

Models in Computational Turbulent Flows

Zeyu Jia* Zhihan Li*

1600010603 1600010653

June 21, 2018

Abstract

Fluids, highly versatile, have both global structures and time-varying details. This results in difficulties for computation and modeling. In spite of such challenges, *computational fluid dynamics* (CFD) has attracted attention since 20th century, and lots of methodologies have been developed due to the huge number of applications of fluids in engineering. Among different kinds of computational fluid dynamics tasks, *turbulent flows* are of high importance because of its universality and complexity. In order to achieve computational possibility, lots of simplified turbulent models have been proposed. In this report, we investigate and summarize three main types of turbulent models: *direct numerical simulation* (DNS) models, *large eddy simulation* (LES) models and *Reynolds-averaged simulation* (RAS) models. The numerical experiments of these models on several cases have been done on our own, together with detailed analysis, and comparison among these experiments presents different properties of models. We show that these three kinds of models enjoy different precision and efficiency, and therefore fit in different cases. Additionally, we present a novel idea about turbulent flow modeling. The project is held on <https://github.com/pppppass/TurbulentFlow>, and Report.pdf is placed in the folder Report as the digital version.

1 Introduction

Fluids are an important subset of phases of matter. The key distinction between fluids and solids are their response to shear forces: shear stress comes in to place in resistance for solids, according to shear modulus; while an absence of such stress results in fluids [4]. In some sense, fluids can be deformed “freely”. Liquids and gases exemplify fluids in every day life.

Equations of fluid motion have been well established, including the continuity equation and the famous Navier–Stokes equations, which reflect the conservation of mass and momentum

*Equal contribution.

respectively [11] [20]. These equations bring opportunities for computational fluid dynamics and fluid simulation. However, because the momentum itself affects flow velocity, the system goes non-linear, and no theorem has been proven about the existence and uniqueness of solution for general cases yet [21].

Because of the freedom of deformation, fluids may have relatively stable global structures in a large scale, while local structures can be space and time-varying. This means fluids enjoy a nature of multi-scale. This leads to difficulties for numerical simulation of fluids, because a grid capturing both coarse and fine structures is bound to have some density.

Turbulent flow, or say turbulence, is a kind of flow phenomenon, on the contrary to laminar flow. Laminar flows, which can be divided into parallel layers, have relatively stable flow velocity. However, in turbulent flow cases, the flows are highly unsteady and the flow velocity changes rapidly. It is said that turbulent flows boost the process of mixing by breaking volumes of different properties into small pieces and enlarging the interaction surface. In this sense, *turbulent diffusion* can be considered, as described in [4]. A prominent property for turbulent flows is chaos: tiny changes of the initial condition may result in completely different patterns. Another property is the change in vorticity, which always lead to different sizes of vortexes and even vortex streets [4].

Because of the chaotic behavior, it is rather complicated to derive properties of general turbulent flows directly from governing equations. Therefore, simplified and statistical models have also been investigated, especially from the communities of physics and probability theory. One of the most famous models is the *energy cascade model* for local isotropic turbulent flows by Richardson [18], which emphasizes the similarity between turbulent flows of different scales. In this model, turbulent flows of large scales contain much energy, and they get broken up and reach relatively equilibrium, and finally enter the dissipation range of scales. Kolmogorov further established concrete hypotheses for these models, and proposed the famous $-5/3$ spectra, which claims the existence of curve segment of slope $-5/3$ in normalized energy spectra [10]. This phenomenon is checked by many experiments, and universality has been verified [17].

On the computational side, there are also simplifications on models designed specially for numerical simulation. There are mainly three categories for corresponding models and algorithms:

- (1) Direct numerical simulation (DNS): directly simulate the turbulent flow according to Navier–Stokes equations, with the fluid model completely unchanged.
- (2) Large eddy simulation (LES): solve only the large-scale part of motion while approximating, or say modeling the small ones.
- (3) Reynolds-averaged simulation (RAS): make advantage of statistical steadiness of the chaotic system and solve the time-averaged equations.

Each model have pros and cons of itself, and these will be discussed in details later.

The plan of this report is as follows. We will introduce basic concepts and equations of fluid dynamics in Section 1 and 2. Different models, say DNS, LES and RAS will be detailedly discussed in Section 3, 4 and 5 respectively. We will conduct a full comparison on these three models in 6, with analysis and experiments provided. Conclusion will be given in Section 7, with acknowledgement Section 8.

We summary our work in this report as follows:

- (1) We perform a literature research about computational fluid dynamics and turbulent flow models. The main contents are digested from [14] [21].
- (2) We analyzed the distinction among three turbulent models, together with numerical experiments given for support. This is done by ourselves.
- (3) We present a novel idea about turbulent flow modeling, which combines recent data-driven methods and tackles some of the disadvantage of present models.

The novelty and originality of this report mainly lie on the analysis, experiments part and the novel idea part.

2 Equations of fluid motion

2.1 Incompressible Navier–Stokes equations

We first establish equations of fluid motion [14] [4], as a preparation of further discussion.

Denote density of the fluid as a scalar ρ , a function of time and position $(t, \mathbf{x}) \in \mathbb{R}^{3+1}$. We also denote the flow velocity as a vector function \mathbf{U} .

We proceed to consider the Lagrangian coordinate first. For a given point \mathbf{x} , we can solve for its trajectory $\mathbf{X}(t, \mathbf{x})$ according to the velocity field from the equation

$$\frac{\partial}{\partial t} \mathbf{X}(t, \mathbf{x}) = \mathbf{U}(t, \mathbf{X}(t, \mathbf{x})). \quad (1)$$

Here \mathbf{X} is considered to be *Lagrangian coordinate*, in accordance to *Eulerian coordinate* \mathbf{x} . Lagrange coordinate provides us a map

$$\Phi^t(\mathbf{x}) = \mathbf{X}(t, \mathbf{x}), \quad (2)$$

which characterizes the “deformation” from time 0 to t . When the fluid is completely steady, say \mathbf{U} is independent of time, this notation coincides with the one in dynamic systems.

For *control volume* (CV) Ω^0 , a simply connected compact region with piece-wise smooth boundary literally, we may consider the corresponding *control mass* (CM), which is the region evolving as \mathbf{U} . Mathematically, the control mass is

$$\Omega^t = \Phi^t(\Omega^0). \quad (3)$$

For an intensive property ϕ , a scalar or a vector, we have

$$\frac{d}{dt} \int_{\Omega^t} \rho \phi d\Omega = \frac{d}{dt} \int_{\Omega^0} \rho \phi d\Omega + \int_{\partial\Omega^0} \rho \phi \mathbf{U} \cdot d\mathbf{S}. \quad (4)$$

If ϕ is itself conserved, we have the integral conservation law

$$\frac{d}{dt} \int_{\Omega^0} \rho \phi d\Omega + \int_{\partial\Omega^0} \rho \phi \mathbf{U} \cdot d\mathbf{S} = \Gamma, \quad (5)$$

where Γ represents the influence of external factors. For the differential one, we introduce *material derivative*

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{U} \cdot \nabla, \quad (6)$$

and the conservation law turns out to be

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \phi \mathbf{U}) = \frac{D \rho \phi}{Dt} + \rho \phi \nabla \cdot \mathbf{U} = \gamma, \quad (7)$$

where γ mirrors the change of ϕ in infinitesimal control mass.

We begin to derive basic equations of fluid motion now. The basic one is the *continuity equation*, which pays attention on mass conservation. If we set $\phi = 1$, we obtain

$$\frac{d}{dt} \int_{\Omega^0} \rho d\Omega + \int_{\partial\Omega^0} \rho \mathbf{U} \cdot d\mathbf{S} = 0, \quad (8)$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0. \quad (9)$$

In the following discourse, we focus on *incompressible fluids* or *solenoidal fluids*, which means the divergence of flow velocity is always zero. There are also some other characterization of the incompressibility, as the following theorem tells. Under mild pressure, liquids like water can be considered to be incompressible, and accounts for popularity of incompressible models.

Theorem 1 For a fluid satisfying continuity equation, the following are equivalent:

- (1) The flow velocity field is divergence-free: $\nabla \cdot \mathbf{U} = 0$;
- (2) The density ρ is constant with respect to the trajectory: $D\rho/Dt = 0$;
- (3) The flow conserves volume: the Jacobian $\det \nabla \Phi^t = 1$.

The continuity equation can be seen as an extension of incompressible condition, taking compression into consideration. In this case, the conservation law boils down to be

$$\rho \frac{D\phi}{Dt} = \gamma. \quad (10)$$

The conservation law of momentum is more subtle then. On Newtonian fluid profile, γ consists of both body forces (e.g. gravity) and surface forces (stress). The equations for

momentum conservation, surface stress, gravity and gravitational potential are

$$\rho \frac{DU_j}{Dt} = \frac{\partial \tau_{ij}}{\partial x_i} - \rho g_i, \quad (11)$$

$$\tau_{ij} = -P\delta_{ij} + \mu \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right), \quad (12)$$

$$\mathbf{g} = -\nabla\psi, \quad (13)$$

$$\psi = gz \quad (14)$$

respectively, where μ is the coefficient of viscosity and P is the pressure. Therefore, using Einstein notation, if we take gravity into consideration, we have

$$\rho \frac{dU_j}{dt} = -\frac{\partial P}{\partial x_j} - \rho \frac{\partial \psi}{\partial x_j} + \mu \frac{\partial^2 U_j}{\partial x_i \partial x_i}. \quad (15)$$

By defining the *modified pressure* $p = P - \rho\psi$ and *kinematic viscosity*, we obtain

$$\frac{D\mathbf{U}}{Dt} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{U}, \quad (16)$$

where the gravity term is resolved.

In all, computational methods for incompressible Newtonian fluids aim to solve

$$\begin{aligned} \frac{D\mathbf{U}}{Dt} &= -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{U}, \\ \nabla \cdot \mathbf{U} &= 0. \end{aligned} \quad (17)$$

This is the *incompressible Navier–Stokes equations* (INS).

Additionally, we introduce the notation of *vorticity*, which is defined as

$$\boldsymbol{\omega} = \nabla \times \mathbf{U}. \quad (18)$$

It is clear from definition that vorticity characterizes the rate of rotation in the fluids. The conservation law can also be derived for vorticity as

$$\frac{D\boldsymbol{\omega}}{Dt} = \nu \nabla^2 \boldsymbol{\omega} + \boldsymbol{\omega} \cdot \nabla \mathbf{U}, \quad (19)$$

which is also a corollary of incompressible Navier–Stokes equations. In turbulent flows, a great part of energy is stored in vorticity, and therefore observing distribution of vorticity is of high importance for computational turbulent flows.

Note that boundary conditions of partial differential equations matter a lot. However, it should be emphasized that the boundary condition of fluid simulation varies according to

physical settings. For example, for walls, the normal velocity \mathbf{V} should be set to zero on the boundary, and p should have zero derivative on the normal direction. For flow patches, the velocity \mathbf{U} should be fixed.

2.2 Nondimensionalization and Reynolds number

It is clear that the equation (17) owns a great variety of invariance, including time and space invariance, orthogonality invariance and affinity invariance. However, it is not obvious that some similarity invariance is held intrinsically in the incompressible Navier–Stokes equations.

We perform nondimensionalization first, by selecting characteristic length scale L_0 and velocity length scale V_0 . By setting $\widehat{\mathbf{x}} = \mathbf{x}/L_0$, $\widehat{t} = tV_0/L_0$, we define

$$\widehat{\mathbf{U}}(\widehat{\mathbf{x}}, \widehat{t}) = \mathbf{U}(\mathbf{x}, t)/V_0, \quad (20)$$

$$\widehat{p}(\widehat{\mathbf{x}}, \widehat{t}) = p(\mathbf{x}, t)/(\rho V_0^2). \quad (21)$$

By this mean, (17) can be changed into

$$\begin{aligned} \frac{D\widehat{\mathbf{U}}}{Dt} &= -\frac{1}{Re}\nabla p + \nabla^2 \widehat{U}, \\ \nabla \cdot \widehat{\mathbf{U}} &= 0. \end{aligned} \quad (22)$$

with the Reynolds number

$$Re = \frac{V_0 L_0}{\nu} = \frac{\rho V_0 L_0}{\mu}. \quad (23)$$

Note that the only constant here is the Reynolds number, and therefore it is the only dependent factor of fluid dynamics. This indicates a Reynolds number similarity can be established as well.

As an result, Reynolds number plays a dominant role in turbulent flows according to experiments. It has been verified that Reynolds number greater than 4000 leads to turbulent flows in some cases, while low reynolds number means laminar flows [8]. In engineering, Reynolds number greater than about 10^4 is of interest, and therefore turbulent flow problems must be considered.

3 Models of direct numerical simulation

The details of this model are referred to books [5] [4].

3.1 Motivation

The direct numerical simulation (DNS) model is one of the most simplest models in simulating turbulent flows. To simulate what happens in turbulent flows, we solve the incompressible Navier–Stokes equations directly (17) (22). This model is the first model developed to simulate

turbulent flows, and is also the most accurate model among all the models. However, since directly solving the incompressible Navier–Stokes equation costs too much time, algorithms of this model is rarely used in practice. We will introduce some basic numerical models and algorithms toward this turbulent flow model.

3.2 Descretization

There are mainly three descretization techniques toward partial differential equations, say the finite difference method (FD, FDM), the finite volume method (FVM) and the finite element method (FEM).

To apply the finite difference method, what we need do first is to discretize the geometric domain into numerical grids. Simple grids parallel to coordinate axes usually apply, but curved grids may also be used for complex geometries. After discretizing the geometric domain, we can utilize finite differences on the grid to represent the derivatives. An example on the real line \mathbb{R} is

$$\frac{\partial u}{\partial x} \Big|_{x_i} = \frac{u(x_{i+1}) - u(x_i)}{x_{i+1} - x_i} + O(x_{i+1} - x_i), \quad (24)$$

where x_{i+1} and x_i are two nodes, and u is the unknown to be solved. In this way, we can transform the partial differential equation into a difference equation, and then approximate the solution by solving systems. Since the descretized functions space on a compact set is often of finite dimension, the finite difference equation can be solved by solving a system of finite variables. This is the idea of finite difference method.

The finite volume method is another category of numerical methods to solve partial differential equations. Similar to the finite difference method, in this method, the domain is been divided into several control volumes at first, and what we are going to define is the function value on the boundary or at the center of each control volumes.

With integral conservation laws, we can find the integral equations for each control volume. If we sum up the integral equations with respect to all control volumes, we can obtain a global conservation. Hence the main problem in the finite volume method is to approximate surface integrals and volume integrals on each control volume. Numerical quadratures and approximations are used here, such as the Simpson’s rule.

The finite element method is a little bit different to the previous two. The main idea comes from Galerkin. That is, instead of solving the equation directly, we write its variational form, and then try to find its weak solution. The domain is also divided into some regions, and the discretized object function space U_h and test function space V_h are composed of piece-wise linear functions (in many cases). Then what we need to do is to find an approximate solution u_h in U_h satisfying the discretized variational problem:

$$\mathcal{L}(u_h, v_h) = 0 \quad (25)$$

for all $v_h \in V_h$. Noticing that this problem is a system of finite variables, it is also tractable to solve.

Every time we discretize quantities, errors get involved. Hence to choose a scheme of both higher order and solving efficiency is our goal.

3.3 Schemes

Schemes need to be carefully designed to numerically solve incompressible Navier–Stokes equations. One scheme of high popularity is

$$\begin{aligned} \frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} + (\mathbf{U}^n \cdot \nabla) \mathbf{U}^n &= -\frac{1}{\rho} \nabla p^{n+1} + \nu \nabla^2 \mathbf{U}^{n+1}, \\ \nabla \cdot \mathbf{U}^{n+1} &= 0. \end{aligned} \quad (26)$$

This scheme can be considered as backward Euler scheme regardless of the convection term. In this scheme, as for space discretization, we can just choose central space schemes.

This Crank-Nicholson scheme is a more accurate scheme than the previous one. The whole scheme can be described as follows:

$$\begin{aligned} \frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} + \left(\mathbf{U}^{n+1/2} \cdot \nabla \right) \mathbf{U}^{n+1/2} &= -\frac{1}{\rho} \nabla p^{n+1} + \frac{\nu}{2} \nabla^2 (\mathbf{U}^{n+1} + \mathbf{U}^n), \\ \nabla \cdot \mathbf{U}^{n+1} &= 0. \end{aligned} \quad (27)$$

Note that viscosity term is modified. Comparing to the previous scheme, the Crank-Nicholson scheme is half-explicit-half-implicit. This can increase the discretization order in time dimension by one. Besides, there are ways to tackle the non-linear term here. The $n + 1/2$ step in the non-linear term can correspond to

$$\left(\mathbf{U}^{n+1/2} \cdot \nabla \right) \mathbf{U}^{n+1/2} = \frac{1}{2} \left(\left(\mathbf{U}^{n+1} \cdot \nabla \right) \mathbf{U}^{n+1} + (\mathbf{U}^n \cdot \nabla) \mathbf{U}^n \right). \quad (28)$$

Another choice is

$$\left(\mathbf{U}^{n+1/2} \cdot \nabla \right) \mathbf{U}^{n+1/2} = \left(\mathbf{U}^{n+1} \cdot \nabla \right) \mathbf{U}^n + (\mathbf{U}^n \cdot \nabla) \mathbf{U}^{n+1} - (\mathbf{U}^n \cdot \nabla) \mathbf{U}^n, \quad (29)$$

which is typical for such coupled problems.

For the partial differential equations which describe the motion of fluid, there is one main difficulty. That is, the equations we solve involve in non-linear terms, and this leads to non-linearity in discretized equations. Hence methods of solving linear equations cannot apply directly, and we will need some other techniques to deal with the non-linearity. There are two ways to solve these non-linear equations: solving simultaneously or solving sequentially.

Newton method is an example of the former way. The core idea is to use tangents to

approximate the function. However, although Newton method enjoys quadratic convergence, it is somehow unstable: the initial guess should always be placed near the real solution. In some extreme cases, convexity also makes a difference. In practice, this method is often combined with other methods, to compensate the weakness of it.

However, methods solving simultaneously are relatively too expensive to simulate fluids. Compared, the sequential solving process is the mainstream. The main idea is to fix some variables to obtain a system which is tractable to solve, and then update these variables in order to start another iteration. This resembles Picard iterations for ordinary differential equations. As in the sequential solving process, we always solve these linear equations with iterative methods. Hence the whole process contains two parts:

(1) Fix all the variables involved in coefficients to get a system of linear equations, and then solve these linear equations. Next, update coefficient variables. Such iterations are called *outer iterations*.

(2) During the previous step, solve the linear equations with some plugged-in variables iteratively. Such iterations are called *inner iterations*.

In order to keep the stability of the outer iteration, sometimes under-relaxation trick is used here.

3.4 Linear system solvers

No matter what method we use to discretize the differential equations, we have to solve linear systems eventually. The following list consists of the most commonly used methods:

- (1) LU decomposition or Gaussian elimination;
- (2) Jacobian method;
- (3) Gauss–Seidel (GS) method;
- (4) Successive over-relaxation (SOR);
- (5) Conjugate gradient (CG) method;
- (6) Incomplete LU decomposition and preconditioned conjugate gradient method (ILU-PCG, Stone method);
- (7) Multi-grid method (MGM).

After discretizing the equations, normally we will obtain a system, and then the previous described methods can be deployed to solve. Some methods, such as the multi-grid method, enjoys a linear time complexity to the scale of the problem, and is therefore commonly used.

4 Models of large eddy simulation

We then introduce basic knowledge about the large eddy simulation model [14] [4].

4.1 Motivation

Even though direct numerical simulation is of great accuracy, the time cost in simulating with that model is extremely high. Engaged with the idea that small motion of the fluid make small contributions in the entire simulation, the large eddy simulation model, which only considers large scale motion, has been proposed.

4.2 Basic model

In order to simply consider the large scale motion, this model first erases all the details of the fluid. That is, any quantity ϕ can be convolve with a filter kernel $K(x, x')$ like

$$\bar{\phi}(\mathbf{x}) = \int K(\mathbf{x}, \mathbf{x}') \phi(\mathbf{x}') d\mathbf{x}'. \quad (30)$$

Here the kernel G should have smoothing property. A local average filter, a band-width filter or a Gaussian kernel may apply. In this equation, there is also a parameter Δ which matters. This parameter describes the scale of the filter, which means that eddies which are larger than Δ will be calculated and the ones smaller than Δ will be ignored or approximated. Obviously Δ should be no larger than the grid size, but usually no other restriction is forced.

Equipped with this, we convolve the incompressible Navier–Stokes equations with the filter kernel to obtain

$$\frac{\partial(\bar{U}_i)}{\partial t} + \frac{\partial(\bar{U}_i \bar{U}_j)}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial}{\partial x_j} \left[\left(\frac{\partial \bar{U}_i}{\partial x_j} + \frac{\partial \bar{U}_j}{\partial x_i} \right) \right]. \quad (31)$$

According to the linearity of the continuity equation and incompressibility, we also have

$$\frac{\partial(\bar{U}_i)}{\partial x_i} = 0. \quad (32)$$

Since the second term in the left of (31) is hard to computed, we introduced the *subgrid-scale Reynolds stress* (SGS Reynolds stress) in (33) to handle this. The subgrid-scale Reynolds stress is defined as

$$\tau_{ij}^s = \bar{U}_i \bar{U}_j - \bar{U}_i \bar{U}_j. \quad (33)$$

As soon as we determine the value of τ_{ij}^s , we can solve (31) numerically, and this gives the filtered velocity field. In this case, the motions of large eddies will be considered and the motions of small eddies will be approximated.

However, there is no direct solution to the filtered equations, because the equations are not closed, meaning that the number of variables is larger than the number of equations. To be exact, the variables include \bar{U}_i , \bar{p} and $\bar{U}_i \bar{U}_j$, the number of which is greater than 4, the number of equations. Therefore, we need to add some restrictions on the sub-grid scale Reynolds stress,

and this leads to several different sub-models which will be introduced later. Major models include the Smagorinsky model, the dynamic model and the deconvolution model. Before we introduce the Smagorinsky model, the filtered rate of strain will be defined first.

For the filtered velocity field $\bar{\mathbf{U}}$, we define the tensor of filtered rate of strain as follows:

$$\bar{S}_{ij} = \frac{1}{2} \left(\frac{\partial \bar{U}_i}{\partial x_j} + \frac{\partial \bar{U}_j}{\partial x_i} \right). \quad (34)$$

Additionally, the filtered rate of strain \bar{S} is defined by

$$\bar{S} = \sqrt{2\bar{S}_{ij}\bar{S}_{ij}}, \quad (35)$$

using Einstein notation. This value is correlated to energy.

4.3 Smagorinsky model

Based on eddy viscosity, Smagorinsky proposed a model in 1963. The core idea of this model is that we restrain our sub-grid scale Reynolds stress in accordance to

$$\tau_{ij}^s - \frac{1}{3}\tau_{kk}^s\delta_{ij} = -\nu_t \left(\frac{\partial \bar{U}_i}{\partial x_j} + \frac{\partial \bar{U}_j}{\partial x_i} \right) = -2\nu_t \bar{S}_{ij}, \quad (36)$$

where δ is the Kronecker δ . This equation can be derived from energy production and dissipation of kinetic energy, which need further statistical hypothesis. The term with coefficient $1/3$ is due to isotropic part, while \bar{S}_{ij} is the symmetric-deviatoric part of filtered velocity, which all comes from the fact that a second-order tensor can be decomposed into an isotropic part, a symmetric-deviatoric part and an anti-symmetric part. Note that here ν_t represents the eddy viscosity, and should be chosen as

$$\nu_t = C_s^2 \rho \Delta^2 \bar{S}, \quad (37)$$

where C_s is the model parameter, and Δ is the size of the filter. The parameter C_s is not constant. Instead, it is selected based on the Reynolds number, and it may be different in different types of fluids. In the Smagorinsky model, the value of C_s varies from 0.2 to 0.065. The smaller it is, the less eddy viscosity the simulated profile has.

4.4 Dynamic model

The Smagorinsky model has one crucial problem, that is, the proper value of C_s keeps varying. To be exact, C_s should be zero in laminar flow, and should be small near the wall and should be large when the Reynolds number is large. This parameter is hard to determine, and the dynamic model adopts a convenient and efficient way to resolve this problem.

Except for the original filter K , this model introduces another additional filter \tilde{K} , whose

size $\tilde{\Delta}$ is greater than the original size Δ . This means the filter \tilde{K} is coarser than the original filter, and we are going to adopt the information we obtain after we exert this coarse filter to determine the Smagorinsky coefficient C_s .

According to previous notations, the original velocity is \mathbf{U} and the velocity after convolution with the filter K is $\bar{\mathbf{U}}$. Here we use $\tilde{\mathbf{U}}$ to denote the velocity field obtained by exerting convolution with \tilde{K} on $\bar{\mathbf{U}}$. In this way, we define

$$T_{ij} = \overline{U_i U_j} - \bar{U}_i \bar{U}_j, \quad (38)$$

$$L_{ij} = T_{ij} - \tilde{\tau}_{ij} = (\bar{U}_i \bar{U}_j) - \bar{U}_i \bar{U}_j. \quad (39)$$

Here the value L_{ij} is called *resolved stress* and can be calculated only in terms of \bar{U}_i . According to (36) and (37), we have

$$\tau'_{ij} = \tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} = -2C_s\Delta^2 \bar{S} \bar{S}_{ij}, \quad (40)$$

$$T'_{ij} = T_{ij} - \frac{1}{3}T_{kk}\delta_{ij} = -2C_s\tilde{\Delta}^2 \tilde{S} \tilde{S}_{ij}. \quad (41)$$

Based on the approximation

$$L_{ij} - \frac{1}{3}L_{kk}\delta_{ij} \approx T'_{ij} - \tilde{\tau}'_{ij}, \quad (42)$$

if we define

$$M_{ij} = 2\Delta^2 (\bar{S} \bar{S}_{ij}) - 2\tilde{\Delta}^2 \tilde{S} \tilde{S}_{ij}, \quad (43)$$

then we obtain

$$L_{ij} - \frac{1}{3}L_{kk}\delta_{ij} \approx -C_s M_{ij}. \quad (44)$$

Directly solving the equation, we discover that

$$C_s \approx -\frac{M_{ij} L_{ij}}{M_{kl} M_{kl}}. \quad (45)$$

To remove the fluctuation in calculating C_s based on the formula (45), we can calculate the average value in the numerator and denominator in (45). This model has been proven to be effective and give good results.

Though this model presents an tractable way to simulate turbulent flows approximately, there are still many weakness and difficulties in it, such as low accuracy.

4.5 Deconvolution Models

This model is different to the previous two models. The core idea is to calculate the unfiltered velocity field according to the filtered velocity field, instead of directly ignoring the small

eddies. If we have the unfiltered velocity, then we can compute the sub-grid scale Reynolds stress directly. In this way, the problem lies in estimation of unfiltered velocity.

We first expand (30) in a Taylor series at a point \mathbf{x} to second order. This will provide us a differential equation for the unfiltered velocity. Suppose the filter kernel is symmetric for \mathbf{x} , then the differential equation can be written as:

$$\bar{u}_i \approx u_i + \frac{\Delta^2}{24} \nabla^2 u_i. \quad (46)$$

To solve this equation, finding the approximate inversion is an appropriate idea. We may use Katapodes' estimation, which is

$$u_i \approx \bar{u}_i - \frac{\Delta^2}{24} \nabla^2 \bar{u}_i. \quad (47)$$

5 Models of Reynolds-averaged simulation

We then explain some details about the well-known and widely-used Reynolds-averaged simulation model here [14] [4].

5.1 Motivation

The advantages of the previously described models, the direct numerical simulation model and the large eddy simulation model, is that they contain many information of the fluid and therefore they are rather accurate on real situation. However, people usually focus on the average forces exerted on the fluid or the degree of mixing between two streams of fluids. Thus most of the information they provided is redundant and escape the range of interest. A century ago, Reynolds proposed the Reynolds-averaged simulation (RAS) model, aiming to simulate the average motion of fluid.

This model does not provide information not useful. Because of this, this model requires much less calculations than the previous two models if we want merely to get a stationary solution. Therefore, this model is one of the most commonly used models in turbulent flow simulation.

5.2 Basic model

Firstly, for every variable, we can write it as the sum of its locally time-averaged value and a oscillating residue as follows:

$$\phi(\mathbf{x}, t) = \langle \phi \rangle(\mathbf{x}) + \phi'(\mathbf{x}, t). \quad (48)$$

Here $\langle \phi \rangle$ is the average value of ϕ , which is defined as follows:

$$\langle \phi \rangle(\mathbf{x}) = \frac{1}{T} \int_t^{t+T} \phi(\mathbf{x}, s) ds, \quad (49)$$

where T is an interval shorter than our interest. To find the stationary solution, we may consider $T \rightarrow +\infty$. One may also consider the effect of chaos into consideration, and in this case $\langle \phi \rangle$ represents the expectation of ϕ under chaotic effect of initial values. The $\langle \phi \rangle$ is called *Reynolds average* formula in (48) is called *Reynolds decomposition*.

For the next step, we are going to give the definition of Reynolds stress. Note that

$$\langle U_i \phi \rangle = \langle (\langle U_i \rangle + U'_i)(\langle \phi \rangle + \phi') \rangle = \langle U_i \rangle \langle \phi \rangle + \langle U'_i \phi' \rangle. \quad (50)$$

The last term in (50) is zero if and only if ϕ' and U'_i are uncorrelated, and this situation rarely happens. Hence the *turbulent scalar flux* $\rho \langle U'_i \phi' \rangle$ needs to be introduced. If we plug U_j into ϕ , we obtain the *Reynolds stress* $\langle U'_i U'_j \rangle$. The Reynolds stress plays a similar role to sub-grids scale Reynolds stress in large eddy simulation model, although historically Reynolds stress was introduced earlier. Reynolds stress is the main part to be dealt with.

If we take the Reynolds average of incompressible Navier–Stokes equation, we can obtain the Reynolds-averaged Navier–Stokes equation:

$$\frac{\partial \langle U_i \rangle}{\partial t} + \frac{\partial}{\partial x_j} \left(\langle U_i \rangle \langle U_j \rangle + \langle U'_i U'_j \rangle \right) = -\frac{1}{\rho} \frac{\partial \langle p \rangle}{\partial x_i} + \frac{\partial \langle \tau_{ij} \rangle}{\partial x_j}, \quad (51)$$

$$\frac{\partial \langle U_i \rangle}{\partial x_i} = 0, \quad (52)$$

where $\langle \tau_{ij} \rangle$ are the mean viscous tensor elements as

$$\langle \tau_{ij} \rangle = \nu \left(\frac{\partial \langle U_i \rangle}{\partial x_j} + \frac{\partial \langle U_j \rangle}{\partial x_i} \right). \quad (53)$$

Similar to large eddy simulation model, the existence of Reynolds stress and turbulent scalar flux also means that the equations are not closed and hence impossible to solve directly. Therefore, we need to add more equations. This gives rise to the following two models: the $k-\epsilon$ model and the v2f model.

5.3 $k-\epsilon$ model

Since the energy dissipation and the transportation of mass link to the viscosity, we can simply assume that the effect of turbulent flows is represented by an increased viscosity. This leads to

the turbulent viscosity model

$$\langle U'_i U'_j \rangle - \frac{2}{3} k \delta_{ij} = -\nu_t \left(\frac{\partial \langle U_i \rangle}{\partial x_j} + \frac{\partial \langle U_j \rangle}{\partial x_i} \right), \quad (54)$$

where ν_t is the turbulent viscosity, and k is the turbulent kinetic energy:

$$k = \frac{1}{2} (\langle U'_x U'_x \rangle + \langle U'_y U'_y \rangle + \langle U'_z U'_z \rangle). \quad (55)$$

According to the turbulent viscosity model, we can obtain the effective viscosity as

$$\nu_{\text{eff}} = \nu + \nu_t. \quad (56)$$

Notice that if we specify ν_t , then we can find the solution to $\langle \mathbf{U} \rangle$ according to the averaged incompressible Navier-Stokes equations, with ν_{eff} replacing ν and $\langle p \rangle + \frac{2}{3} \rho k$ replacing p . The transformed equations have identical structure to incompressible Navier-Stokes equations, and similar solving techniques can be adopted.

Next we will talk about how to determine the value of ν_t . In the simplest model, we can just do the dimensional analysis and try to determine it. In this way, ν_t can be characterized by

$$\nu_t = C_\nu q L, \quad (57)$$

where C_ν is a dimensionless constant, L is the length scale of the turbulent flows and q here is a velocity with $q = \sqrt{2k}$. In order to determine L , the most popular model involves the dissipation term ϵ . This model constrains L as

$$\epsilon = \frac{k^{3/2}}{L}. \quad (58)$$

Therefore, the turbulent viscosity is

$$\nu_t = \rho C_\nu \frac{k^2}{\epsilon}. \quad (59)$$

Commonly the value C_ν is chosen as $C_\nu = 0.09$. Endowed with this, we can finish the calculation. This model is called $k-\epsilon$ model.

5.4 v2f model

There are also problems with the above $k-\epsilon$ model. The changes of the kinetic energy k and the dissipation ϵ are so rapid that we cannot find an appropriate model to depict them easily. Another problem is that proper boundary conditions near walls are hard to model. Boundary layers may occur and leads to computational issues. To be exact, the normal velocity \mathbf{V} and the fluctuation normal velocity V'^2 should be almost zero near walls, but it is very hard to enforce

such constraints for computation. Here v2f model aims to deal with this, by introducing another dumping function f . According to numerical experiments, this model can handle boundary conditions better while the cost is nearly the same as k - ϵ model.

6 Comparison between models

6.1 Analysis

In this section, we will compare the previously described three models, and present our originally written (except noted) analysis of the advantages and disadvantages of them.

For the direct numerical simulation model, it is the easiest one to implement, since the idea of this model is to simulate the fluids directly. Hence this model can be deployed not only to turbulent flow simulation, but also to all other computational fluid dynamics problems. Besides, since we do not make any approximation in the model, this model is the most accurate one in simulating turbulent flows. However, this model has some vital weakness, that is, the calculation of numerical simulation is so time-expensive that this model cannot be used in practice.

Some qualitative calculation can also be carried out [21]. According to the theory of Kolmogorov, the grid size Δ must be smaller than the scale of dissipation, and the region size L must be greater than the scale of energy. This means that the density of grids on one axis

$$N = L/\delta > O(\text{Re}^{3/4}), \quad (60)$$

where the power $3/4$ results in the ratio of scales in energy cascade models [14]. This means the number of cube grids is at least $N^3 > O(\text{Re}^{9/4})$. According to CFL condition, the Courant number should be controlled below 1, and therefore near linear correlation between time step and space step is required. This further implies that the complexity is roughly at least

$$N^4 > O(\text{Re}^3). \quad (61)$$

As mentioned before, in real world engineering, the Reynolds number Re is usually no less than about 10^4 , and this is definitely computational-impossible in a short period even with clusters. It should be emphasized that grids contradicting above constraint cannot capture turbulent flows appropriately, and therefore coarse grids do not make sense here.

The large eddy simulation model is an approximate model. With the idea that the small turbulent flows are insignificant to energy transportation and material exchanges, this model points out a way to eliminate the effect of small turbulent flows. Hence only large turbulent flows need to be calculated in this model, and small turbulent flows are directly ignored or approximated, say modeled. This simple operation gives rise to much reduction in complexity, and we can simulate in higher precision even on a coarse grid. This is a big progress of

turbulent flow simulation. However, since some approximation has been added on this model, the accuracy of the solution must have been deteriorated, especially when the grid size is not changed. Besides, though this model saves much time from the direct numerical simulation model, it is still so time-expensive that only some small scale simulation problems can be done with this model. Directly solving the turbulent simulation for large scale fluids still seems impossible even with this model.

Reynolds-averaged simulation is the most widely used model in turbulent flow simulation. Under the umbrella that people usually only care about the motion of fluids in a statistical averaged manner, this model divides the motion of fluids into the average motion and the fluctuation. What this model focuses on is to find solvable partial differential equations for the average motion. This idea brings to the definition and analysis of Reynolds stress. In general, the Reynolds-averaged simulation model does not take much time for stationary solutions, but from the solving process we know that the details along time will be lost. Thus this model is rather inaccurate, but it points out a path to simulate the motion of fluids in acceptable time. Therefore in the real engineering, only the time complexity of Reynolds-averaged simulation permits us to solve for a solution for turbulent flow simulation. In addition, Reynolds-averaged simulation model can handle “near steady” cases, in which the flow is not steady at all but statistical stability may be utilized.

In real world engineering, it is important to find a trade off between accuracy and efficiency. Therefore, Reynolds-averaged simulation enjoys high popularity. However, as the computational capability of computer increases, large eddy simulation model attracts more and more attention in recent days. Although direct numerical simulation is usually intractable, it is the golden standard for verification [21]. Due to the defects of these models, it is still a problem to increase either accuracy or efficiency and attain a better trade off nowadays.

Comparison between these models are summarized in Table 1.

Table 1 Comparison between models

Model	DNS	LES	RAS
Accuracy	High, standard	Medium	Low, acceptable
Efficiency	Low, unacceptable	Medium	High
Grid size	Dense	Relatively coarse	Relatively coarse
Time step	Small	Relatively large	Relatively large
Equations	Simple	Relatively complicated	Relatively complicated
Feature	High accuracy	Multi-scale capability	Handle statistical steadiness
Usage	Verification	Large scale	Stationary

6.2 Experiments

We will examine two well acknowledged profile here. The experiments are carried out originally.

The first numerical experiment is about the lid-driven cavity problem [2]. In this problem, a 2-dimensional square box having three fixed walls and a moving lid is filled with fluids. The size of the box is 0.1 m, and the velocity of moving wall is set to be 1 m s^{-1} . The kinetic viscosity ν is chosen to be $10^{-5} \text{ m}^2 \text{ s}^{-1}$ to achieve the Reynolds number $\text{Re} = 10^4$ here.

The results at 5 s of 80×80 grid are shown in Figure 1. For large eddy simulation, the basic Smagorinsky model is used. For Reynolds-averaged simulation, $k-\epsilon$ model is used.

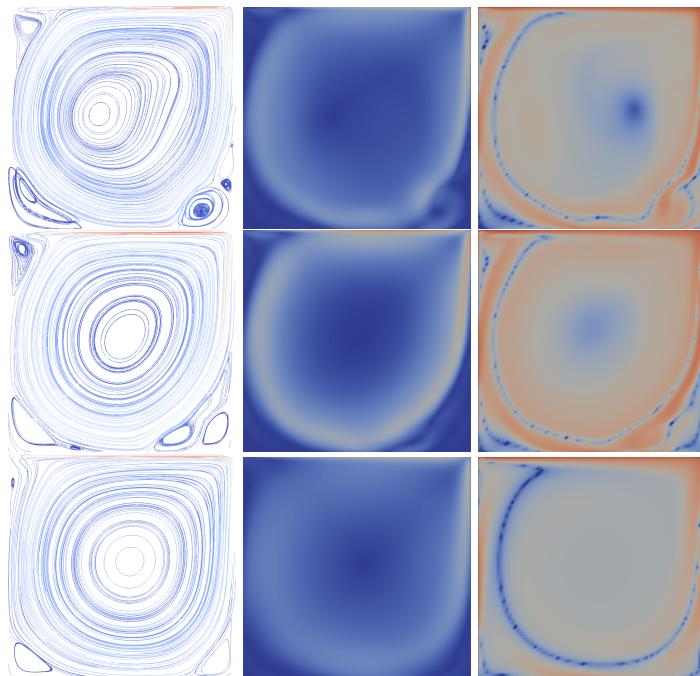


Figure 1 Simulation results at 5 s for 80×80 grid

First row: DNS, second row: LES, last row: RAS. First column: flow stream lines; second column: magnitude of flow velocity \mathbf{U} ; third column: logarithm of magnitude of vorticity ω . Blue means smaller quantities and red means larger. Best viewed in color and on digital display. (Digital version of this report can be found in <https://github.com/pppppass/TurbulentFlow/blob/master/Report/Report.pdf>)

From this figure, it can be easily verified that direct numerical simulation has the highest accuracy: small structures of vortexes are kept. Large eddy simulation removes some small structure, and biased the center of the largest vortex, while Reynolds-averaged simulation erases patterns of small scale almost completely.

However, to study the difference of efficiency, we have to study the dynamics of these models and related algorithms. The dynamic from 1 s to 5 s is shown in Figure 2.

From this figure, it can be directly observed that the direct numerical simulation on coarse grids does not converge into the stationary solution rapidly. On the contrary, the solution of large eddy simulation and Reynolds-averaged simulation converges faster because of the introduction of space or time averages. It is clear that Reynolds-averaged simulation gets stable in two seconds, and this proves that this simulation model can capture statistical steadiness. Another important comparison is between the first, second and fourth row: the result of direct

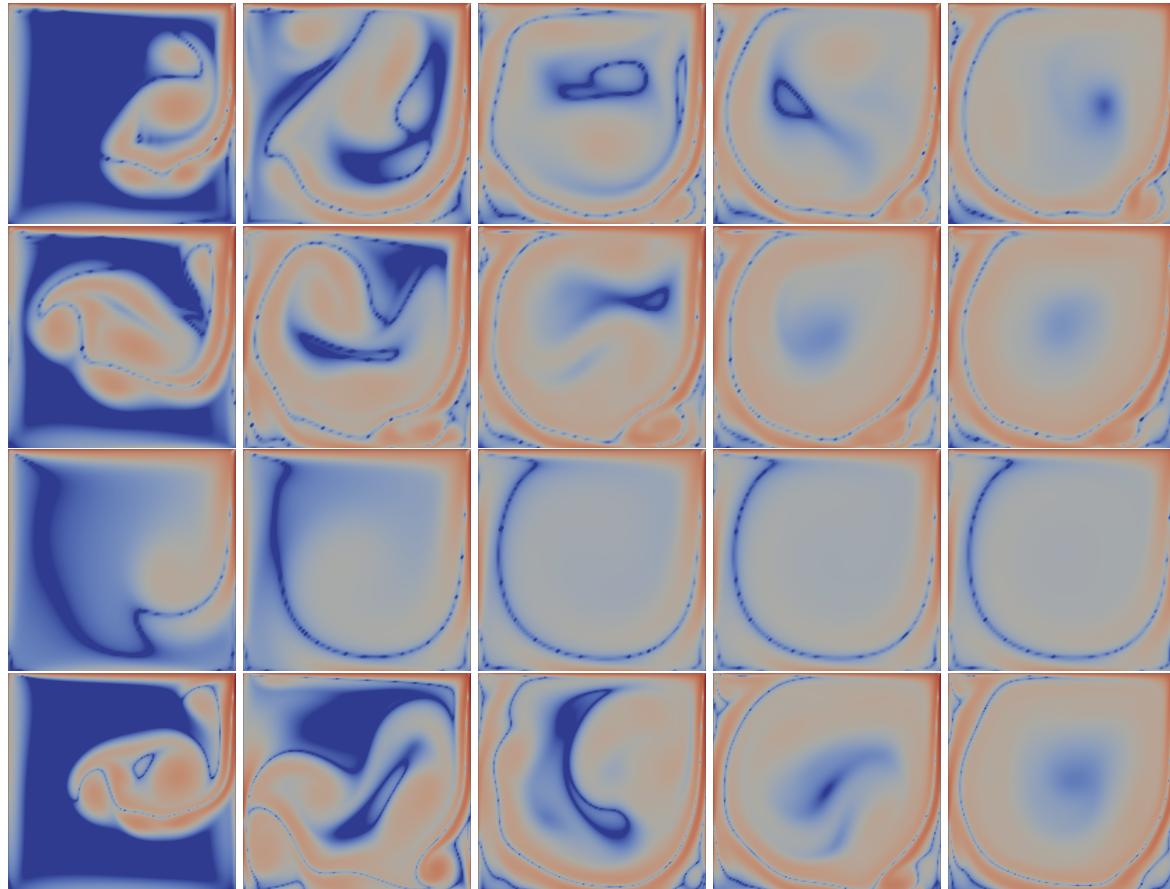


Figure 2 Simulation dynamics from 1 s to 5 s

These are figures of logarithm of magnitude of vorticity ω . First row: DNS for 80×80 grid; second row: LES for 80×80 grid; third row: RAS for 80×80 grid; fourth row: DNS for 160×160 grid. Columns from left to right represent 1, 2, 3, 4, 5 s respectively. Blue means smaller quantities and red means larger. Best viewed in color and on digital display.

numerical simulation on the fine grid is more similar to the result large eddy simulation on the coarse grid, especially at 1 s and 5 s. This tells the success of multi-scale modeling of large eddy simulation: the structures of small scales are directly modeled rather than computed, and therefore it is possible to simulation on a coarse grid using large eddy simulation as if direct numerical simulation is performed on a fine grid. This highly increases efficiency and cuts down computations.

We summarize the running time for different simulation models and grid sizes. This is shown in Table 2.

Table 2 Comparison on running time (s)

Grid size	DNS	LES	RAS
20×20	4.62	5.19	6.16
40×40	21.84	25.56	28.20
80×80	161.82	174.46	198.67
160×160	1380.04	1564.41	1890.42

Because the number of equations gets larger for large eddy simulation and Reynolds-averaged simulation, the running time also becomes longer. However, one should not forget about the grid size: from previous arguments, large eddy simulation behaves well on coarse grid, and significantly cuts down the running time. For Reynolds-averaged simulation, the system gets stable in a fast sense, and the simulated time can also be noticeably decreased. If we want to find the statistically stationary solution , Reynolds-averaged simulation can be even faster.

We examine the example of flows over a cylinder. The cylinder of radius 0.5 m is placed into a 2-dimensional rectangular flow pipe of width 2 m. The center of cylinder section is 2 m to the inlet and 18 m to the outlet. The inlet provides flow of velocity $U_0 = 1 \text{ m s}^{-1}$. To break the symmetry, we add $U'_0 = 0.1 \text{ m s}^{-1}$ vertically (the y axis). The kinetic viscosity is $5 \times 10^{-4} \text{ m s}^{-1}$. As a result, the Reynolds number is about 3000 (considering the half upper region and ignore U'_0).

The simulation result is shown in Figure 3. Kármán vortex street can be observed.

In this figure, the distinction among these three models are again magnified. The comparison between the first two rows shows that the vortex street of large eddy simulation is more regular and continuous, while that of direct numerical simulation introduces some artifacts here. To be exact, the vortex street in direct numerical simulation gets broken up and the size of vortices varies a lot, which is not expected. This is because the size of grids does not reach three fourth power of the Reynolds number, as mentioned above, and therefore direct numerical simulation cannot capture such details. For large eddy simulation, part of the vortices are modeled and therefore accuracy and regularity can be preserved. Note that Reynolds-averaged simulation fails here, because the profile itself does not have statistical steadiness: it has periodical

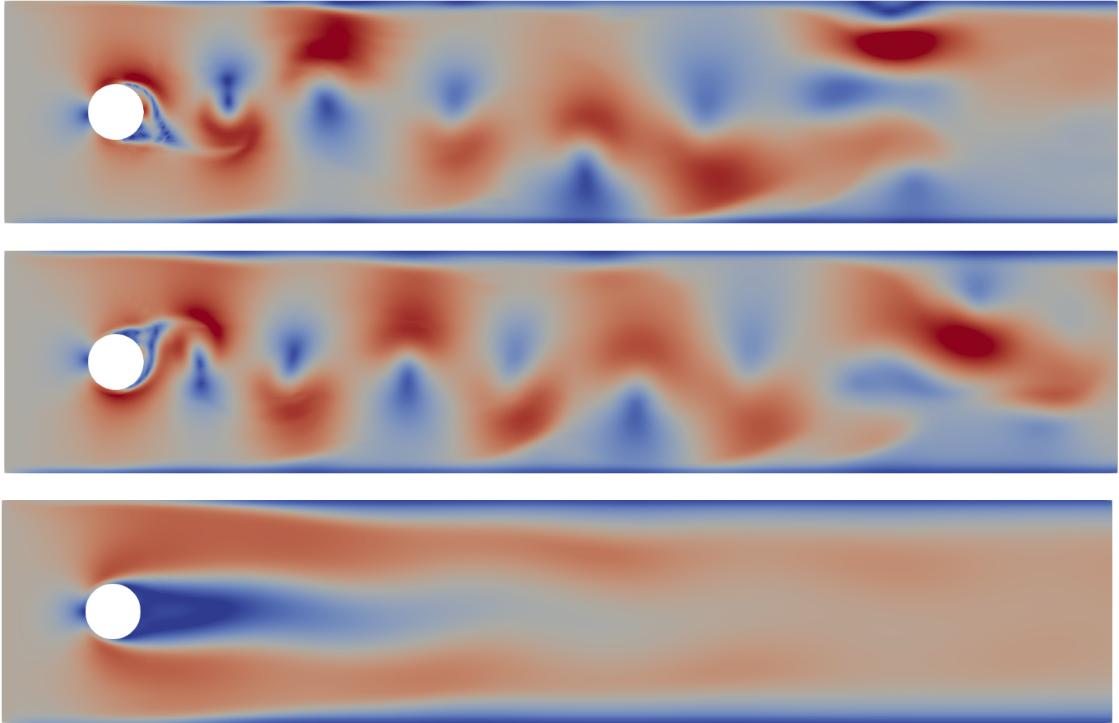


Figure 3 Simulations result of flows over a cylinder at 20 s

These are figures of magnitude of flow velocity \mathbf{U} . First row: DNS; second row: LES; last row: RAS. Blue means smaller quantities and red means larger. Best viewed in color and on digital display.

movement after all.

These results are carried out on a computer with Intel Core i7-6500U (4 threads) and 7886 MiB RAM. The operating system is Arch Linux 4.16.13 64-bit. We deploy OpenFOAM 5.0 [19] for numerical simulation. Detailed settings including initial values, boundary conditions and numerical schemes are provided in <https://github.com/pppppass/TurbulentFlow/tree/master/Experiments>.

6.3 Further discussion and novel ideas

We present some further discussion about the three models and present a novel idea for turbulent flow modeling in this sub-section.

Generally speaking, the Reynolds-averaged simulation is “one point correlation” model, which pays attention on the probability distribution of velocity (and pressure) on one point. In the work related to Kolmogorov [10] [17] [14], statistical behavior of two point is analyzed to get the energy spectrum. However, such “two points correlation” model is under development. The computational difficulty lies here because the problem size get squared, and also it is not clear about how we can utilize the correlation of velocity of a pair of points.

In another sense, the large eddy simulation model emulates the correlation among points of a local area: it makes use of a filter and try to stabilize the quantities. However, the filter is

directly given: it force use to filter out fine details, or say some special kind of correlation is retained. This leads to the loss of high frequency signals.

Another problem should also be considered. For Reynolds-averaged simulation, we have only the expectation, say Reynolds-averaged velocity $\langle U_i \rangle$ and the covariance matrix, say Reynolds stress $\langle U_i U_j \rangle$ for the three-dimensional random variable \mathbf{U} . Therefore, it is rather impossible to capture the probability distribution, without statistical hypothesis of Gaussian distribution. However, it can be a problem, if we want to find the “statistical stable” solution and then generated random configuration of \mathbf{U} from it. In other words, only the expectation and covariance can be solved, and real turbulent flows cannot be simulated from Reynolds-averaged simulation model. The failure of flows over a cylinder also exemplifies this issue.

All problems boil down to capture a complicated probability distribution. In the communities of statistics and machine learning, there are a lot of methods to emulate and sample from a probability distribution. Recent development of machine learning and deep learning shows the potential of such distribution probability of data-driven methods. Additionally, according to the first hypothesis of Kolmogorov [10], the turbulent flows of small scales are independent to each other, and therefore some locality is inherited then.

Borrowing the ideas of data-driven methods, which worths a trial is to use a generative model to capture the probability distribution of turbulent flows. Such generative model can be a deep belief network [7] [12] or a deep Boltzmann machine [16] [13] [15]. Recent development in deep learning also provides us generative models like variational autoencoders [9] [3] and generative adversarial networks [6] [1]. Physics law should also be considered as constrains. This means that we need to carefully design the architecture of generative model. We may use sampling techniques like Monte Carlo Markov Chain to emulate turbulent flows.

In conclusion, we may combine statistical models or data-driven models to capture the probability distribution of turbulent flows, including both many-point correlation and detailed density distribution of velocity. The key idea is to use a generative model to capture the chaos and relative probability distributions, with physics laws coupled. Such model is expected to capture the chaotic behavior of turbulent flows, and thus provides us a new aspect of turbulent flow simulation. This is a recently developed idea for novelty, and further research may be carried on.

7 Conclusion

This report presents an introduction to turbulent simulation, including the mathematical formulation and three different models: direct numerical simulation, large eddy simulation and Reynolds-averaged simulation. Comparison among these three models are also showed in this report. We also present detailed analysis and numerical experiments on various cases with these three models, and show the distinctions among these models.

Turbulent flows are of high importance in computational mathematics and engineering. It

is also an interesting question whether it is possible to combine classical models with recent popular learning-based algorithms. One of the key challenge for learning-based algorithms is to “learn” the model directly. After all, in some sense, the problem of turbulent flows is mainly a problem of models.

8 Acknowledgement

Ten days ago, we had no acquaintance with anything in the field of computational fluid dynamics and turbulent flow simulation at all. Although we finished the literature survey and comparative analysis and experiments on our own, we express our thanks to Zexing Li, whose talk initiated us to begin such an adventure.

9 Reference

References

- [1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. “Wasserstein GAN”. In: *arXiv:1701.07875 [cs, stat]* (Jan. 26, 2017). arXiv: [1701.07875](https://arxiv.org/abs/1701.07875).
- [2] Charles-Henri Bruneau and Mazen Saad. “The 2D lid-driven cavity problem revisited”. In: *Computers & Fluids* 35.3 (Mar. 1, 2006), pp. 326–348. ISSN: 0045-7930. doi: [10.1016/j.compfluid.2004.12.004](https://doi.org/10.1016/j.compfluid.2004.12.004).
- [3] Carl Doersch. “Tutorial on Variational Autoencoders”. In: *arXiv:1606.05908 [cs, stat]* (June 19, 2016). arXiv: [1606.05908](https://arxiv.org/abs/1606.05908).
- [4] Joel H. Ferziger and M. Peri. *Computational methods for fluid dynamics*. 3rd, rev. ed. Berlin ; New York: Springer, 2002. 423 pp. ISBN: 978-3-540-42074-3.
- [5] George E. Forsythe and Wolfgang R. Wasow. *Finite-Difference Methods for Partial Differential Equations, Applied Mathematics Series*. 1960.
- [6] Ian Goodfellow. “NIPS 2016 Tutorial: Generative Adversarial Networks”. In: *arXiv:1701.00160 [cs]* (Dec. 31, 2016). arXiv: [1701.00160](https://arxiv.org/abs/1701.00160).
- [7] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. “A fast learning algorithm for deep belief nets”. In: *Neural computation* 18.7 (2006), pp. 1527–1554.
- [8] J. P. Holman. “Heat transfer, 1986”. In: *Mc GranHill Book Company, Soyhern Methodist University* (1986).
- [9] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *arXiv:1312.6114 [cs, stat]* (Dec. 20, 2013). arXiv: [1312.6114](https://arxiv.org/abs/1312.6114).
- [10] Andrej Nikolaevich Kolmogorov. “Equations of turbulent motion in an incompressible fluid”. In: *Dokl. Akad. Nauk SSSR*. Vol. 30. 1941, pp. 299–303.

- [11] L. D. Landau and E. M. Lifshitz. *Fluid Mechanics, Second Edition: Volume 6*. 2 edition. Amsterdam u.a: Butterworth-Heinemann, Jan. 15, 1987. 552 pp. ISBN: 978-0-7506-2767-2.
- [12] Honglak Lee et al. “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations”. In: *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 609–616.
- [13] Vinod Nair and Geoffrey E. Hinton. “Rectified linear units improve restricted boltzmann machines”. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814.
- [14] Stephen B. Pope. *Turbulent flows*. IOP Publishing, 2001.
- [15] Ruslan Salakhutdinov and Hugo Larochelle. “Efficient learning of deep Boltzmann machines”. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 2010, pp. 693–700.
- [16] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. “Restricted Boltzmann machines for collaborative filtering”. In: *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 791–798.
- [17] Katepalli R. Sreenivasan. “On the universality of the Kolmogorov constant”. In: *Physics of Fluids* 7.11 (Nov. 1, 1995), pp. 2778–2784. ISSN: 1070-6631. doi: [10.1063/1.868656](https://doi.org/10.1063/1.868656).
- [18] G. I. Taylor. “Diffusion by Continuous Movements”. In: *Proceedings of the London Mathematical Society* s2-20.1 (1922), pp. 196–212. ISSN: 1460-244X. doi: [10.1112/plms/s2-20.1.196](https://doi.org/10.1112/plms/s2-20.1.196).
- [19] H. G. Weller et al. “A tensorial approach to computational continuum mechanics using object-oriented techniques”. In: *Computers in Physics* 12.6 (Nov. 1, 1998), pp. 620–631. ISSN: 0894-1866. doi: [10.1063/1.168744](https://doi.org/10.1063/1.168744).
- [20] Frank M. White. *Fluid Mechanics Fourth Edition*. McGraw-hill, 2017.
- [21] Zhaoshun Zhang, Guixiang Cui, and Chunxiao Xu. *Theory and Modeling of Turbulence*. Tsinghua University Press, 2005.