Direct Numerical Simulation of Particle-laden Isotropic Turbulence

Kyongmin Yeo

Computational Turbulence Lab.
Department of Mechanical Engineering
Yonsei University

September 23, 2005

Contents

Chapter 1

Navier-Stokes solver

1.1 Navier-Stokes equations in Fourier series

The Navier-Stokes equations in the rotational form are

$$\frac{\partial u_i}{\partial t} = -\frac{\partial P}{\partial x_i} + H_i + \nu \nabla^2 u_i, \tag{1.1}$$

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{1.2}$$

where

$$\begin{aligned} P &= \frac{p}{\rho} + \frac{1}{2} u_j u_j, \\ H_i &= \epsilon_{ijk} u_j \omega_k. \end{aligned}$$

The pressure Poisson equation can be obtained by taking the divergence of equation (1.1),

$$\nabla^2 P = \frac{\partial H_j}{\partial x_j}. \tag{1.3}$$

Expanding equations (1.1, 1.3) in Fourier series gives

$$\frac{d\hat{u}_i}{dt} = -ik_i \hat{P} + \hat{H}_i - \nu k^2 \hat{u}_i, \tag{1.4}$$

$$-k^2 \hat{P} = ik_j \hat{H}_j, \tag{1.5}$$

where $\hat{}$ denotes a Fourier coefficient, k_i is a wavenumber in i-direction, and $k^2 = k_1^2 + k_2^2 + k_3^2$. To eliminate the pressure in the Navier-Stokes equations, substituting equation (1.5) into equation (1.4) yields,

$$\frac{d\hat{u}_i}{dt} = -\frac{1}{k^2} k_i k_j \hat{H}_j + \hat{H}_i - \nu k^2 \hat{u}_i. \tag{1.6}$$

Table 1.1: Time stepping coefficients for RK3

	a_n	b_n	$c_n = \sum_{i=1}^{n}(a_{i-1} + b_{i-1})$
1^{st}	8/15	0	0
2^{nd}	5/12	-17/60	8/15
3^{rd}	3/4	-5/12	2/3

To treat the viscous terms analytically, one can multiply equation (1.6) by an exponential function,

$$f(t) = e^{\nu k^2 t}. \tag{1.7}$$

Then, the resulting equation reads

$$f(t)\frac{d\hat{u}_i}{dt} + \hat{u}_i\frac{df(t)}{dt} = \left[-\frac{1}{k^2}k_i k_j \hat{H}_j + \hat{H}_i\right] \times f(t), \tag{1.8}$$

and thus

$$\frac{d\hat{u}_i f(t)}{dt} = \left[-\frac{1}{k^2}k_i k_j \hat{H}_j + \hat{H}_i\right] \times f(t). \tag{1.9}$$

To solve equation (1.9) numerically, applying a 3^{rd}-order 3-step Runge-Kutta scheme (RK3) to the nonlinear terms gives,

$$\hat{u}_i^{n+1} = \left[a_n dt \widehat{NL}_i^n + \hat{u}_i^n\right] e^{\nu k^2 (c_n - c_{n+1})dt} + b_n dt \widehat{NL}_i^{n-1} e^{\nu k^2 (c_{n-1} - c_{n+1})dt}, \tag{1.10}$$

or

$$\hat{u}_i^{n+1} = \left[a_n dt \widehat{NL}_i^n + \hat{u}_i^n\right] e^{-\nu k^2 (a_n + b_n)dt} + b_n dt \widehat{NL}_i^{n-1} e^{-\nu k^2 (a_n + b_n + a_{n-1} + b_{n-1})dt}, \tag{1.11}$$

where

$$\widehat{NL}_i = -\frac{1}{k^2}k_i k_j \hat{H}_j + \hat{H}_i, \tag{1.12}$$

and a_n, b_n, and c_n denote the Runge-Kutta coefficients (Table 1.1). In the limit of infinite Reynolds number the RK3 method approaches the CFL number limit $\sqrt{3}$ [1].

1.2 External forcing

The Navier-Stokes equations with an external forcing to maintain stationary turbulence are

$$\frac{d\hat{u}_i}{dt} = -ik_i\hat{P} + \hat{H}_i - \nu k^2\hat{u}_i + \hat{f}_i, \tag{1.13}$$

in which $\hat{f}_i$ denotes an external forcing. To ensure the divergence-free condition, $\hat{f}_i$ is defined to be the projection of a vector $\hat{\boldsymbol{b}}$ onto the plane normal to the wavenumber vector $\boldsymbol{k}$,

$$\hat{f}_i = \hat{b}_i - \frac{k_i}{k^2}(k_j\hat{b}_j). \tag{1.14}$$

Following the study by Eswaran & Pope [2], a three-dimensional complex vector $\hat{\boldsymbol{b}}$ is non-zero in the range $0 < k \leq k_f$, in which k_f is the maximum forcing wavenumber, and composed of six independent Uhlenbeck-Ornstien processes,

$$\hat{\boldsymbol{b}} = \begin{bmatrix} UO1 \\ UO3 \\ UO5 \end{bmatrix} + i\begin{bmatrix} UO2 \\ UO4 \\ UO6 \end{bmatrix}. \tag{1.15}$$

Each stochastic process, $UO1 - UO6$, is chosen so as to satisfy the Langevin equation with a time scale T_L^f and standard deviation σ_f [3], that is

$$dUO = -\frac{UO}{T_L^f}dt + \left(\frac{2\sigma_f^2}{T_L^f}\right)^{1/2} dW_t, \tag{1.16}$$

in which W_t denotes a Wiener process such that,

$$dW_t \sim \mathcal{N}(0, dt). \tag{1.17}$$

An analytical solution to equation (1.16) is given by [4, 5]

$$UO(t) = UO(0)e^{-t/T_L^f} + e^{-t/T_L^f}\int_0^t e^{s/T_L^f}\left(2\sigma_f^2/T_L^f\right)^{1/2} dW_s. \tag{1.18}$$

Equation (1.18) can be solved discretely by applying Itô integral [4]. For each substep of RK3, equation (1.18) reads [6]

$$UO^{n+1} = e^{-(a_n+b_n)dt/T_L^f}\left[UO^n + \left(2\sigma_f^2/T_L^f\right)^{1/2} dW^n\right], \tag{1.19}$$

in which discretized Wiener process is

$$dW^n \sim \mathcal{N}(0, (a_n + b_n)dt). \tag{1.20}$$

Table 1.2: Simulation parameters

N	ν	k_f	ϵ^f	T_L^f
64^3	0.0158	$\sqrt{2}$	0.01306	0.6369
128^3	0.006416	$\sqrt{2}$	0.01581	0.6369

Eswaran & Pope [2] suggested an empirical relation between a desired Reynolds number and the forcing parameters:

$$Re_\lambda \simeq \frac{8.5}{(\eta_T k_0)^{5/6} N_F^{2/9}}, \tag{1.21}$$

in which N_F is the number of forced modes, η_T denotes the predicted Kolmogorov lengthscale, and k_0 is the smallest wavenumber. For two values of k_f, $\sqrt{2}$ and $2\sqrt{2}$, the corresponding values of N_F are 18 and 92, respectively. Once, η_T is determined for the desired Re_λ, the rest forcing parameters are obtained by the following set of equations,

$$\begin{aligned} \eta_T &\equiv (\nu^3/\epsilon_T^*)^{1/4}, & (1.22)\\ \epsilon_T^* &\equiv \frac{4\epsilon^f N_F}{1 + T_L^f (N_F)^{1/3}/\beta}, & (1.23)\\ \epsilon^f &\equiv \sigma_f^2 T_L^f, & (1.24) \end{aligned}$$

in which ϵ_T^* is the predicted dissipation rate and β is a constant.

1.3 Results

The direct numerical simulations using 64^3 and 128^3 grid points are performed. The stochastic forcing suggested by Eswaran & Pope [2] is employed to maintain stationary turbulent flows. The forcing parameters for each simulations are shown table 1.2.

Turbulence field generated by random Gaussian numbers was used as an initial condition. Figure 1.1 shows the evolution of turbulent quantities with time for 64^3 simulation. The definitions of dissipation rate (ϵ) is

$$\epsilon = \frac{1}{2}\nu \left\langle \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \right\rangle. \tag{1.25}$$

Although it is inadequate to assert that the turbulent field is stationary only based on the time histories of dissipation rate and turbulent intensities, it seems that the

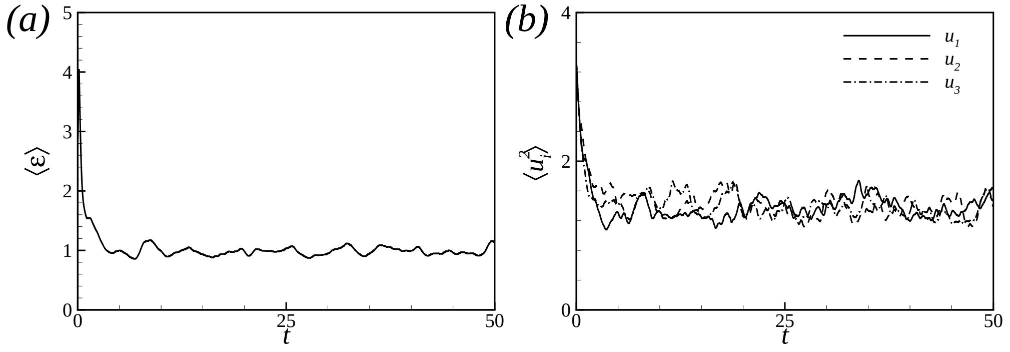

Figure 1.1: Temporal evolution of (a) dissipation rate and (b) turbulent intensities from an initial field generated by Gaussian random numbers.

memory of initial condition is lost quickly and, soon after the short period of initial transient state, turbulent field rapidly moves toward isotropic and stationary state. The simulation was performed for sufficiently long time - in this case, about 20 times longer than the initial transition - and then, the turbulent field was saved as an initial condition.

The Eulerian statistics for 64^3 and 128^3 cases are shown in table 1.3. The Taylor microscale in isotropic turbulence can be obtained by the following relationship,

$$\langle \epsilon \rangle = 15\nu \langle u^2 \rangle / \lambda^2. \tag{1.26}$$

Yeung & Pope [7] suggested that it is desirable to maintain the ratio of the smallest scale of motion to the grid size larger than 0.5. In other words,

$$k_{max}\eta \geq 1.5, \tag{1.27}$$

in which, k_{max} is the maximum resolved Fourier mode and η is the Kolmogorov lengthscale ($\equiv (\nu^3/\epsilon)^{1/4}$). The computational grids in 64^3 and 128^3 simulations are a little coarser than the recommended grid size. This can be adjusted by varying forcing parameters. Figure 1.2 displays one-dimensional energy spectra. The small increase of energy at high wavenumbers is a consequence of aliasing error and can be eliminated by enhancing grid resolution. However, because the error is very small

Table 1.3: Eulerian statistics

N	Re_λ	$\langle u^2 \rangle^{1/2}$	$\langle \epsilon \rangle$	η	$k_{max}\eta$
64^3	58	1.347	0.942	0.0452	1.36
128^3	92	1.342	0.853	0.0236	1.44

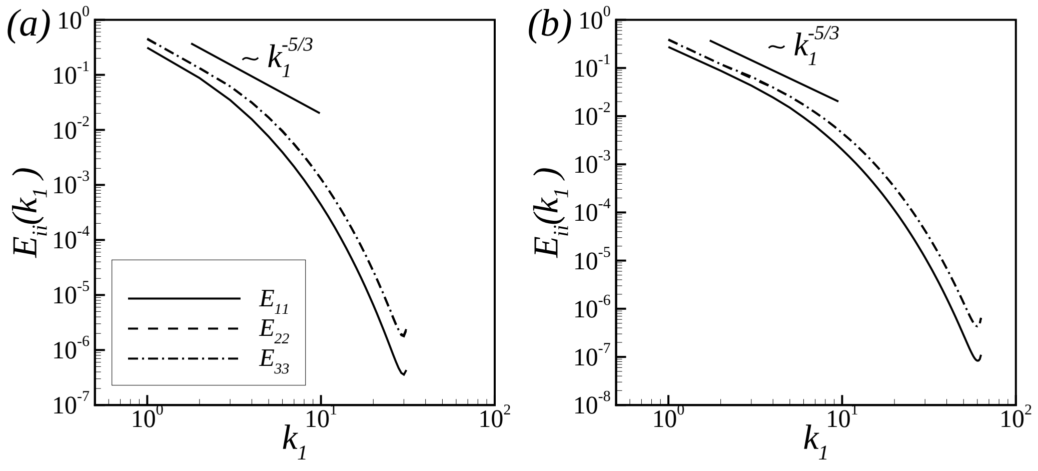

Figure 1.2: One-dimensional energy spectra for (a) 64^3 and (b) 128^3 simulations.

and appears only at the highest wavenumber, there seems no need to adjust the grid resolutions. In figure 1.2, it is shown that the energy spectra at low wavenumbers show the Kolmogorov's -5/3 scaling. It is also shown that the inertial range becomes widened with increasing Reynolds number. The transverse energy spectra, $E_{22}(k_1)$ and $E_{33}(k_1)$, fall on a curve, indicating that the velocity field is isotropic.

Chapter 2

Algorithm for tracking particles

2.1 Derivation of 3-dimensional 4-point Hermite interpolation

Typically, locations of particles do not coincide with the locations of computational grid where the flow quantities, such as velocities, pressures, and accelerations, are evaluated. Therefore, it is essential to obtain the fluid quantities at the position of a particle by employing an accurate interpolation scheme. Undoubtedly, the direct summation of phase-shifted Fourier coefficients, known as a spectral interpolation, is the most exact method [8]. Considering its unpractically high computational cost, however, it seems not feasible to employ the spectral interpolation to obtain Lagrangian statistics. Recently, Choi *et al.* [9] showed that a 4th-order Hermite interpolation combined with the Chebyshev polynomials in one-direction outperforms other tested interpolations schemes, such as linear, 6th-order Lagrange polynomial, and 2-point Hermite interpolations, with reasonable computation time. Therefore, in this work, the 4-point Hermite interpolation in 3-dimension is adopted to obtain flow quantities in the location of a particle.

The 4-point Hermite interpolation in x-direction is given by

$$u(x,m,n) = \sum_{l=1}^{4} u(l,m,n)H_l + \frac{\partial u(l,m,n)}{\partial x}G_l, \tag{2.1}$$

in which H_l and G_l denote basis functions of Hermite interpolation. Extension of equation (2.1) to 2-dimension is

$$\begin{aligned} u(x,y,n) &= \sum_{m=1}^{4} u(x,m,n)H_m + \frac{\partial u(x,m,n)}{\partial y}G_m \\ &= \sum_{m=1}^{4} \left[\sum_{l=1}^{4} u(l,m,n)H_l + \frac{\partial u(l,m,n)}{\partial x}G_l\right] H_m \\ &\quad + \frac{\partial}{\partial y}\left[\sum_{l=1}^{4} u(l,m,n)H_l + \frac{\partial u(l,m,n)}{\partial x}G_l\right] G_m \\ &= \sum_{m=1}^{4}\sum_{l=1}^{4} u(l,m,n)H_lH_m + \frac{\partial u(l,m,n)}{\partial x}G_lH_m \\ &\qquad + \frac{\partial u(l,m,n)}{\partial y}H_lG_m + \frac{\partial^2 u(l,m,n)}{\partial x\partial y}G_lG_m. \end{aligned} \tag{2.2}$$

Similarly, the 3-dimensional 4-point Hermite interpolation reads,

$$\begin{aligned} u(x,y,z) &= \sum_{n=1}^{4} u(x,y,n)H_n + \frac{\partial u(x,y,n)}{\partial z}G_n \\ &= \sum_{n=1}^{4}\sum_{m=1}^{4}\sum_{l=1}^{4} u(l,m,n)H_lH_mH_n + \frac{\partial u(l,m,n)}{\partial x}G_lH_mH_n \\ &\qquad + \frac{\partial u(l,m,n)}{\partial y}H_lG_mH_n + \frac{\partial u(l,m,n)}{\partial z}H_lH_mG_n \\ &\qquad + \frac{\partial^2 u(l,m,n)}{\partial x\partial y}G_lG_mH_n + \frac{\partial^2 u(l,m,n)}{\partial x\partial z}G_lH_mG_n \\ &\qquad + \frac{\partial^2 u(l,m,n)}{\partial y\partial z}H_lG_mG_n + \frac{\partial^3 u(l,m,n)}{\partial x\partial y\partial z}G_lG_mG_n. \end{aligned} \tag{2.3}$$

The basis functions of 4-point Hermite interpolation are given by [9]

$$\begin{aligned} H_1(\xi) &= (11\xi^7 - 52\xi^6 + 59\xi^5 + 50\xi^4 - 124\xi^3 + 56\xi^2)/108, \\ H_2(\xi) &= (27\xi^7 - 81\xi^6 - 54\xi^5 + 270\xi^4 + 27\xi^3 - 297\xi^2)/108 + 1, \\ H_3(\xi) &= (-27\xi^7 + 108\xi^6 - 27\xi^5 - 270\xi^4 + 108\xi^3 + 216\xi^2)/108, \\ H_4(\xi) &= (-11\xi^7 + 25\xi^6 + 22\xi^5 - 50\xi^4 - 11\xi^3 + 25\xi^2)/108, \\ G_1(\xi) &= (3\xi^7 - 15\xi^6 + 21\xi^5 + 3\xi^4 - 24\xi^3 + 12\xi^2)h/108, \\ G_2(\xi) &= (27\xi^7 - 108\xi^6 + 54\xi^5 + 216\xi^4 - 189\xi^3 - 108\xi^2)h/108 + h\xi, \\ G_3(\xi) &= (27\xi^7 - 81\xi^6 - 27\xi^5 + 189\xi^4 - 108\xi^2)h/108, \\ G_4(\xi) &= (3\xi^7 - 6\xi^6 - 6\xi^5 + 12\xi^4 + 3\xi^3 - 6\xi^2)h/108, \end{aligned} \tag{2.4}$$

where $\xi = (x - x_2)/h$, $\xi = (y - y_2)/h$, or $\xi = (z - z_2)/h$ and h is the grid spacing in each direction.

2.2 Tracking scheme for a fluid particle

2.2.1 Equation of motion

The trace of a particle is obtained by integrating the equation of motion of the particle. The equation of motion for a fluid particle is

$$\frac{d\boldsymbol{X}(t,\boldsymbol{X}_0)}{dt} = \boldsymbol{V}(t,\boldsymbol{X}_0), \tag{2.5}$$

where $\boldsymbol{X}$ is the location of a fluid particle at time t, whose initial position is $\boldsymbol{X}_0$, and $\boldsymbol{V}$ is the velocity of a fluid particle at time t. Since a fluid particle exactly follows the motion of fluid, equation (2.5) can be numerically integrated easily once the fluid velocity at $\boldsymbol{X}$ is obtained by using the interpolation scheme shown in the previous section.

In accordance with the Navier-Stokes solver, equation (2.5) is integrated employing RK3 scheme. The discretized equation is

$$X_i^{n+1} = X_i^n + a_n dt V_i^n + b_n dt V_i^{n-1}. \tag{2.6}$$

Because V_i should be evaluated for each fluid particles, computational cost of equation (2.6) is very high. To reduce the computational cost, the 4th-order central difference scheme is used to calculate the derivatives in the interpolation scheme.

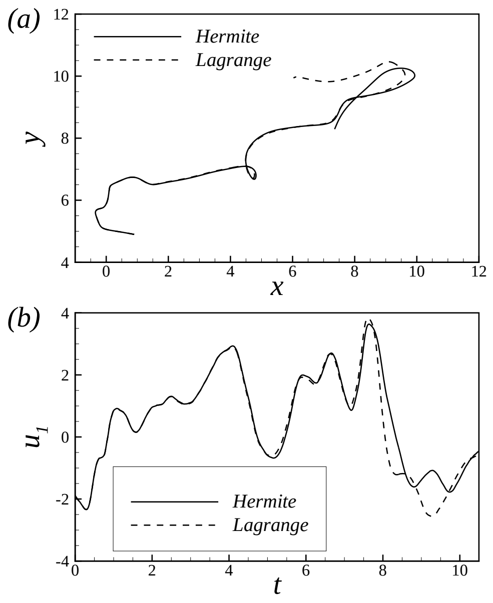

Figure 2.1: (a) Sample particle trajectories obtained by Hermite and Lagrange interpolations on $x-y$ plane and (b) time histories of u_1.

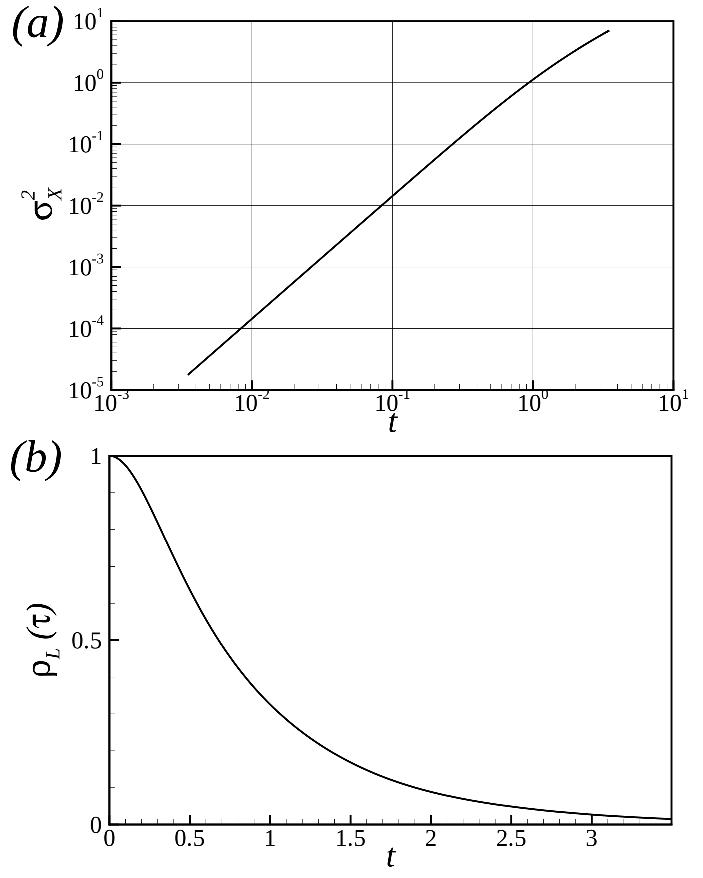

Figure 2.2: (a) Mean-square dispersion and (b) Lagrangian velocity auto-correlation function for $Re_\lambda = 58$.

2.2.2 Results

To verify the interpolation scheme, a particle trajectory obtained employing the Hermite interpolation is compared to that using 6th-order Lagrange interpolation [8]. Figure 2.1 shows sample trajectories and time histories of u_1. In early time, both Hermite and Lagrange interpolation gives almost the same result. The trajectories start to deviate from each other when the fluid particle shows a rotational motion in late time. This deviation is an anticipated result because of the low accuracy of the Lagrange interpolation [9].

The mean-square dispersion and Lagrangian auto-correlation function are defined as

$$\sigma_X^2 = \langle (X(t_0+\tau) - X(t_0))^2 \rangle, \quad (2.7)$$

$$\rho_L(\tau) = \frac{\langle \phi(t_0+\tau)\phi(t_0) \rangle}{\langle \phi^2(t_0) \rangle}, \quad (2.8)$$

in which τ denotes time lag. σ_X^2 and the Lagrangian velocity auto-correlation for

64^3 simulation are shown in figure 2.2. It is shown that the early-time dispersion is proportional to t^2, as predicted by Taylor's theory [7, 9]. The integral timescale (T_L) can be obtained by integrating the Lagrangian velocity auto-correlation (figure 2.2 *b*). In this simulation, the ratio of integral timescale to the Kolmogorov timescale ($\tau_\eta \equiv (\nu/\epsilon)^{1/2}$) is about 6.95, which is in good agreement with the previous results by Yeung & Pope [7].

2.3 Tracking scheme for a heavy particle

2.3.1 Equation of motion

Equation of motion for a heavy particle in turbulence, omitting high-order terms, is given by [10]

$$m_p \frac{dV_i}{dt} = (m_p - m_f) g_i + m_f \frac{Du_i}{dt} - \frac{1}{2} m_f \frac{d}{dt}(V_i - u_i) - 6\pi a \mu (V_i - u_i), \qquad (2.9)$$

in which m_p is the mass of a particle, M_f is the mass of fluid displaced by the particle, g_i is the gravitational acceleration, and a represent the radius of a particle. The terms on the right hand side of equation (2.9) represent the force due to buoyancy, fluid acceleration, inertia of added mass, and Stokes drag, respectively. Dividing equation (2.9) by m_p and rearranging yields,

$$(\lambda + 2)\frac{dV_i}{dt} = 2(1-\lambda) g_i + 2\lambda \left(\frac{Du_i}{Dt} + \frac{1}{2}\frac{du_i}{dt} \right) - \frac{12\pi a \mu}{m_p}(V_i - u_i), \qquad (2.10)$$

where $\lambda = \rho_f/\rho_p$ and ρ_p and ρ_f are, respectively, the density of a particle and fluid. From the fact that

$$\begin{aligned} \rho_p &\gg \rho_f, \\ m_p &= \frac{4}{3}\pi a^3 \rho_p, \end{aligned}$$

a simplified equation of motion for a heavy particle is

$$\frac{dV_i}{dt} = \frac{1}{\tau}(u_i - V_i) + g_i. \qquad (2.11)$$

Here, τ ($\equiv 2\rho_p a^2/9\mu$) is a characteristic timescale of a particle, namely the particle response time [11]. Note that equation (2.11) holds only when the particle Reynolds number based on the relative velocity between ambient fluid and a particle is significantly smaller than 1 [12, 13, 14].

As the particle response time decreases, equation (2.11) becomes stiffer to solve numerically [14]. To avoid this problem, the equation of motion in terms of relative

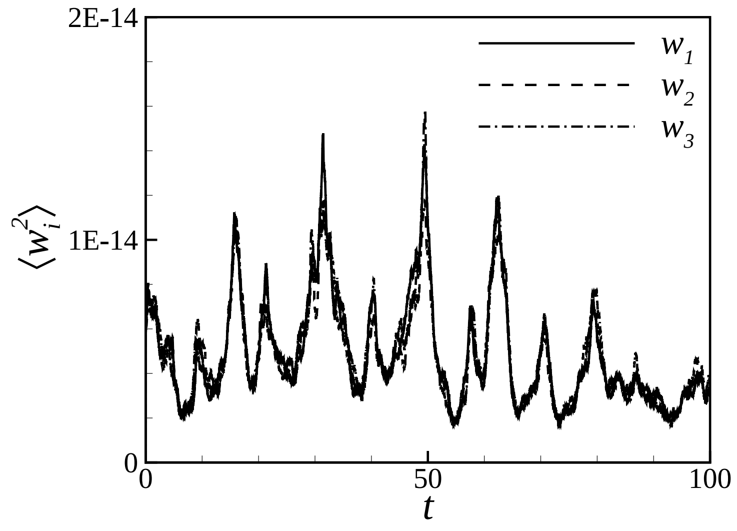

Figure 2.3: Time history of mean-square relative velocities for $Re_\lambda = 58$.

velocity is solved instead of the particle velocity. The evolution of relative velocity in the absence of gravity is given by

$$\frac{dW_i e^{t/\tau}}{dt} = -e^{t/\tau}\frac{du_i}{dt}, \tag{2.12}$$

in which W_i is the relative velocity between a particle and fluid ($\equiv V_i - u_i$). Discretizing equation (2.12) employing RK3 yields,

$$A_f^{p,n} = \frac{u_i^{n+1} - u_i^n}{(a_n + b_n)dt}, \tag{2.13}$$

$$W_i^{n+1} = (W_i^n - a_n dt A_f^{p,n})e^{-(a_n+b_n)dt/\tau} - b_n dt A_f^{p,n-1} e^{-(a_n+b_n+a_{n-1}+b_{n-1})dt/\tau}. \tag{2.14}$$

A_f^P denotes the changing rate of fluid velocity following the particle trajectory. For a simple ideal flow - the fluid velocity is proportional to the location of a particle, $u_i = \alpha X_i$ - it is found that equation (2.14) is numerically stable even when τ is about 100 times smaller than dt.

2.3.2 Results

To assess the particle-tracking scheme, an extreme case, $\tau_p = dt/35$, was simulated. Figure 2.3 shows the mean-square relative velocity with time. The definition of

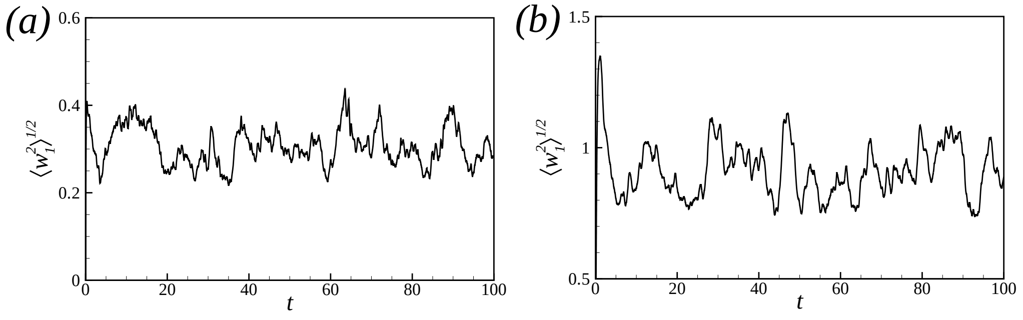

Figure 2.4: Time histories of root-mean-square relative velocities for $Re_\lambda = 58$: (a) $\tau_p = \tau_\eta$, (b) $\tau_p = 10\tau_\eta$.

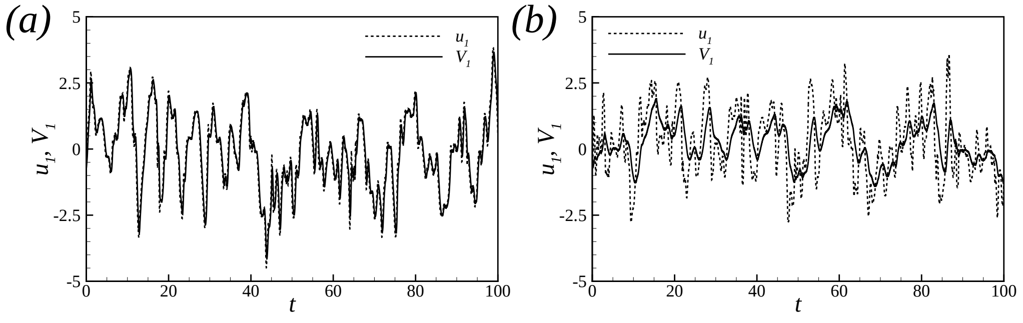

Figure 2.5: Variations of fluid and particle velocities at the location of a particle: (a) $\tau_p = \tau_\eta$, (b) $\tau_p = 10\tau_\eta$.

mean-square relative velocity is

$$\langle W_i^2(t) \rangle = \frac{1}{N_p} \sum_{i=1}^{N_p} (u_i(t, \boldsymbol{X}_0) - V_i(t, \boldsymbol{X}_0))^2, \tag{2.15}$$

in which N_p denotes the number of particles. Since the particle relaxation time is very small, 1000 time smaller than τ_η, the particle almost exactly follows the motion of fluid element. Although it is not shown, the Lagrangian velocity auto-correlation of the particle is also the same with that of fluid element. In this test, it is found that calculating velocities of particles from the relative velocities is numerically stable and gives reliable results even for very small relaxation time.

Figure 2.4 displays the root-mean-square (r.m.s.) relative velocity for two cases; case 1 is $\tau_p = \tau_\eta$ and case 2 is $\tau_p = 10\tau_\eta$. It is shown that the r.m.s. W_i of case 2 is much larger than the other one. Nevertheless the initial conditions for the two cases are the same, an initial sharp increase of relative velocity is observed for case 2. Since the characteristics timescale of case 1 is comparable to the smallest timescale for fluid, the particles follows the motion of fluid closely and, thus, the initial transient state of the r.m.s. relative velocity seems not observed. Figure 2.5 shows the variations of u_i and V_i following the motion of particles. As mentioned, the particle velocity follows the fluid velocity quickly for case 1, while there is a serious time lag for a particle velocity to catch up the fluid velocity for case 2.

Chapter 3

Implementation

3.1 Naming conventions

Most variable names in the **Iso_par** is determined using the following rules.

1. nx, ny, nz are the maximum wavenumbers and nxp, nyp, nzp represent the 3/2 times expanded grid points for de-aliasing.

2. In the case of complex variables, the last character r and i denote real and complex parts, respectively.

3. The variables start with u, H, and om are storages for velocities, nonlinear terms, and vorticities, respectively.

4. The last loop index represents the axis, that is $ur(x, y, z, 1)$ is the real part of the Fourier coefficient of velocity in x direction. However, the last loop indices for the particle and fluid velocities indicate the Runge-Kutta substep.

5. Digits in the middle of temporal variables indicate a two-dimensional plane in which the variables are used. For example, $u3r$ contains the real part of velocity Fourier coefficient in $x - z$ plane and $om2r$ is the real part of vorticity Fourier coefficient in $x - y$ plane.

6. loc and vel mean the storages for location and velocity of particles, respectively, and the first character $p\_$ and $f\_$ are, respectively, the variables for inertial and fluid particles.

7. $alpha, eta, beta$ represent the wavenumbers in x, y, z directions, respectively

8. Variables starting with pre are what used in FFT and should not be changed during calculation, while $w2r, w2i, w3r, w3i$ are temporary storages for FFT.

9. avg means the storage for averaged quantities and amp is the storage for Eulerian statistics.

3.2 Program structure

The program **Iso_par** is, roughly, composed of 3 steps. First, the stationary turbulent field is obtained from input file and simulations are initialized. Second, time is advanced by subroutines **nonlin_1st**, **nonlin_2nd**, and **nonlin_3rd**. Lastly, turbulence statistics are obtained and the computed turbulent field is saved.

3.2.1 Step 1

ppar prints out information about grid size given in **par.f**.

rparam reads the file **iso.i** which contains control information and the maximum cfl number is defined in this subroutine.

prepr initializes variables for **VECFFT** and define wavenumbers in each direction.

getcoeff defines the number of forced modes for each threads. This subroutine is necessary only when using OpenMP.

rdisc reads the input file which contains velocities, stochastic processes for isotropic forcing, and the information about particles. When the grid size in the input file is not identical with that in **par.f**, grid expansion or contraction is performed. In this case, to remove the effects of transient flow, you should run the simulation sufficiently long time before obtaining statistics.
When variable time step, **getdt** calculates cfl number and adjusts time step based on the maximum cfl number defined in **rparam**.

ctim computes CPU time.

3.2.2 Step 2

nonlin_1st saves $\partial u/\partial y$ and $\partial w/\partial y$ used to evaluate vorticities and calculates

$$\hat{u}_i'^n = \hat{u}_i^n e^{-\nu k^2(a_n+b_n)dt} + b_n dt \widehat{NL}_i^{n-1} e^{-\nu k^2(a_n+b_n+a_{n-1}+b_{n-1})dt}.$$

Then, the subroutine performs backward FFT in y-direction.

nonlin_2nd performs backward FFTs in x- and z-directions. The subroutine also computes the nonlinear terms and obtain fluid velocities at the location of particles using Hermite interpolation. Then, forward FFTs of the computed nonlinear terms in x- and z-directions are performed.

nonlin3rd performs forward FFTs in y-direction. Applying stationary forcing, the velocities at $n+1$ are computed,

$$\begin{aligned}\widehat{NL}_i^n &= -\frac{1}{k^2}k_i k_j \hat{H}_j^n + \hat{H}_i^n + f_i^n, \\ \hat{u}_i^{n+1} &= a_n dt \widehat{NL}_i^n e^{-\nu k^2 (a_n+b_n)dt} + \hat{u}_i'^n.\end{aligned}$$

Among forced modes, conjugate pairs are artificially adjusted.

G_forcing computes the Uhlenbeck-Ornstien processes for the next time step.

tracking calculates particle velocities at the present time step and the location of particles at the next time step.

3.2.3 Step 3

getstat calculates Eulerian statistics such as ϵ, u^2, and $\int E(k)/k \; dk$.

loc_stat and **vel_stat** calculates, respectively, Lagrangian statistics of position and velocities of particle and fluid element.

loc_ini and **vel_ini** set the location and velocities of particles at the present time step as an initial condition to calculate Lagrangian statistics.

wdisc writes velocities, external forces, and velocities and position of particles.

3.3 Miscellaneousness

par.f contains information about the size of simulation.

iso.i contains control parameter such as input and output file name, maximum number of iteration, time step, and kinematic viscosity.

getxyb gets variables on $x-y$ plane, performing zero-padding for dealiasing. This subroutine should be used before backward FFT in y direction.

getxyf gets variables on $x-y$ plane. Zero-padding is not performed in this subroutine. This subroutine is typically used before forward FFT in y direction.

getxz gets variables on $x-z$ plane, performing zero-padding. This subroutine is used before two-dimensional FFT in $x-z$ plane.

putxyb puts variables on $x - y$ plane in temporary storages into main storages without performing dealiasing.

putxyf puts variables on $x - y$ plane in temporary storage into main storages performing dealiasing.

putxz puts variables on $x - z$ plane in temporary storage into main storages performing dealiasing.

chkdiv checks whether the velocity field satisfies the divergence-free condition or not.

Hermite calculates the fluid velocities at the location of a particle using 4th-order central differencing scheme.

Variable *s_id* gives an i.d. to the currently initialized Lagrangian statistics set.

Bibliography

[1] A. Lundbladh *et al.* , *An efficient spectral method for simulation of incompressible flow over a flat plate*, Tech. Rep. 1999:11, Royal Institute of Technology, Stockholm (1999).

[2] V. Eswaran & S. B. Pope, "An examination of forcing in direct numerical simulations of turbulence," *Comp. Fluids*, **16**, 257 (1988).

[3] S. B. Pope, *Turbulent Flows* (Cambridge Univ. Press, Cambridge, 2000).

[4] B. K. Oksendal, *Stochastic differential equations: an introduction with applications*, 5th Edition (Springer, New York, 1998).

[5] I. Iliopoulos & T. J. Hanratty, "Turbulent dispersion in a non-homogeneous field," *J. Fluid Mech.*, **392**, 45 (1999).

[6] C. Lee, B. Kim, & N. Kim, "A simple Lagrangian pdf model for wall-bounded turbulent flows," *KSME Int. J.*, **14**, 900 (2000).

[7] P. K. Yeung & S. B. Pope, "Lagrangian statistics from direct numerical simulations of isotropic turbulence," *J. Fluid Mech.*, **207**, 531 (1989).

[8] S. Balachandar & M. R. Maxey, "Methods for evaluating fluid velocities in spectral simulations of turbulence," *J. Comp. Phys.*, **83**, 96 (1989).

[9] J.-I. Choi, K. Yeo & C. Lee, "Lagrangian statistics in turbulent channel flow," *Phys. Fluids*, **16**, 779 (2004).

[10] M. R. Maxey & J. J. Riley, "Equation of motion for a small rigid shpere in a nonuniform flow," *Phys. Fluids*, **26**, 883 (1983).

[11] C. Crowe, M. Sommerfeld & Y. Tsuji, *Multiphase flows with droplets and particles* (CRC Press, U.S., 1998).

[12] K. D. Squires & J. K. Eaton, "Measurements of particles dispersion obtained from direct numerical simulations of isotropic turbulence," *J. Fluid Mech.*, **226**, 1 (1991).

[13] S. Elgobashi & G. C. Truesdell, "On the two-way interaction between homogeneous turbulence and dispersed solid particles. I: Turbulence modification," *Phy. Fluids* A, **5**, 1790 (1993).

[14] M. Soltani & G. Ahmadi, "Direct numerical simulation of particle entrainment in turbulent channel flow," *Phys. Fluids*, **7**, 647 (1995).