A Preview of Theory and Numeric for Large Eddy Simulation

By

Iftekhar Zaheer Naqavi

Advisors
Dr. E. Savory & Dr. R.J. Martinuzzi

Advanced Fluid Mechanics Research Group
Department of Mechanical and Materials Engineering
The University of Western Ontario
June 2004

Overview:

- **Flow modeling**
- **Tackling turbulence**
- **Why LES**
- **Brief history of LES**
- **Navier-Stokes equations**
- **Space filtering**
- **Numerical solution**
- **Few test results**

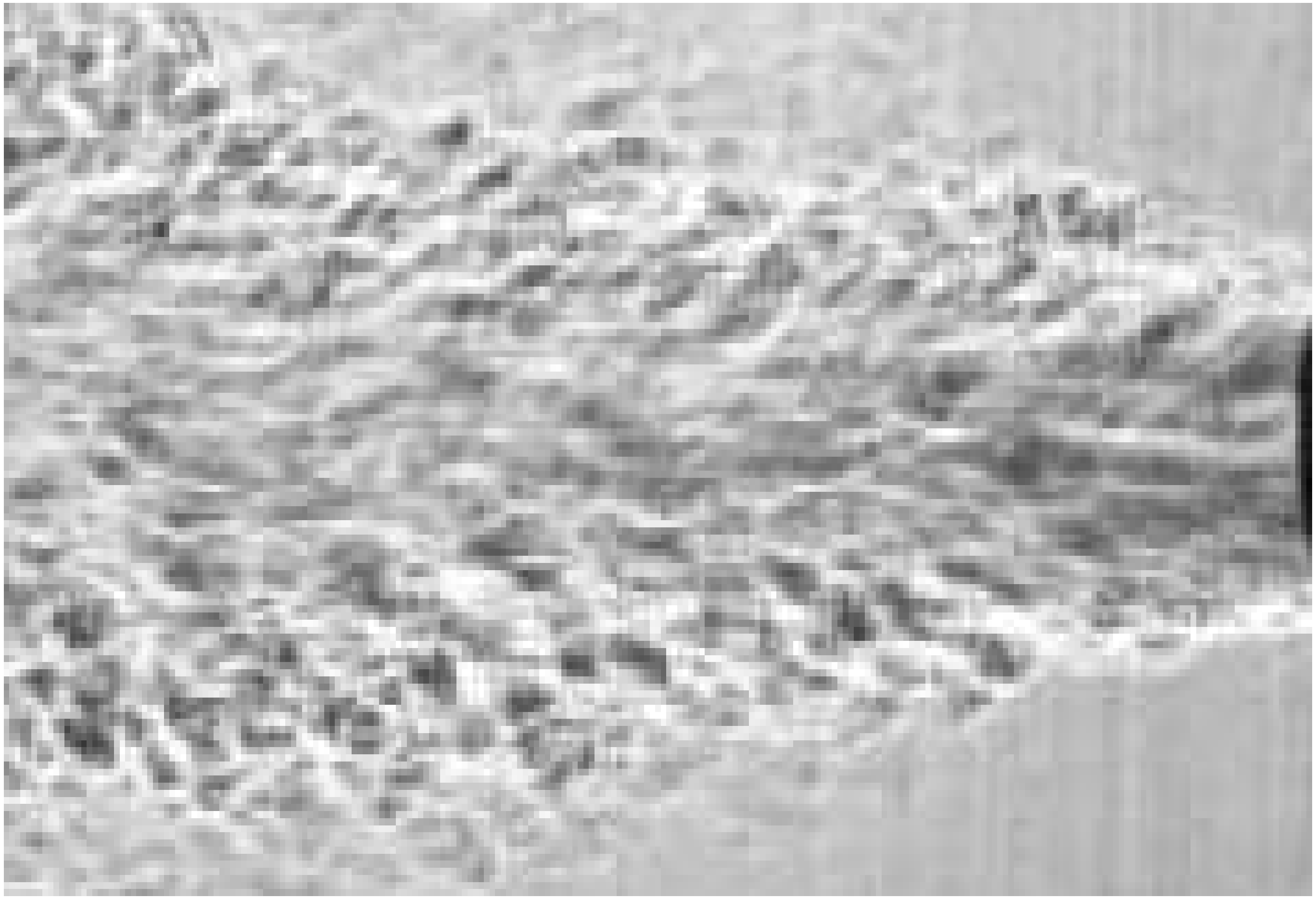
Mathematical Modeling; The Navier-Stokes Equations:

- **Prediction of pressure and velocity field.**
- **Analytical solution of non-linear partial differential equations is not possible, generally.**
- **One can obtain numerical solution of entire field with reasonable accuracy.**

Tackling Turbulence:

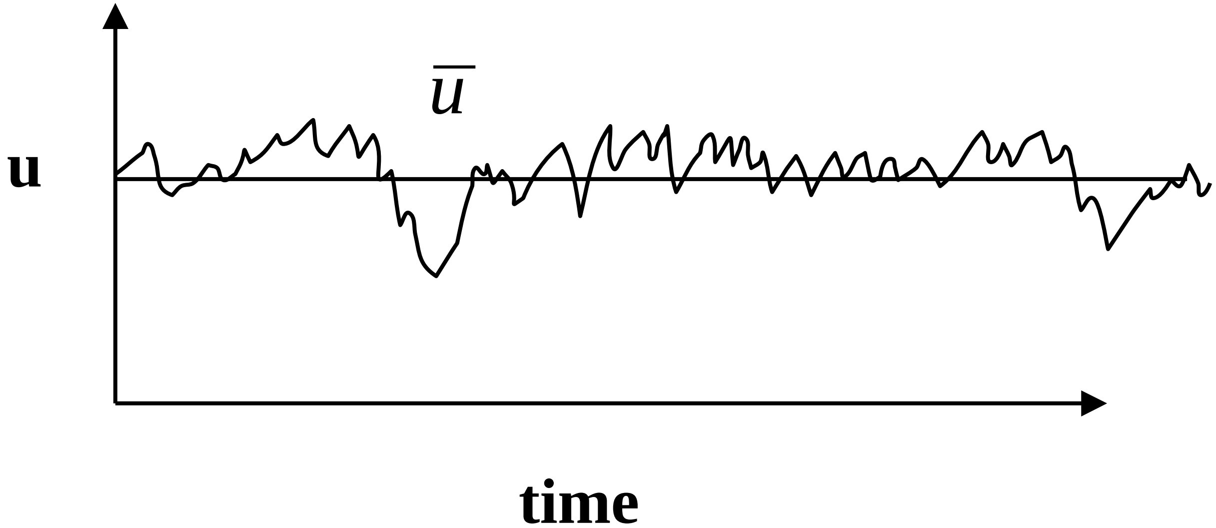
- **Solving laminar flow problem is not a challenge any more.**
- **Turbulence is now the biggest challenge for CFD.**
- **Different perspective of turbulence result in different simulation strategies.**

  1. **RANS (Reynolds Averaged Navier-Stokes)**
  2. **DNS (Direct Numeric Simulation)/LES**



A turbulent jet

RANS:



• **Turbulent velocity is considered to be fluctuating about a mean value.**

• **By taking a time average one can reduce the dynamical complexity.**

DNS/LES:

- **LES and DNS consider turbulent flow as,**

  'Unsteady, three dimensional flow.'

Why LES:

- **High quality predictive numerical solution can enhance the understanding of flow.**
- **With LES inertial flow scales can be resolved.**
- **Code development paved way to do more fundamental work.**
- **Creates new ideas.**
- **Current code is quite flexible and can be modified to tackle boundary layer type flows, free surface flows, mixing layers, bluff bodies and even complex geometries.**

Brief History of LES:

- **RANS type turbulence modeling is almost eighty years old.**
- **First LES type modeling was performed by Deardorff (1970, 1973)**
- **Moin & Kim perform first formal LES in (1982).**

Navier-Stokes Equations:

Continuity Eq:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \qquad \text{———} \quad (1)$$

or,

$$\frac{\partial u_i}{\partial x_i} = 0 \qquad i = 1,2,3$$

Momentum Eq:

$$\frac{\partial u}{\partial t}+\frac{\partial u^2}{\partial x}+\frac{\partial uv}{\partial y}+\frac{\partial uw}{\partial z}=-\frac{\partial P}{\partial x}+\frac{1}{\mathrm{Re}}\left[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right] \qquad (2)$$

$$\frac{\partial v}{\partial t}+\frac{\partial vu}{\partial x}+\frac{\partial v^2}{\partial y}+\frac{\partial vw}{\partial z}=-\frac{\partial P}{\partial y}+\frac{1}{\mathrm{Re}}\left[\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right] \qquad (3)$$

$$\frac{\partial w}{\partial t}+\frac{\partial wu}{\partial x}+\frac{\partial wv}{\partial y}+\frac{\partial w^2}{\partial z}=-\frac{\partial P}{\partial z}+\frac{1}{\mathrm{Re}}\left[\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right] \qquad (4)$$

$$\frac{\partial u_i}{\partial t}+\frac{\partial u_i u_j}{\partial x_j}=-\frac{\partial P}{\partial x_i}+\frac{1}{\mathrm{Re}}\frac{\partial^2 u_i}{\partial x_j \partial x_j} \qquad i, j=1,2,3 \qquad (5)$$

Space Filtering:

- **LES is based on concept of space filtering.**
- **It simulates only the eddies of size larger than some critical length scale.**
- **Small scale eddies are modeled.**

Large and small eddies

Filtering operation for any variable ‘u’ is defined as:

$$\bar{u} = \int_{-\infty}^{\infty} G(x-\xi)\, u(\xi)\, d\xi \quad \text{——} (6)$$ **(Convolution Integral)**

$$\bar{u} = Filtered\ Variable$$

$$G(x-\xi) = Kernel\ of\ transform$$

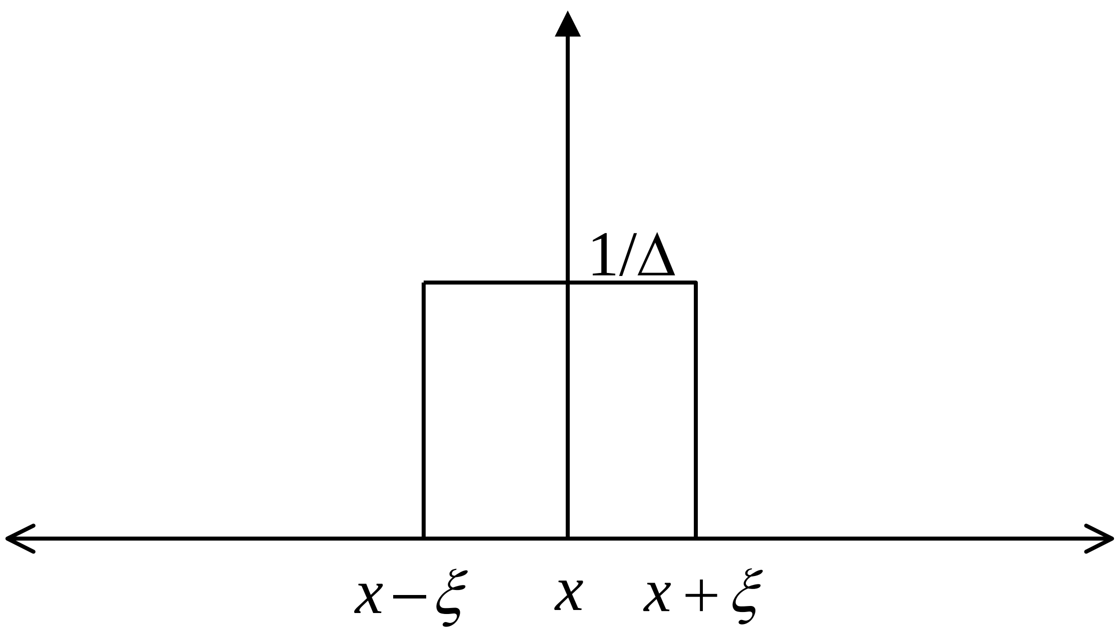
According to above definition filtering operation is weighted averaging.

Subgrid part is given as; $u' = u - \bar{u} \quad \text{——} (7)$

A simplest filter is,

$$G(x-\xi)=\begin{cases}\dfrac{1}{\bar{\Delta}} & \text{if } |x-\xi|\le \bar{\Delta}/2 \\ 0 & \text{otherwise}\end{cases} \qquad \text{———} \qquad (8)$$

Its called 'Top Hat' filter.



$$\overline{u} = \int_{x-\overline{\Delta}/2}^{x+\overline{\Delta}/2} \frac{1}{\overline{\overline{\Delta}}} u(\xi)\, d\xi \qquad \text{———} \quad (9)$$

Here filtered quantity is clearly an average value over filter width Δ at any given point in space.

Filtering of N-S Equations:

Continuity Equation:

$$\int_{-\infty}^{\infty} G(x_i - \xi)\frac{\partial u_i}{\partial x_i} d\xi = 0$$

$$\Rightarrow \frac{\partial}{\partial x_i}\int_{-\infty}^{\infty} G(x_i - \xi) u_i d\xi = 0$$

$$or \quad \frac{\partial \overline{u_i}}{\partial x} = 0 \qquad \text{———————} \qquad (10)$$

Filtered continuity equation is exactly same as unfiltered equation.

Similarly for momentum equation,

$$\frac{\partial \overline{u_i}}{\partial t} + \frac{\partial \overline{u_i u_j}}{\partial x_j} = -\frac{\partial \overline{P}}{\partial x_i} + \frac{1}{\mathrm{Re}} \frac{\partial^2 \overline{u_i}}{\partial x_j \partial x_j} \qquad \text{——— (11)}$$

Filtered momentum equation is not same as original one.

$$\frac{\partial \overline{u_i}}{\partial t} + \frac{\partial \overline{u_i}\,\overline{u_j}}{\partial x_j} = -\frac{\partial \overline{P}}{\partial x_i} + \frac{1}{\mathrm{Re}} \frac{\partial^2 \overline{u_i}}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} \qquad \text{——— (12)}$$

and,

$$\tau_{ij} = \overline{u_i u_j} - \overline{u_i}\,\overline{u_j} \qquad \text{——— (13)}$$

Which is subgrid stress tensor.

Numerical Solution:

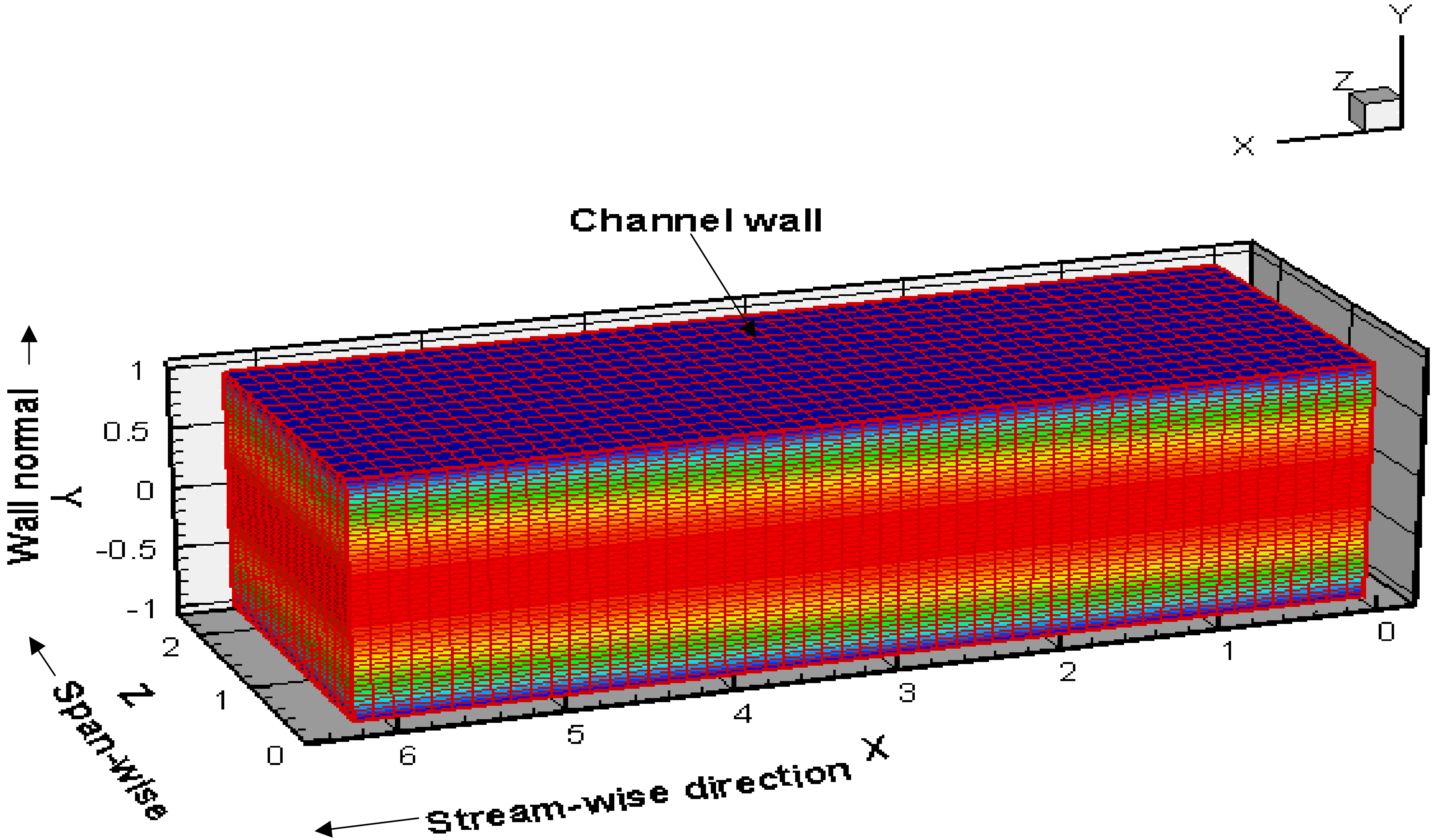
- **It is interesting to note that filtered N-S equations are quite similar to unfiltered N-S eq.**

- **Only difference is the term involving subgrid stress tensor.**

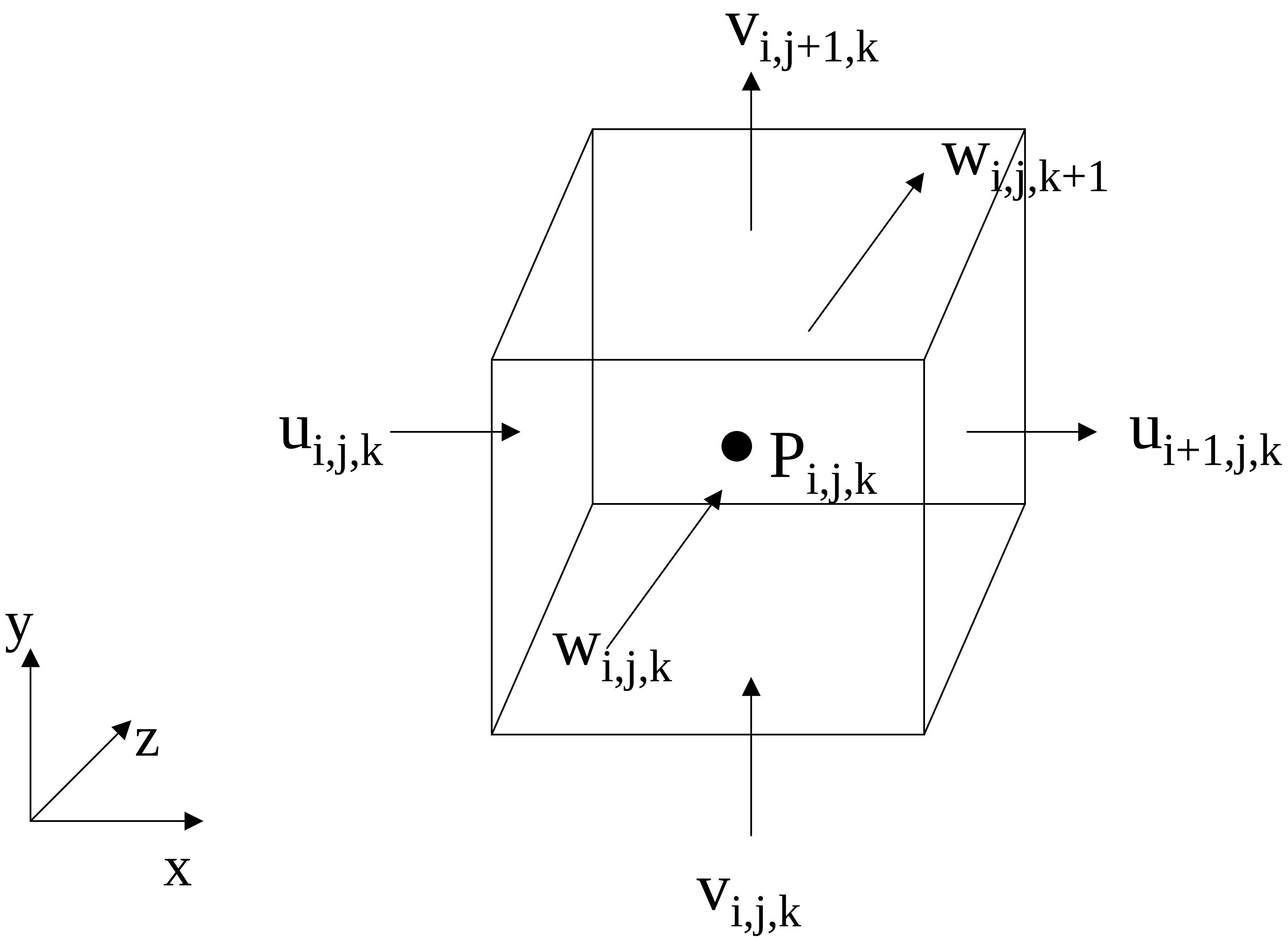
- **In LES objective of numerical solution is to solve unsteady 3-dimensional flow equations.**

- **Current algorithm is called Fractional Step scheme.**

- **N-S equations are being solved on structured staggered grid.**



Geometry and grid for channel



A unit cell in staggered grid.

Algorithm:

- **Take a pressure field and a solenoidal velocity field as initial condition.**
- **Solve x-, y- & z-momentum equations for u, v and w velocity component respectively, for next time step.**
- **For discretization of space derivatives second order finite difference scheme is used.**
- **For time integration non-linear terms are discretized explicitly using Runge-Kutta or Adams-Bashfort schemes, for linear terms implicit Crank-Nicolson scheme is used.**

- **Resulting velocity field is globally divergence free but not locally.**
- **Using a scalar function ϕ, velocity field is projected to divergence free velocity field.**
- **Finally pressure is calculated for next time step.**

Discretized Momentum Equation:

$$\frac{\hat{u}_i^k - u_i^n}{\Delta t} = \gamma^k H_i^n + \rho^k H_i^{n-1} + \frac{\alpha^k}{2\mathrm{Re}} L_{jj}\left(\hat{u}_i^k + u_i^n\right) - \alpha^k G_i P^n \quad (14)$$

$$H_i = \frac{\partial u_i u_j}{\partial x_j}$$ **(Descretized non-linear convection term)**

$$G_i = \frac{\partial}{\partial x_i}$$ **(Discrete gradient operator)**

$$L_{jj} = \frac{\partial}{\partial x_j \partial x_j} \qquad L_{11} = \frac{\partial^2 u}{\partial x^2} = \frac{u_{i-1,j,k} - 2u_{i,j,k} + u_{i+1,j,k}}{\Delta x^2}$$

(Discrete Laplacian operator)

Where γ, ρ and α are parameters for Runge-Kutta or Adams-Bashfort scheme.

If we define; $\Delta u_i = \hat{u}_i^k - u_i^n$

Then eq. (14) will become,

$$\Delta u_i - \beta^k L_{jj} \Delta u_i = \left(\gamma^k H_i^n + \rho^k H_i^{n-1} - \alpha^k G_i P^n \right) + 2\beta^k L_{jj} u_i^n \quad \text{—— (15)}$$

Where,

$$\beta^k = \Delta t \alpha^k \big/ 2\mathrm{Re}$$

Discretized equation is a linear system.

At each node six neighboring nodes are involved in eq.

This linear system results in a large banded matrix for each momentum equation.

$$[A]\{\Delta u_i\} = \{RHS_i\}$$

$$\begin{bmatrix}
a & b & & & & & & f & g & & & & & & \\
c & a & b & & & & & & f & g & & & 0 & & \\
 & c & a & b & & & & & & . & . & & & & \\
 & & c & a & b & & & & & & . & . & & & \\
 & & & . & . & . & & & 0 & & & . & . & & \\
 & & & & . & . & . & & & & & & . & . & \\
 & & & & & . & . & . & & & & & & f & g \\
d & & & & & & . & . & . & & & & & & f \\
e & d & & & 0 & & & . & . & . & & & & & \\
 & . & . & & & & & & . & . & . & & & & \\
 & & . & . & & & & & & . & . & . & & & \\
 & & & . & . & & & & & & . & . & . & & \\
 & & & & . & . & & & & & & . & . & . & \\
 & & 0 & & & e & d & & & & & & c & a & b \\
 & & & & & & e & d & & & & & & c & b
\end{bmatrix}
\begin{Bmatrix}
\Delta u_1 \\ \Delta u_2 \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ \Delta u_n
\end{Bmatrix}
=
\begin{Bmatrix}
RHS_1 \\ RHS_2 \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ RHS_n
\end{Bmatrix}
\quad \text{— (16)}$$

nxn matrix for n nodes.

Solution techniques: Gaussian elimination or LU decomposition.

Required number of operation for 'n' nodes ~$O(n^3)$.

Modest LES requirement n = 64x100x100

= 640,000 nodes.

State of the art $n = 512^3$

~ 130 million nodes.

I am aiming at n = 128x100x150

~ 2 million nodes

Problem: Calculation requirements are prohibitively high.

Solution: Approximate factorization.

$$[A]\{\Delta u_i\} = \{RHS_i\}$$

$$(1 - A_{i1})(1 - A_{i2})(1 - A_{i3})\Delta u_i = RHS_i \qquad (17)$$

or,

$$(1 - A_{i1})\Delta u_i^{**} = RHS_i$$

$$(1 - A_{i2})\Delta u_i^{*} = \Delta u_i^{**} \qquad (18)$$

$$(1 - A_{i3})\Delta u_i = \Delta u_i^{*}$$

In case of structured grid 1-A_{i1}, 1-A_{i2} & 1-A_{i3} are tri-diagonal matrices.

$$\begin{bmatrix} a & b & & & & & & & & \\ c & a & b & & & & & 0 & & \\ & c & a & b & & & & & & \\ & & \cdot & \cdot & \cdot & & & & & \\ & & & \cdot & \cdot & \cdot & & & & \\ & & & & \cdot & \cdot & \cdot & & & \\ & & & & & \cdot & \cdot & \cdot & & \\ & & & & & & c & a & b & \\ & & 0 & & & & & c & a & b \\ & & & & & & & & c & a \end{bmatrix} \quad \text{nxn}$$

Solution technique; TDMA or Thomas algoritm.

Required number of operations for ‘n’ nodes ~ O(3n)

Divergence Free Velocity Field:

Momentum equation gives $\hat{u}_i^k$ **which is non-solenoidal.**

At new time step divergence free velocity u_i^{n+1} **is defined as;**

$$\frac{u_i^{n+1}-\hat{u}_i^k}{\Delta t}=-\alpha^k G_i\phi \qquad (19)$$

If u_i^{n+1} **is divergence free,**

$$D_i u_i^{n+1} = 0$$

D_i = *Discrete divergence operator*

Apply $\mathbf{D_i}$ on (19),

$$D_i G_i \phi = \frac{1}{\alpha^k \Delta t} D_i \hat{u}_i^k$$

$$\Rightarrow \quad L_{ii} \phi = \frac{1}{\alpha^k \Delta t} D_i \hat{u}_i^k \qquad \text{——— (20)}$$

(Poisson eq.)

Solve (20) for ϕ and use (19) to get u_i^{n+1}.

Solution of Poisson Equation:

- **In case of Poisson equation we are looking again at a huge matrix to invert.**
- **In the current situation we have periodic boundary condition in spanwise direction.**
- **By applying Fourier decomposition in that direction we decoupled all x-y planes.**
- **Now we can solve for each Fourier coefficient a set of equation only in one plane, i.e. x-y plane.**
- **For Poisson equation solution in each plane Fishpak is used.**
- **Fishpak requires $n^2 \log_2 n$ opertions for 'n' nodes.**

Mathematical Details:

Poisson equation can be expressed as;

$$L_{ii}\phi = F_i$$

$$\left(L_{11} + L_{22} + L_{33}\right)\phi = F_i \qquad \text{———— (21)}$$

If 3 is periodic, Fourier transform of (21) will give,

$$\left(L_{11} + L_{22} + k_3^{\,2}\right)\hat{\phi} = \hat{F}_i \qquad \text{———— (22)}$$

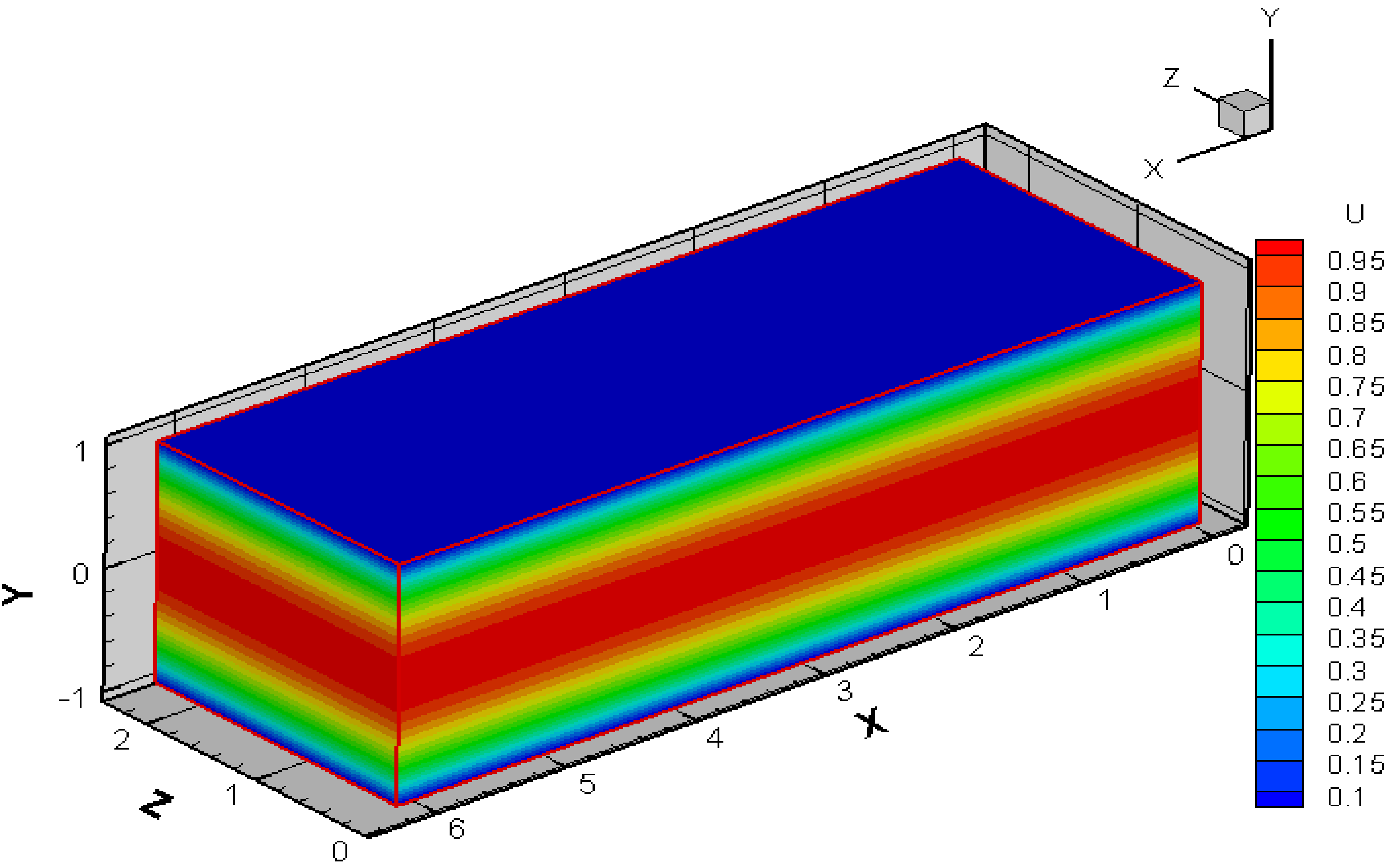
Solve for $\hat{\phi}$ **with Fishpak.**

Inverse Fourier transform will give ϕ**.**

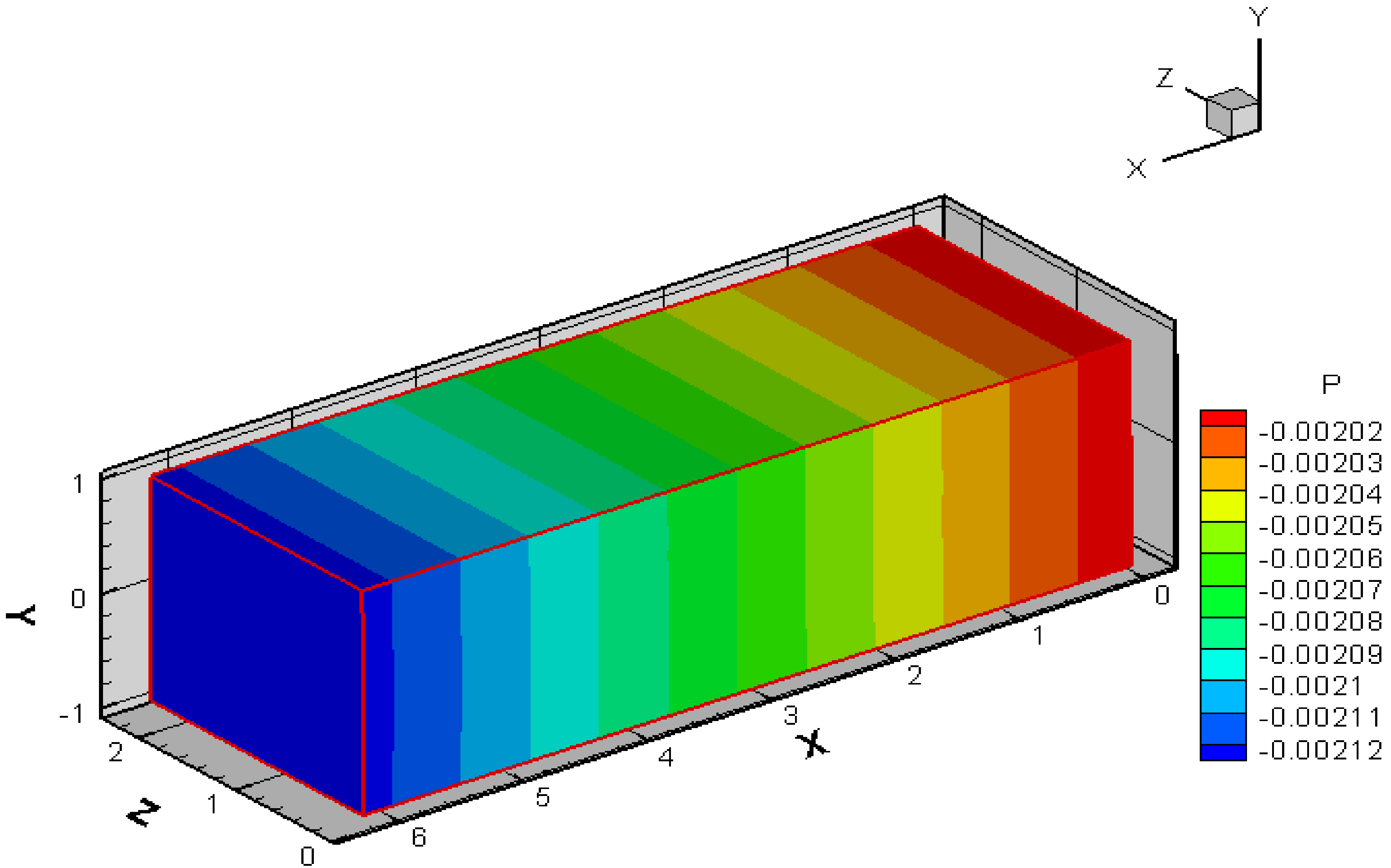
Pressure Calculation:

Once ϕ is known, pressure for next time step is calculated as,

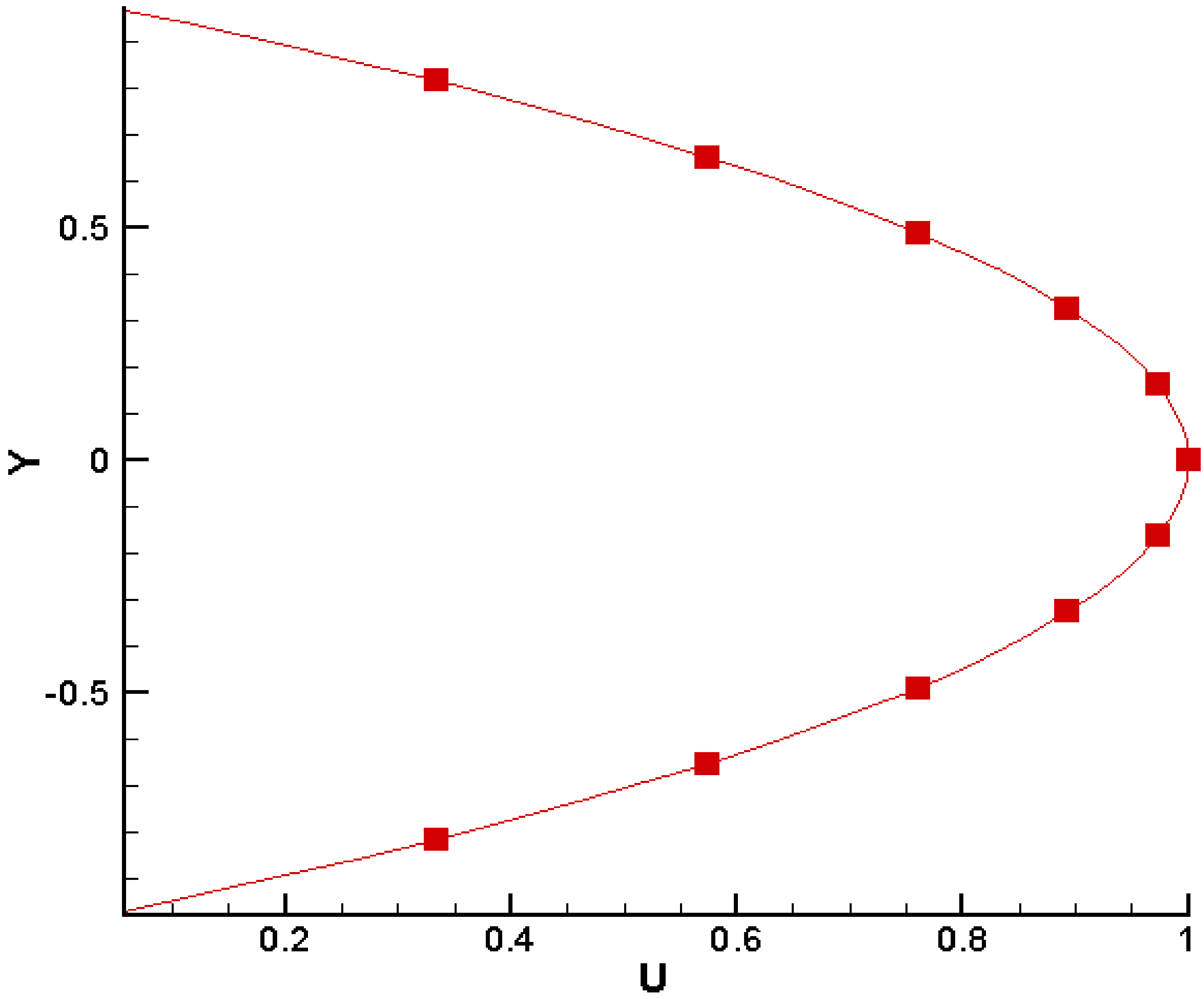
$$P^{n+1} = P^{n} + \phi - \beta L_{jj}\phi \qquad (23)$$



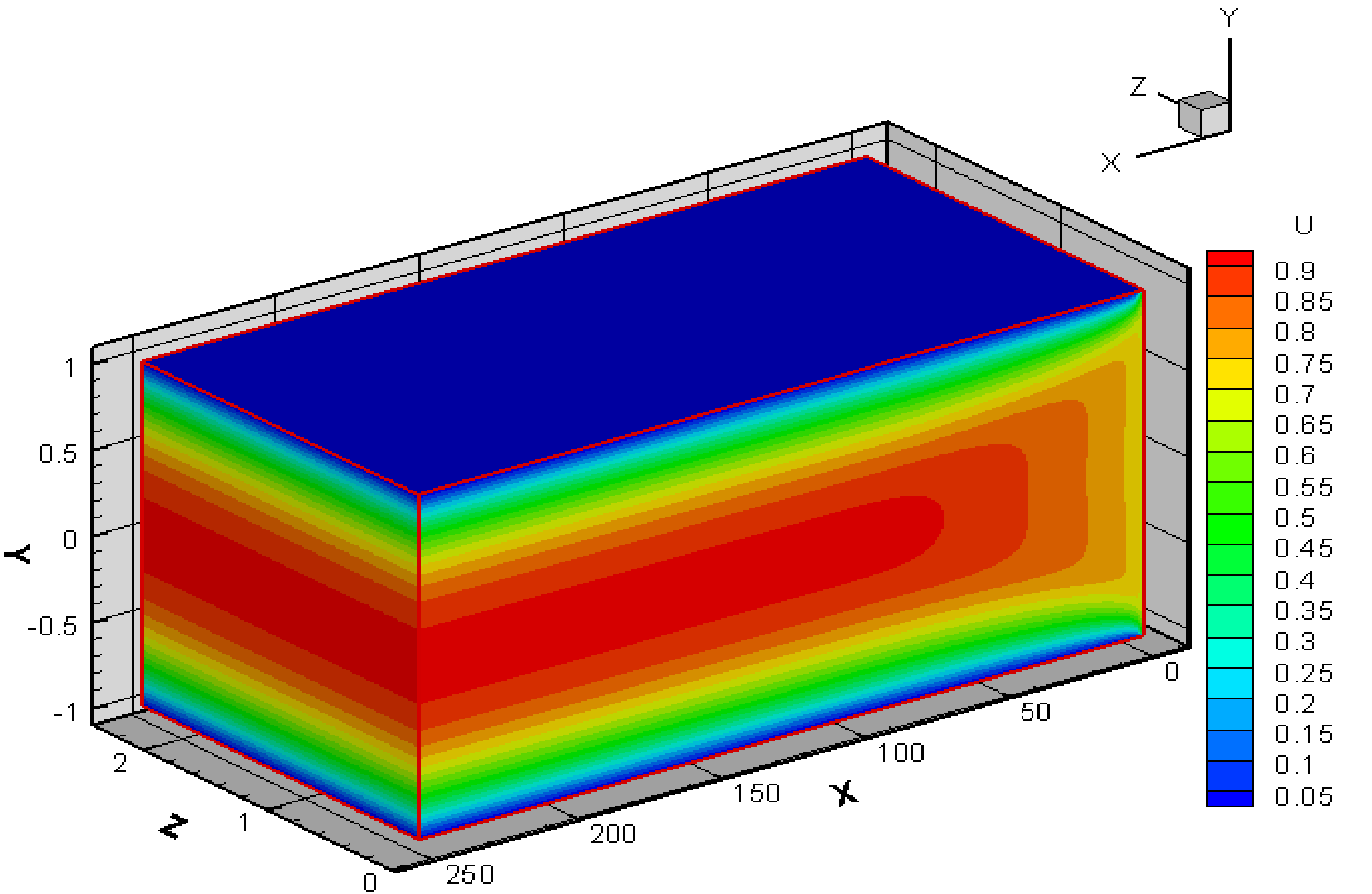
Velocity field for fully developed channel flow.



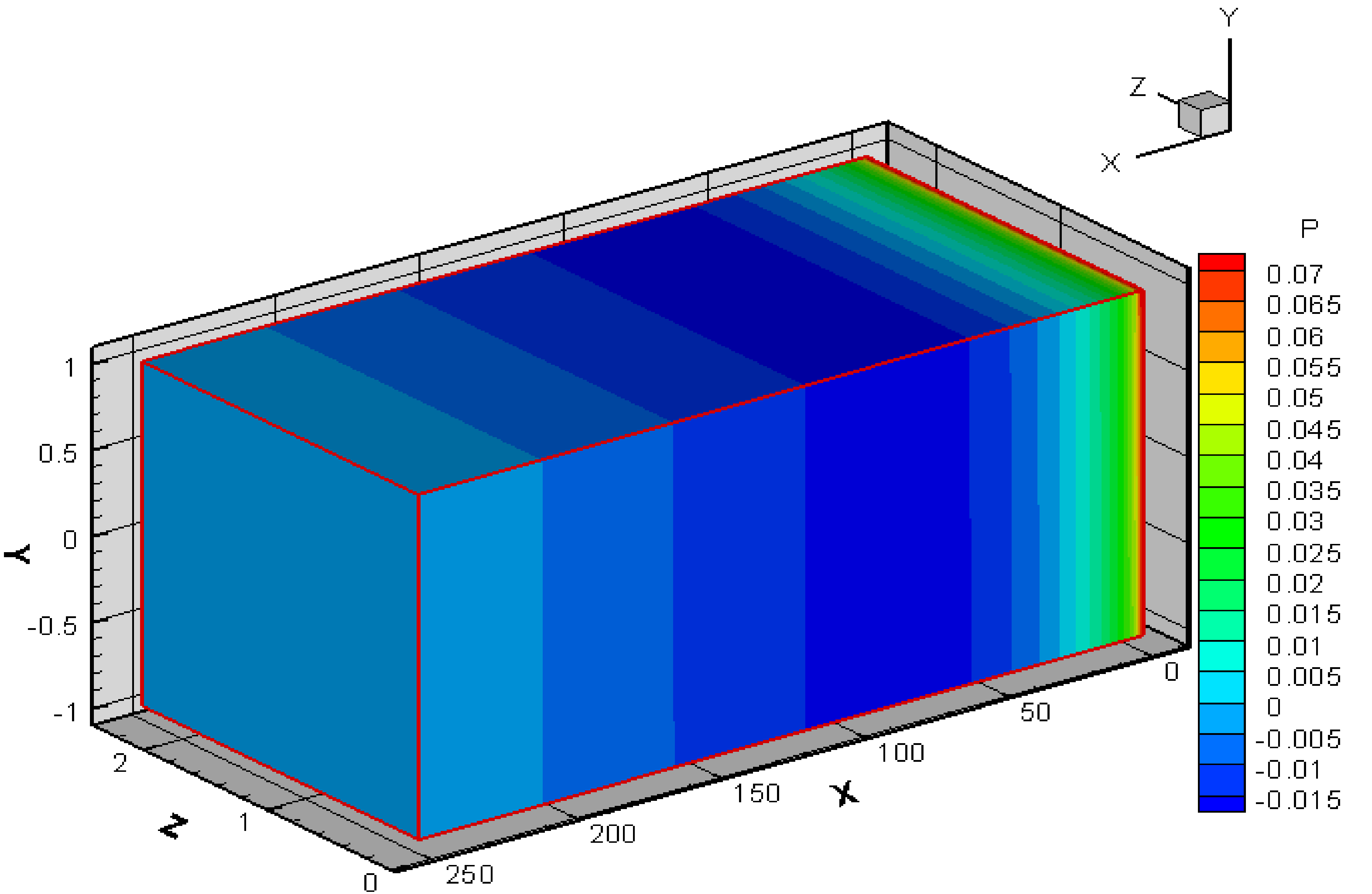
Pressure field for fully developed channel flow.



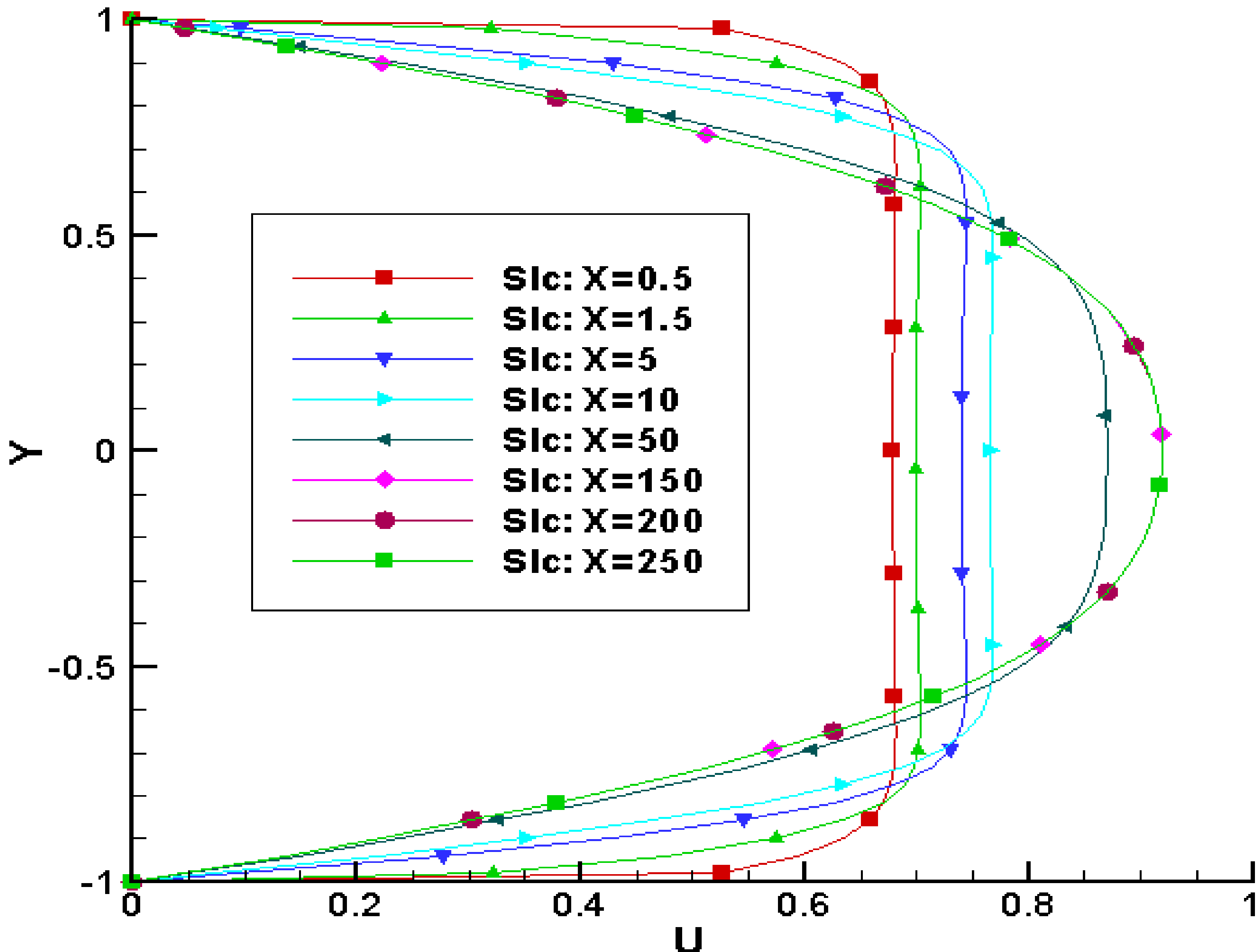
Velocity Profile



Velocity field for channel inlet developing flow.



Pressure field for channel inlet developing flow.



Velocity profiles at different locations along the channel for inlet developing flow.