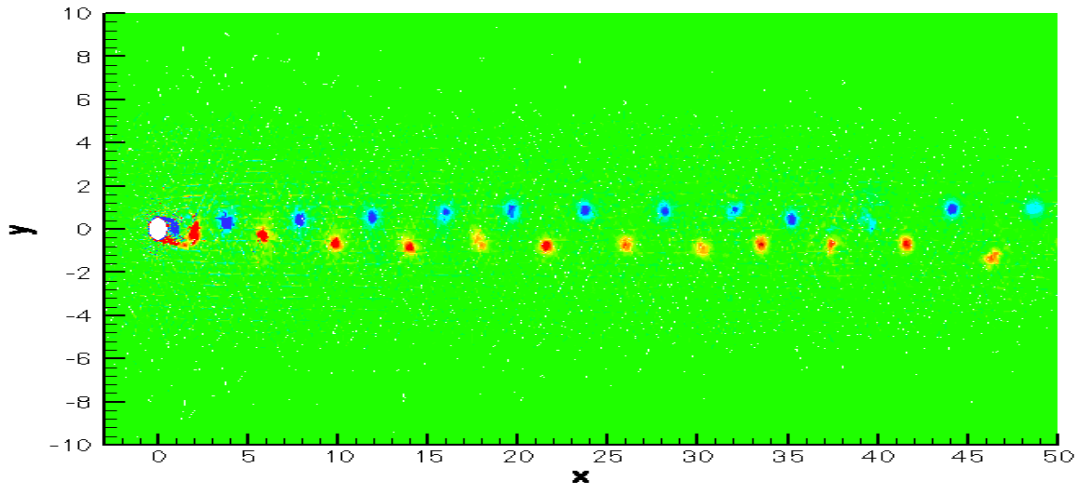
6272 Elements, P=5, 64 Fourier planes

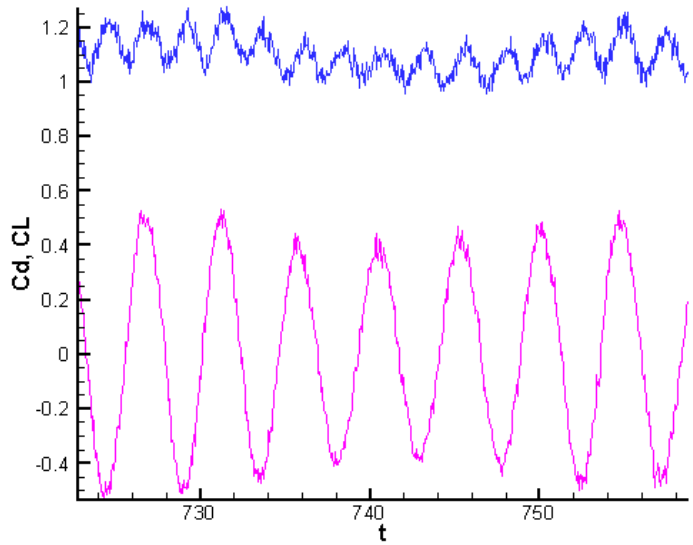
Drag coefficient

$$C_D = \frac{Drag}{\rho D U^2 / 2}$$

Lift coefficient

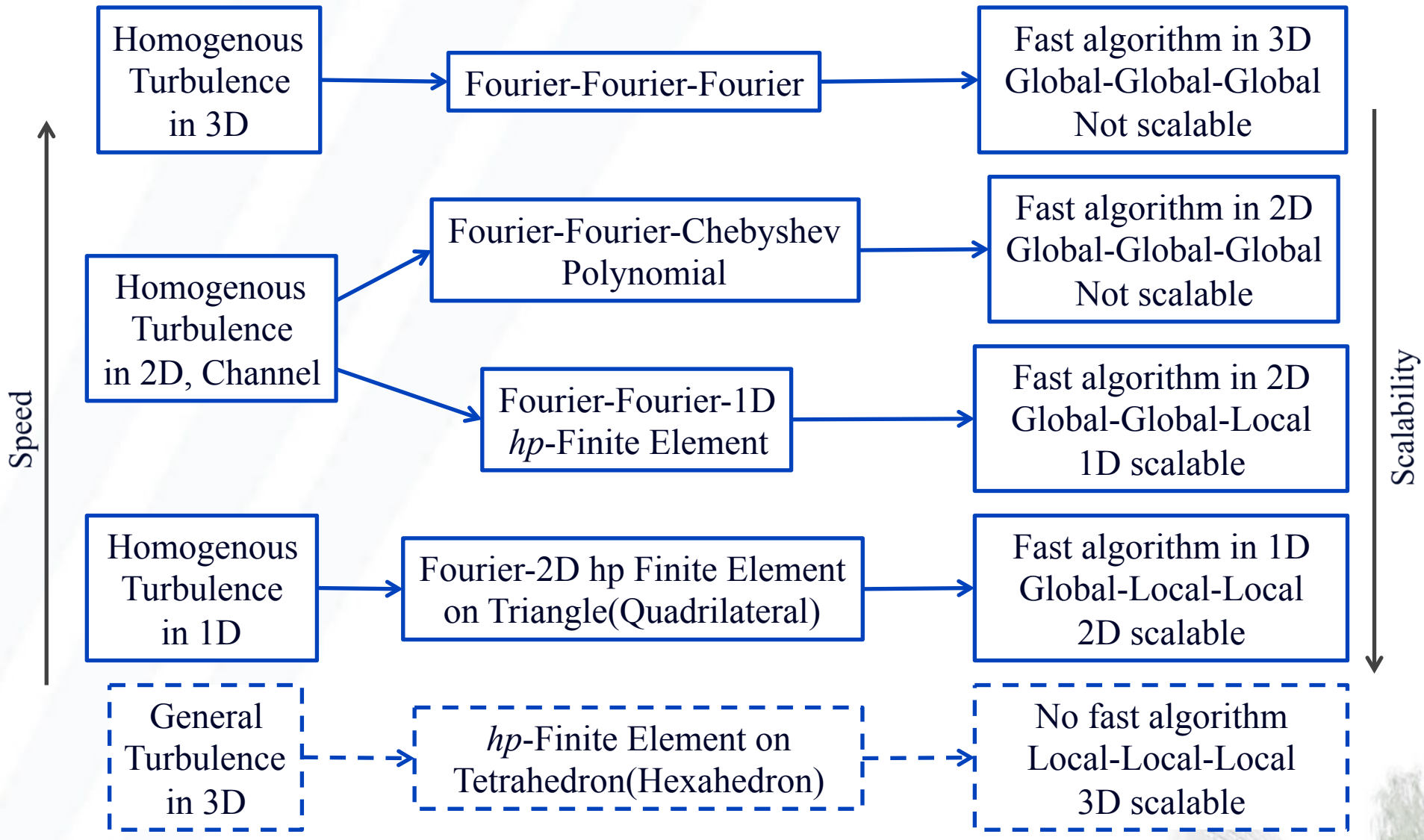
$$C_L = \frac{Lift}{\rho D U^2 / 2}$$



# Speed and Scalability





# Spherical domain



$$\rho\left(\frac{\partial u_r}{\partial t}+u_r\frac{\partial u_r}{\partial r}+\frac{u_\theta}{r\sin(\varphi)}\frac{\partial u_r}{\partial \theta}+\frac{u_\varphi}{r}\frac{\partial u_r}{\partial \varphi}-\frac{u_\theta^2+u_\varphi^2}{r}\right)=-\frac{\partial p}{\partial r}+\rho g_r+\cdots\cdots$$

Under development

$$\rho\left(\frac{\partial u_\theta}{\partial t}+u_r\frac{\partial u_\theta}{\partial r}+\frac{u_\theta}{r\sin(\varphi)}\frac{\partial u_\theta}{\partial \theta}+\frac{u_\varphi}{r}\frac{\partial u_\theta}{\partial \varphi}+\frac{u_r u_\theta+u_\theta u_\varphi\cot(\varphi)}{r}\right)=-\frac{1}{r\sin(\varphi)}\frac{\partial p}{\partial \theta}+\rho g_\theta+\cdots\cdots$$

$$\rho\left(\frac{\partial u_\varphi}{\partial t}+u_r\frac{\partial u_\varphi}{\partial r}+\frac{u_\theta}{r\sin(\varphi)}\frac{\partial u_\varphi}{\partial \theta}+\frac{u_\varphi}{r}\frac{\partial u_\varphi}{\partial \varphi}+\frac{u_r u_\varphi-u_\theta^2\cot(\varphi)}{r}\right)=-\frac{1}{r}\frac{\partial p}{\partial \varphi}+\rho g_\varphi+\cdots\cdots$$

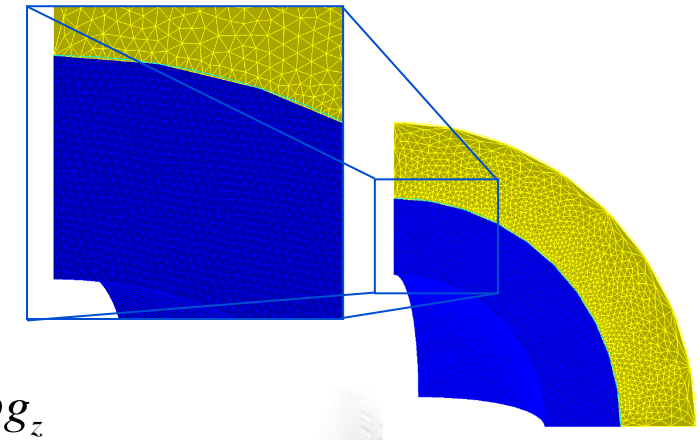
$$\frac{1}{r^2}\frac{\partial}{\partial r}(r^2u_r)+\frac{1}{r\sin(\phi)}\frac{\partial u_\phi}{\partial \theta}+\frac{1}{r\sin(\phi)}\frac{\partial}{\partial \phi}(\sin(\phi)u_\phi)=0$$

---

$$\rho\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}\right)=-\frac{\partial p}{\partial x}+\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)+\rho g_x$$

$$\rho\left(\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}\right)=-\frac{\partial p}{\partial y}+\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right)+\rho g_y$$

$$\rho\left(\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}\right)=-\frac{\partial p}{\partial z}+\mu\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right)+\rho g_z$$



# Parallel Poisson solver on unstructured grid



- Continuous Galerkin Method
- Nodal Base
- 3D Tetrahedron
- Static Condensation
- Conjugate Gradient Method
- Parallel implementation through MPI
- Domain partition by ParMetis or manually
- Mesh generated by CUBIT
- 3~10 Polynomial orders
- Dirichlet Boundary Condition

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = f(x,y,z)$$

$$\int_V f \cdot \varphi dv = \int_V \nabla^2 u \cdot \varphi dv$$

$$= \int_V [\nabla(\nabla u \varphi) - \nabla u \cdot \nabla \varphi] dv$$

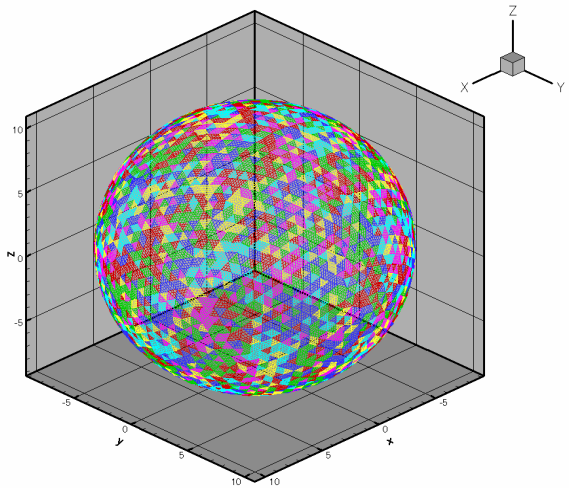
$$= \int_{\partial V} \vec{n} \cdot (\nabla u \varphi) dS - \int_V \nabla u \cdot \nabla \varphi dv$$

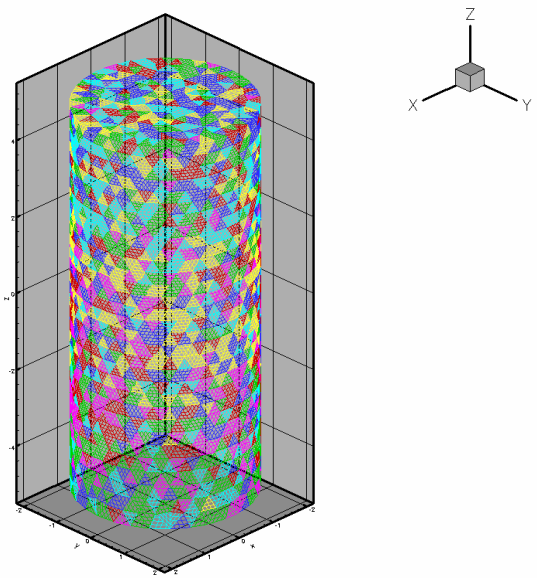
$$\int_V \nabla u \cdot \nabla \varphi dv = \int_{\partial V} \vec{n} \cdot (\nabla u \varphi) dS - \int_V f \cdot \varphi dv$$

$$\int_{\partial V} \vec{n} \cdot (\nabla u \varphi) dS = \int_{\partial V} (n_x \varphi \frac{\partial u}{\partial x} + n_y \varphi \frac{\partial u}{\partial y} + n_z \varphi \frac{\partial u}{\partial z}) dS$$

$$u = \sin(x) \cdot \sin(y) \cdot \sin(z)$$

$$f = -3 \cdot \sin(x) \cdot \sin(y) \cdot \sin(z)$$



# Conclusion



- Extreme Computing is necessary for direct numerical simulation of turbulence, because the cost is proportional to $Re^3$;
- High order numerical methods are necessary for turbulence DNS to reduce the size of the mesh;
- Although FFT can achieve faster speed, the scalability is not good as the Fourier expansions are global functions;
- Numerical methods with local expansions usually have better scalability than those with global expansions;
- Benefits of extreme computing have been shown for DNS of channel turbulence;
- Domain decomposition method has been used to make use of large number of processors;
- We are going to add real GPU to accelerate our simulations in the future!

# Thanks !