
Renumbered Iterative Algorithm for Navier-Stokes Simulation

Is Bunyamin SURYO

May - July 2009

Internship of MASTER 1

University of Nice - Sophia Antipolis, France

Advised by Prof. Alain Dervieux

The TROPICS Project team, INRIA Sophia Antipolis, France

MATHMODS Program

Overview

Introduction

Navier-Stokes Simulation

Experiment Description

Results and Discussions

Conclusions

1.Introduction

1.1 Description of INRIA Sophia Antipolis

The French National Institute for Research in Computer Science and Control conducts research in the field of information and communication science and technology.

The scientific research of INRIA are grouped 5 major:

1. Applied Mathematics, Computation and Simulation
2. Algorithmics, Programming, Software and Architecture
3. Networks, Systems and Services, Distributed Computing
4. Perception, Cognition, Interaction
5. Computational Sciences for Biology, Medicine and the Environment

1.Introduction

1.1 Description of INRIA Sophia Antipolis

One of project team in Applied Mathematics, Computation and Simulation theme is

Program Transformations for Scientific Computing (TROPICS)

1. Automatic Differentiation (AD)
2. Computational Fluid Dynamics (CFD) application of AD

Advisor: Prof.Alain DERVIEUX

Co-advisor: Ms.Anca BELME, Dr.Stephen WORNOM

1.Introduction

1.2 The Training Aims

In numerical simulation:

- Central Processing Unit (CPU) cost
- Reducing CPU cost
- The behaviours of the sparse matrix
- The developing algorithm of the numerical approximation

In this training:

Improving the algorithm of mesh numbering for the Navier-Stokes computation in the case of the flow around a circular cylinder.

2. Navier-Stokes Simulation

2.1 Navier-Stokes equations

The Navier-Stokes equations commonly written as

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \nabla \cdot \mathbf{T} + \mathbf{f}$$

where:

\mathbf{v} : the flow velocity

ρ : the fluid density

p : the pressure

\mathbf{T} : stress tensor

\mathbf{f} : body forces

∇ : the del operator.

2. Navier-Stokes Simulation

2.2 Turbulent Flow

What is turbulence?

At Reynolds number (Re_D), e.g. $Re_D \succeq 20000$ for external flow, the velocity and all other flow properties vary in a random and chaotic way.

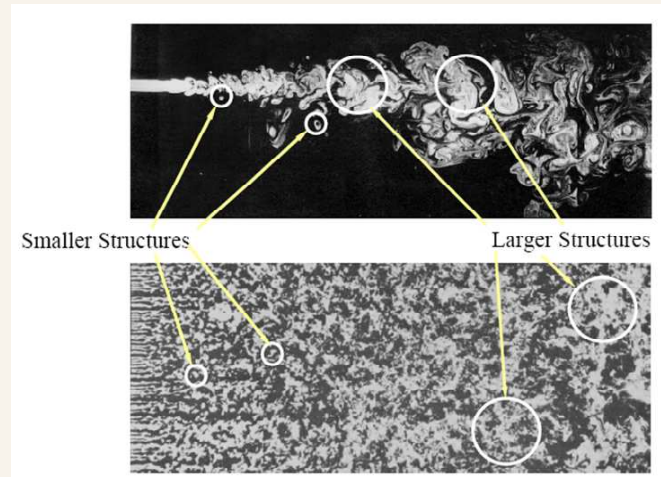


Figure 1: Flow structure of turbulent flow

2. Navier-Stokes Simulation

2.2 Turbulent Flow

How to model the turbulence?

a simple mathematical model that is used to predict the turbulence phenomena.

such as DNS, LES, RANS model

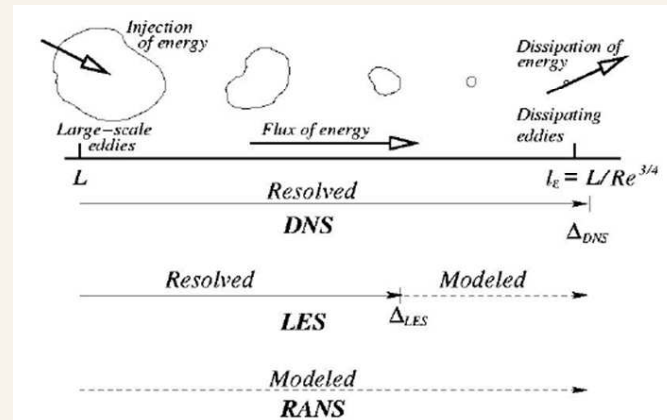


Figure 2: Comparison of DNS, LES, and RANS

2. Navier-Stokes Simulation

2.3 Computational Fluid Dynamics (CFD)

What is CFD?

Using computers and numerical techniques to analyze the problems involving fluid flow.

The main steps in a CFD simulation are:

1. Pre-processing
2. Solving
3. Post-processing

2. Navier-Stokes Simulation

2.3 Computational Fluid Dynamics (CFD)

Physical problems can be represented as PDE.

PDE's can be discretized by using a scheme:

1. Finite Difference
2. Finite Volume
3. Finite Element

which gives us a linear system

2. Navier-Stokes Simulation

2.3 Computational Fluid Dynamics (CFD)

Need a set of computer program/code for discretizing the problem by a such way.

This set of computer program, for instance in FORTRAN, is developed by:

1. Approximate theory,
2. Algorithmic theory.

2. Navier-Stokes Simulation

2.4 Iterative Method for Solving Linear System

Domain discretization and linearization lead to a matrix system problem that is formulated as

$$A.u = f$$

where A is a square matrix.

Solution algorithms for a linear system:

- Direct method: Gauss elimination
- Iterative method: Incomplete Lower Upper (ILU) factorization, Generalized Minimal Residual Method (GMRES)

2. Navier-Stokes Simulation

2.4 Iterative Method for Solving Linear System

In Iterative methods:

⇒ the behaviours/properties of generated matrix are important.

If the generated matrix is a tri-diagonal matrix:

⇒ ILU is an exact solver in one step.

If the generated matrix is a sparse matrix:

1. Treat the sparse matrix by a preconditioner: ILU preconditioner
2. Iterate by a robust iterative method: GMRES

Question:

How to get a generated matrix as 'good' as possible?

2. Navier-Stokes Simulation

2.4 Iterative Method for Solving Linear System

What is preconditioner?

modification of an original linear system.

Let us consider a matrix M that approximates the coeff.matrix A , the transformed system is:

$$M^{-1}Au = M^{-1}f$$

which has the same solution as the original system $A.u = f$

We call matrix M as preconditioner matrix.

One of the simple preconditioner is ILU.

This entails a decomposition of the form $A = LU - R$, where:

L and U are the lower and upper parts of A .

R is the residual matrix

2. Navier-Stokes Simulation

2.5 Best Numbering for ILU

Let us consider two different vertex numberings:

LEXICOGRAPHIC and ORTHOGONAL LEXICOGRAPHIC

⇒ Lexicographic

$$k = i + (j - 1)i_{max}$$

⇒ Orthogonal Lexicographic

$$k = j + (i - 1)j_{max}$$

where $u_k \equiv a_{i,j}$

2. Navier-Stokes Simulation

2.5 Best Numbering for ILU

Domain of vertice numbering schemes

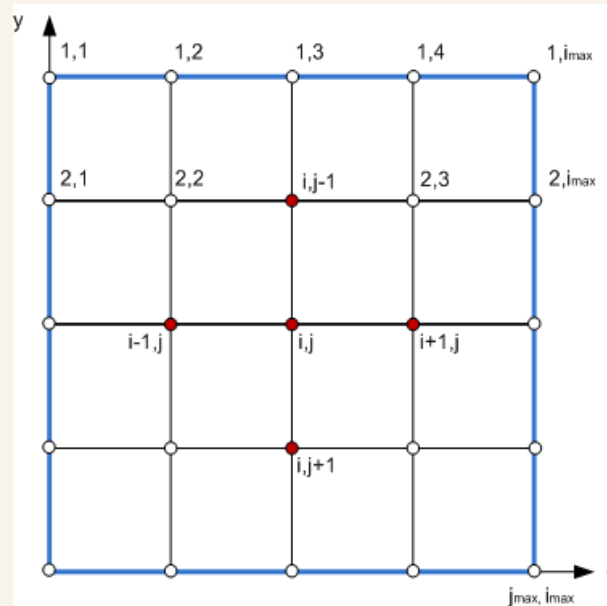


Figure 3: Domain of lexicographic and orthogonal lexicographic

2. Navier-Stokes Simulation

2.5 Best Numbering for ILU

Generated matrix:

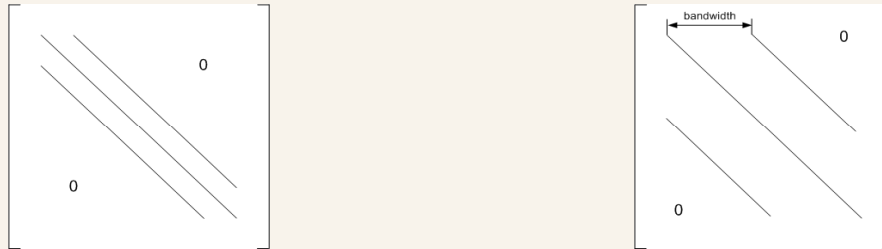


Figure 4: Generated matrix by lexicographic (left) and orthogonal lexicographic(right)

Conclusion:

If $|a_{i-1,j}| \gg |a_{i,j-1}|$

Then choose LEXICOGRAPHIC

3.Experiments Description

3.1 Numerical Code AERO-08

- Parallel computation on 8 processors of the cluster NEF at INRIA Sophia Antipolis
- Using a numerical code AERO-08:
FORTRAN code,
Mix of FE-FV,
Compressible/Incompressible flow,
Turbulence models,
Schwartz decomposition method,etc.

3.Experiments Description

3.2 Compiling and Running AERO-08 Code

Diagram of numerical experiment

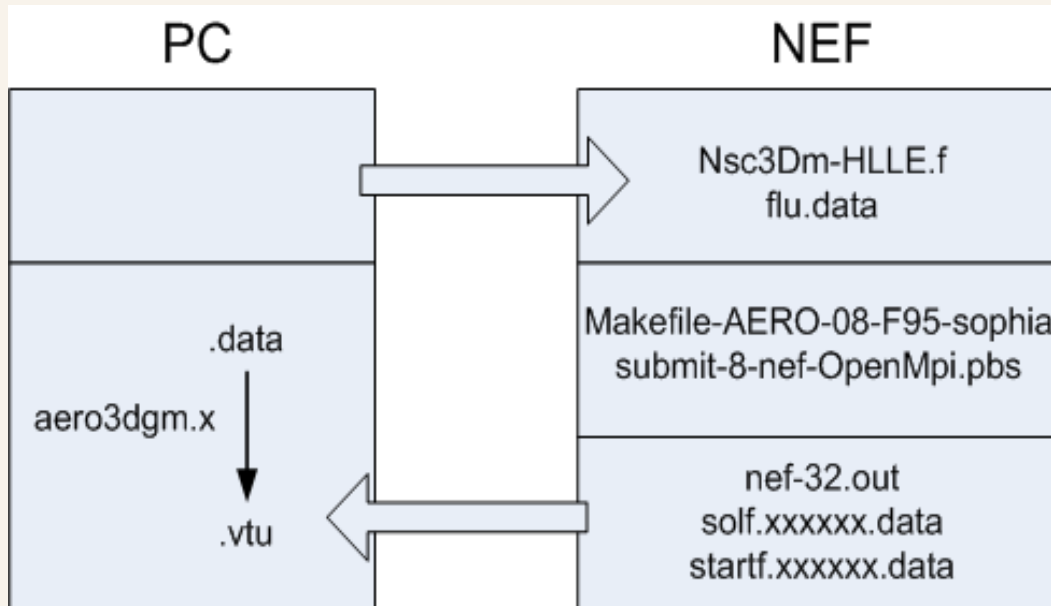


Figure 5: Diagram of numerical experiment

3.Experiments Description

3.3 Renumerotation Method

What is this?

A method of vertices re-numbering for all vertices close to cylinder surface.

How?

Start with a node in the boundary layer

Find the closest neighbor until we reach the given number of layers

All these are performed in the subroutine `RenumILU.f`

3.Experiments Description

3.3 Renumerotation Method

This process has results as

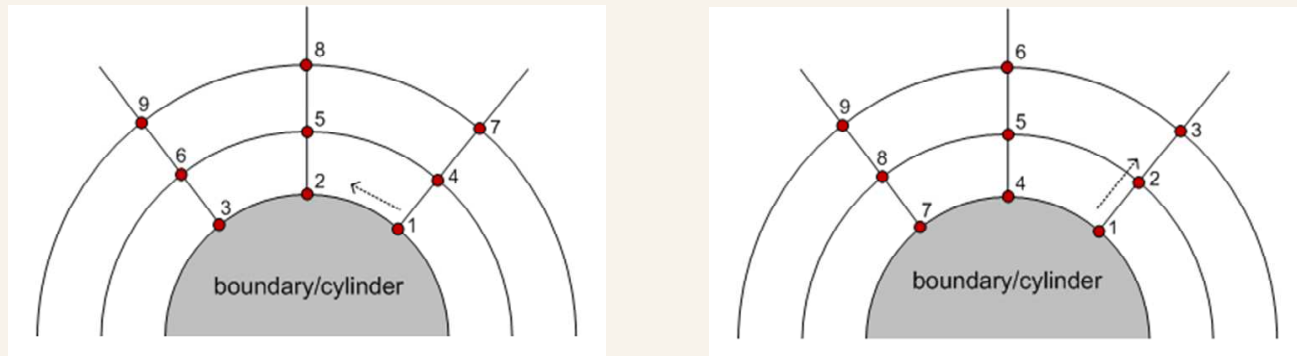


Figure 6: Vertice numbering before (left) and after (right) apply `RenumILU.f`

4.Results and Discussions

4.1 Domain Decomposition

Domain of the computation:

consist of 43282 vertices and 228792 tetrahedrals.

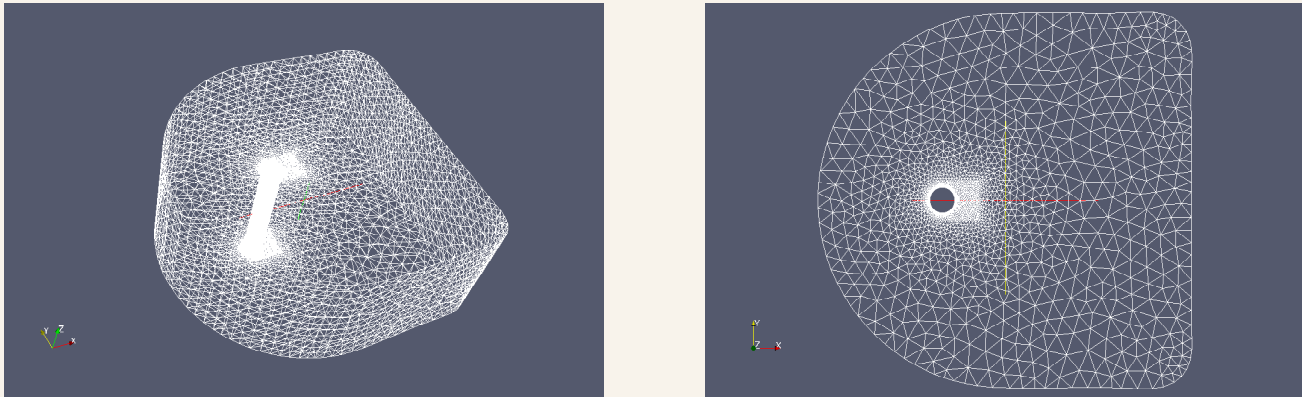


Figure 7: Global domain in 3 dimensional view (left) and Global domain in 2 dimensional view (right)

4. Results and Discussions

4.1 Domain Decomposition

Using parallel computation in 8 processors.

The number of vertices and tetrahedral for each subdomain:

- subdomain 1 : $ns_{max}, nt_{max} = 5827, 28938$
- subdomain 2 : $ns_{max}, nt_{max} = 5581, 28091$
- subdomain 3 : $ns_{max}, nt_{max} = 6127, 29449$
- subdomain 4 : $ns_{max}, nt_{max} = 5713, 27781$
- subdomain 5 : $ns_{max}, nt_{max} = 5744, 27767$
- subdomain 6 : $ns_{max}, nt_{max} = 5853, 28957$
- subdomain 7 : $ns_{max}, nt_{max} = 5931, 28889$
- subdomain 8 : $ns_{max}, nt_{max} = 5802, 28920$

4.Results and Discussions

4.1 Domain Decomposition

8 subdomains of the computation:

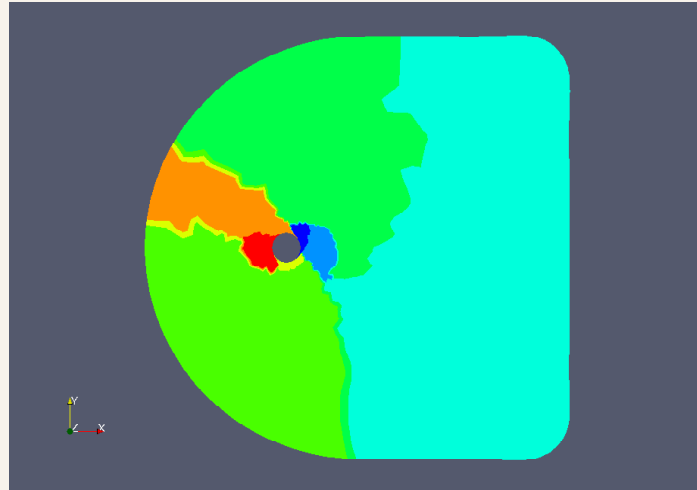


Figure 8: Domain decomposition

4.Results and Discussions

4.2 Effects of Renumerotation

Inside the main routine Nsc3Dm-HLLE.f:

⇒ the subroutine SubMesh.f ⇒ RenSom.f ⇒ RenumILU.f

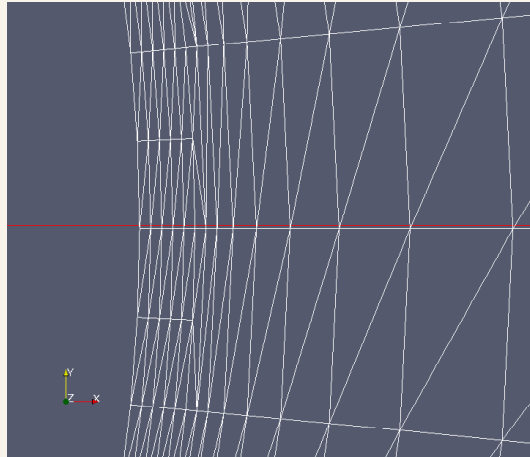


Figure 9: Meshing layers close to surface as the object for applying RenumILU.f

4.Results and Discussions

4.2 Effects of Renumerotation

Comparison before and after applying `RenumILU.f`:

$Re=100$, GMRES iteration= 1

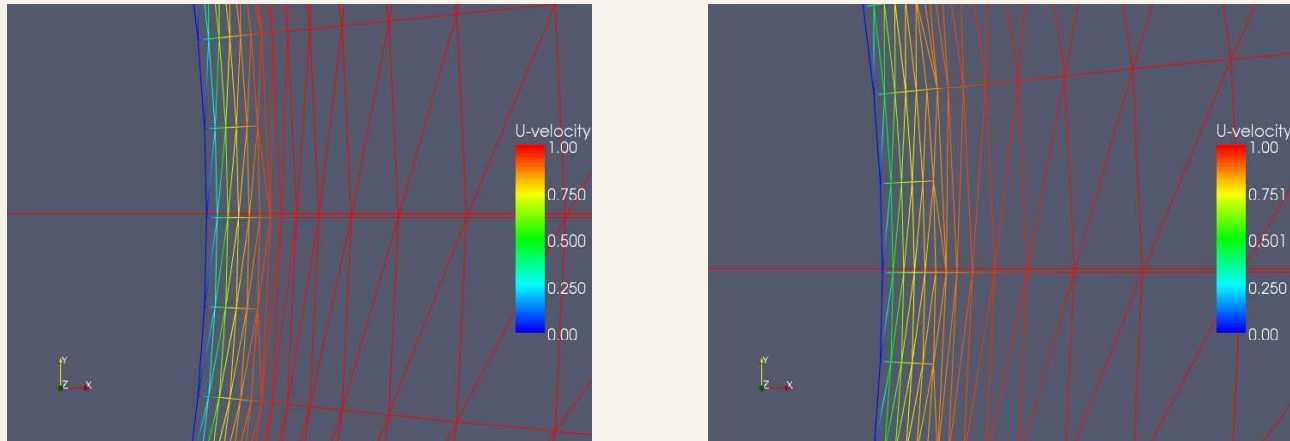


Figure 10: Velocity gradient before (left) and after (right) applying `RenumILU.f`

4.Results and Discussions

4.2 Effects of Renumerotation

Comparison before and after applying `RenumILU.f`:
 $Re=100$, GMRES iteration=1

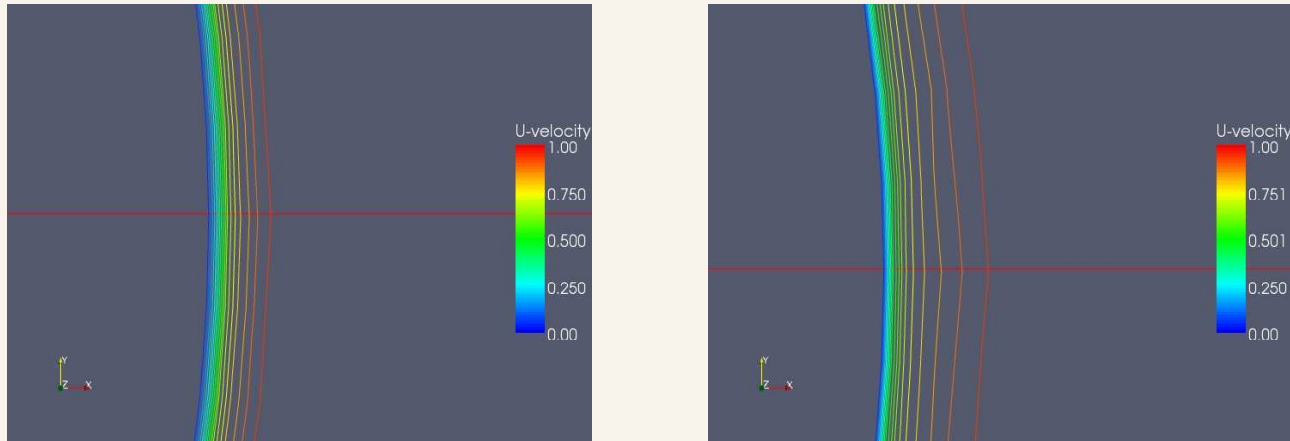


Figure 11: Isovelocity gradient before (left) and after (right) applying `RenumILU.f`

4.Results and Discussions

4.2 Effects of Renumerotation

Comparison before and after applying `RenumILU.f`:
 $Re=100$, GMRES iteration=1

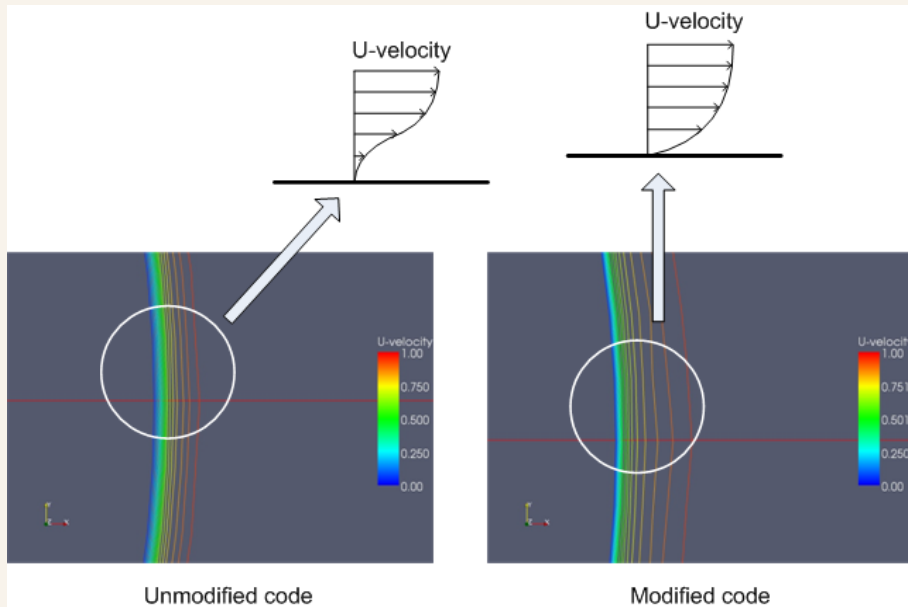


Figure 12: Velocity profile on the surface/boundary

5. Conclusions

1. Numerical approximation leads to generate $A.u = f$
2. The properties of matrix A contribute to the convergence rate of the iterative method
3. The properties of matrix A are contributed by how the numerical approximation developed
4. The new subroutine `RenumILU.f` accelerate the convergence rate of GMRES
5. The acceleration of the convergence rate reduce the computation cost

THANK YOU