A particle-simulation method to study mixing efficiencies

Takahide Okabe

March 15, 2007

1 Introduction

Mixing by fluid flows is a ubiquitous natural phenomenon that plays a central role in many of the applied sciences and engineering. A geophysical example is the mixing of aerosols (e.g., CO_2 supplied by a volcano, say, or by human activity) in the atmosphere. Aerosols are dispersed by molecular diffusion on the smallest scales but are more effectively spread globally by atmospheric flows. The density—and density fluctuations—of some aerosols influence the albedo of the earth and thus have an environmental impact. Hence it is important to understand fundamental properties of dispersion, mixing, and the suppression of concentration fluctuations by stirring flow fields.

At the most basic level, the mixing of a passive scalar can be modeled by an advection-diffusion equation for the scalar concentration field with a specified stirring flow field. In this work we will focus on problems where fluctuations in the scalar field are generated and sustained by temporally steady but spatially inhomogeneous sources. The question of interest here is this: for a given source distribution, how well can a specified stirring flow mix the scalar field? Mixing can be measured by the scalar variance over the domain. A well-mixed scalar field will have a relatively uniform density with "small" variance while increased fluctuations in the scalar density will be reflected in a "larger" variance. We put quotes around the quantifiers small and large because the variance is a dimensional quantity whose magnitude depend on the choice of units employed. A dimensionless measure of the scalar fluctuations is necessary to give precise meaning to these characterizations.

Several years ago Thiffeault *et al* [1] introduced the notion of "mixing efficiency" for a velocity field stirring a steadily sustained scalar by comparing the bulk (space-time) averaged density variance with and without advecting flow. In the absence of stirring the mixing is accomplished by molecular diffusion alone, which can be very effective on small scales but is not generally very good at breaking up and disbursing large scale fluctuations quickly. Stirring can greatly enhance the transport of the scalar from regions of excess density to regions of depletion, however, suppressing the variance far below its diffusion-only value. The magnitude of this variance suppression by the stirring, i.e., the ratio of the variance without stirring to the variance in the presence of stirring, is a dimensionless quantity that provides a sensible gauge of the mixing efficiency of the flow. Different advection fields will have different efficiencies stirring scalars supplied by different sources.

It is then of obvious interest both to determine theoretical limits on mixing efficiencies for various source configurations and to explore whether those limits may be achieved.

In this project we develop a computational scheme that is easy to implement and applicable to the study of mixing by any advection field and with any source distribution. The idea is to develop a method that accurately simulates advection and diffusion of large numbers of passively advecting particles introduced by a steady source, and to measure density fluctuations by "binning" the particles to produce an approximation of the "hydrodynamic" concentration field. Unlike a numerical PDE code, a particle code does not prefer specific forms of advection or source (PDE methods generally work best with smooth fields). There is, however, no free lunch: the accuracy of the particle code is ultimately limited by the finite number of particles that can be tracked. The limitation to finite numbers of particles inevitably introduces statistical errors due to discrete fluctuations in the local density and systematic errors in the variance measurements due to binning. But these problems can be addressed and as will be shown in this report, for some applications the method proves to be computationally efficient and quantitatively accurate.

The most significant upside of a particle code—and one of the most significant motivations for this work—is that it can easily handle (i.e., resolve) small scales in sources and, subsequently, in the concentration field. It is even applicable to delta-function sources whose resolution requirements would strain standard PDE methods. Delta-function scalar sources are the most singular physically relevant distributions, and at the same time the simplest to implement in a particle tracking scheme: just introduce the particles at the same point in space. A delta-function source could serve, for example, as a model of a smoke stack supplying an aerosol into the atmosphere when the smallest scales in the flow are larger than the radius of the outlet.

The remainder of this report is organized as follows. In section 2 the mathematical model is presented, basic quantities characterizing mixing phenomena are defined, and some general results about mixing efficiencies are reviewed. In section 3 the particle simulation scheme is explained in detail. The problems inherent to a discrete particle method, and solutions to these problems, are also discussed. The particle code as implemented is numerically validated in section 4. There, the variance from the particle code is compared with exact solutions and the results of a PDE code for some benchmark problems. In section 5 the particle method is used to measure the mixing efficiency of a particular statistically homogeneous flow stirring ever smaller-scale sources (down to a delta-function source). This is a new result, and it is qualitatively and quantitatively compared to previous analysis of upper bounds on the mixing efficiency for such sources. We close this report with conclusive remarks and provide appendices containing details of the computer code used to implement the scheme.

2 Basic facts about the mixing efficiency problem

In this section we review basic facts about the mixing efficiency problem as formulated by Thiffeault, Doering & Gibbon *el al* [1] and further developed by Plasting & Young [2], Doering & Thiffeault [3], and Shaw *et al* [4]. The dynamics is given by the advection-diffusion equation for the concentration of a passive scalar $\rho(t, \mathbf{x})$ with time-independent

but spatially inhomogeneous source field $S(\mathbf{x})$:

$$\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = \kappa \Delta \rho + S(\mathbf{x}), \tag{1}$$

where κ is the molecular diffusivity and $\mathbf{u}(t, \mathbf{x})$ is a specified advection field that satisfies (at each instant of time) the incompressibility condition

$$\nabla \cdot \mathbf{u} = 0. \tag{2}$$

For simplicity, the domain is the d-torus, i.e., $[0, L]^d$ with periodic boundary conditions. We will limit attention in this report to stirring fields that satisfy the properties of statistical homogeneity and isotropy defined by

$$\overline{u_i(\mathbf{x}, \cdot)} = 0 \tag{3}$$

$$\overline{u_i(\mathbf{x}, \cdot) u_j(\mathbf{x}, \cdot)} = \frac{U^2}{d} \delta_{ij} \tag{4}$$

where the overbar is time average and U is the root mean square speed of the velocity field, a natural indicator of the intensity of the stirring (recall that d is the spatial dimension). These statistical properties are shared by homogeneous isotropic turbulence on the torus.

We are interested in fluctuations in the concentration ρ so the spatially averaged background density is irrelevant. It is easy to see from (1) that the spatial average of ρ grows linearly with time at the rate given by the spatial average of S. Hence we may change variables to spatially mean-zero quantities

$$\theta(t, \mathbf{x}) = \rho(t, \mathbf{x}) - \frac{1}{L^d} \int d^d x' \, \rho(t, \mathbf{x}') \tag{5}$$

and

$$s(\mathbf{x}) = S(\mathbf{x}) - \frac{1}{L^d} \int d^d x' \, S(\mathbf{x}') \tag{6}$$

that satisfy

$$\frac{\partial \theta}{\partial t} + \mathbf{u} \cdot \nabla \theta = \kappa \Delta \theta + s(\mathbf{x}). \tag{7}$$

(We must also supply initial conditions for ρ and/or θ but they play no role in the long-time statistically steady statistics that we are interested in.)

The "mixedness" of the scalar may be characterized by, among other quantities, the long-time averaged variance of ρ, proportional to the long-time averaged L^2 norm of θ,

$$\langle \theta^2 \rangle := \lim_{T \to \infty} \frac{1}{T} \int_0^T dt \, \frac{1}{L^d} \int d^d x \, \theta^2(t, \mathbf{x}) \tag{8}$$

The smaller $\langle \theta^2 \rangle$ is, the more uniform the distribution is. The "mixing efficiency" of a stirring field is naturally evaluated by comparing the scalar variance to the variance with

the same source but in the absence of stirring. To be perfectly precise, we compare $\langle\theta^2\rangle$ to $\langle{\theta_0}^2\rangle$ where θ_0 is the solution to

$$\frac{\partial\theta_0}{\partial t} = \kappa\Delta\theta_0 + s(\mathbf{x}) \tag{9}$$

(with, say, the same initial data although these will not affect the long-time averaged fluctuations). Formally, then, the dimensionless *mixing efficiency* is defined

$$\mathcal{E}_0 := \sqrt{\frac{\langle\theta_0^2\rangle}{\langle\theta^2\rangle}}. \tag{10}$$

This efficiency carries the subscript 0 because we can also define *multiscale mixing efficiencies* by weighting large/small wavenumber components of the scalar fluctuations:

$$\mathcal{E}_p := \sqrt{\frac{\langle|\nabla^p\theta_0|^2\rangle}{\langle|\nabla^p\theta|^2\rangle}} \quad (p = -1, 0, 1). \tag{11}$$

As discussed in Doering & Thiffeault [3], Shaw *et al* [4] and Shaw [5], $\mathcal{E}_{\pm1}$ provide a gauge of the mixing efficiencies of the flow as measured by scalar fluctuations on relatively small and large length scales respectively. In this project, however, we will focus exclusively on the mixing efficiency at "moderate" length scales, $\mathcal{E}_0$.

There is a theoretical upper bound on $\mathcal{E}_0$ valid for any statistically stationary homogeneous and isotropic stirring field [3, 5, 4]:

$$\mathcal{E}_0 \le \sqrt{\frac{\sum_{\mathbf{k}\neq\mathbf{0}} |\hat{s}(\mathbf{k})|^2/k^4}{\sum_{\mathbf{k}\neq\mathbf{0}} |\hat{s}(\mathbf{k})|^2/(k^4 + \frac{\mathrm{Pe}^2}{L^2 d}k^2)}} \tag{12}$$

where $\hat{s}(\mathbf{k})$ are the Fourier coefficients of the source and the P*é*clet number

$$\mathrm{Pe} := \frac{UL}{\kappa}. \tag{13}$$

is a dimensionless measure of the intensity of the stirring. Generally we anticipate that $\mathcal{E}_0$ is an increasing function of Pe and the estimate in (12) guarantees that $\mathcal{E}_0(\mathrm{Pe}) \lesssim \mathrm{Pe}$ as $\mathrm{Pe} \to \infty$, the scaling expected if there is any residual variance suppression in the singular vanishing diffusion limit (i.e., $\kappa \to 0$ with all other parameters held fixed).

The upper limit to the mixing efficiency in (12) depends on the stirring field only through U via Pe, but it depends on all the details of the source distribution. As studied in depth in references [3, 4, 5], the structure of the scalar source can have profound effects on the behavior, i.e., the high Pe scaling, of $\mathcal{E}_0$. It is precisely this source-dependence of the qualitative behavior of $\mathcal{E}_0(\mathrm{Pe})$ that motivates this development of a computational method that can handle singular source distributions.

In the remainder of this report we focus on the two-dimensional torus ($d = 2$) and for computational simplicity and efficiency we take as the stirring field the "random sine flow" defined for all time by

$$\mathbf{u}(t,\mathbf{x}) = \begin{cases} w\sin(\frac{2\pi y}{L} + \phi_n)\hat{\mathbf{i}} & \text{for} \quad nT < t \le nT + T/2 \\ w\sin(\frac{2\pi x}{L} + \phi'_n)\hat{\mathbf{j}} & \text{for} \quad nT + T/2 < t \le (n+1)T \end{cases} \tag{14}$$

where T is the period, $n = 0, 1, 2, \ldots$, and ϕ_n and ϕ'_n are random phases chosen independently and uniformly on $[0, 2\pi)$ in each half cycle. Then $U = w/\sqrt{2}$.

3 A particle code

In a particle code to solve the advection-diffusion equation, the concentration field ρ is represented by a distribution of particles. Particles introduced by generating random locations using the properly normalized source $S(\mathbf{x})$ as a probability distribution function, and advecting them with the flow. Given a particle distribution, $\rho(t, \mathbf{x})$ is measured by covering the domain with bins counting the number of particles per bin.

A particle code is employed because it can deal with a small-scale source. It is easily applicable for any source fields and advection fields, while the spectral method prefers fields whose Fourier expansion is simple. The downside of a particle code is that it necessarily introduces statistical errors: the number density of particles calculated by dividing the domain into bins is only resolved down to the lengthscale of bin size, and the measurement of ρ always includes error due to the use of finite number of particles.

In this section the numerical scheme based on a particle code is explained. The code mainly consists of three parts: 1) Time evolution, 2) calculation of variance, and 3) a particle subtraction scheme. The time evolution is realized by displacing each particle with appropriate advection and diffusion, and by adding new particles in accordance with a source term. We calculate spatial variance at a random instant once each half cycle in order to take its time average. A subtraction scheme removes group of particles that are well-mixed and this not participating in time evolution any longer. The subtraction scheme is necessary and crucial to prevent a calculation from slowing down due to an ever increasing number of particles in the system. Details of the code are presented in the appendix.

3.1 Variance calculation

The variance $\langle\theta^2\rangle$ is measured by monitoring the fluctuations in the number of particles per bin and time averaging. In two dimensions the domain is divided into l^2 bins and the code calculates $\langle n^2\rangle$ where n is the number of particles in a bin and $\langle\theta^2\rangle$ is initially approximated by

$$\langle n^2\rangle - \langle n\rangle^2 = \left(\frac{L}{l}\right)^2 \langle\theta^2\rangle. \tag{15}$$

We say "initially" because the expression above includes both the hydrodynamic fluctuations of interest *and* discreteness fluctuations resulting solely from the fact that each bin contains a finite number of particles. We will discuss corrections to this expression for the variance to account for this effect below in Section 3.3. Beyond these inevitable fluctuations due to discreteness, because of the binning density fluctuations are observed only down to the length scales $\sim \frac{L}{l}$, which is one of the sources of error in this procedure.

The variance is calculated once per each half period, and the instant when it is calculated is determined randomly in order to obtain an unbiased time average. Thus each half period is divided into two parts, before and after variance calculation, and the particle transport and source processes are appropriately adapted.

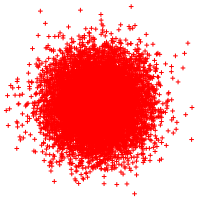

Figure 1: Gaussian due to diffusion only.

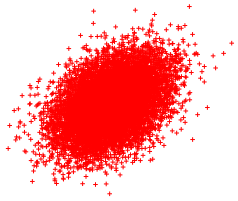

Figure 2: Distorted Gaussian due to shear.

3.2 Time evolution: particle transport

At each time step, the system is evolved by advection and diffusion, and by the source terms. First we focus on particle motion, and then on the particle input.

An advection-only equation would be solved by evolving particles along characteristics, and a diffusion-only equation would be solved by adding Gaussian random noise to each particle. With both advection and diffusion we need to solve a stochastic differential equations to determine the proper displacement of the particles during a time step. The stochastic differential equation is

$$d\mathbf{X} = \mathbf{u}(t, \mathbf{X})dt + \sqrt{2\kappa}\, d\mathbf{W} \tag{16}$$

where $\mathbf{W}(t)$ is a standard vector-valued Wiener process.

In order to solve (16) we will assume that the displacement due to the noise in a half-period $T/2$ is much smaller than the wavelength of the random sine flow. Then during each half period the drift field $\mathbf{u}(t, \mathbf{X})$ experienced by each particle can be approximated by a constant flow with a linear shear superposed. For the first half of the period for a particle starting at $(x_0, y_0) = (X(t = 0), Y(t = 0))$ we approximate (16) by

$$\begin{cases} dX = w \sin(\frac{2\pi y_0}{L} + \phi)dt + w \cos(\frac{2\pi y_0}{L} + \phi)\frac{2\pi}{L}(Y - y_0)dt + \sqrt{2\kappa}\, dW_1 \\ dY = \sqrt{2\kappa}\, dW_2 \end{cases} \tag{17}$$

and for the second half of the period, starting from $(x_0', y_0') = (X(t = T/2), Y(t = T/2))$,

$$\begin{cases} dX = \sqrt{2\kappa}\, dW_1 \\ dY = w \sin(\frac{2\pi x_0'}{L} + \phi')dt + w \cos(\frac{2\pi x_0'}{L} + \phi')\frac{2\pi}{L}(X - x_0')dt + \sqrt{2\kappa}\, dW_2. \end{cases} \tag{18}$$

Therefore, during the first half period we evolve the position of a particle through a time interval Δt (where $\Delta t \le T/2$ need *not* be small) by the map

$$x \;\rightarrow\; x_0 + w \sin(\frac{2\pi y_0}{L} + \phi)\, \Delta t + \sqrt{\frac{1}{6}S^2\kappa\Delta t^3 + 2\kappa\Delta t} \times N_1 + \sqrt{\frac{1}{2}S^2\kappa\Delta t^3} \times N_2 \tag{19}$$

$$y \;\rightarrow\; y_0 + \sqrt{2\kappa\Delta t} \times N_2 \tag{20}$$

where N_1 and N_2 are independent $N(0,1)$ random variables. A similar map is employed during the second half of the period. These stochastic maps include the shear—in the approximation that the shear remains constant for each particle during each half cycle—that causes a "distortion" of a Gaussian cloud of particles; see Figures 1 and 2.

3.3 Time evolution: particle input

The steady scalar source is realized by introducing new particle one by one using normalized $S(\mathbf{x})$ as a probability distribution function. Numerically, such probability distribution function can be realized by mapping uniform random numbers with an inverse of cumulative probability distribution function in question. In Figures 3, 4 and 5, sample source terms are visualized by putting many (in these examples 10^4) particles at once. The monochromatic source in Figure 3 is $S(\mathbf{x}) = A[1 + \sin(2\pi(x+y)/L)]$.

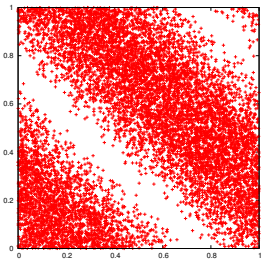

Figure 3: Monochromatic

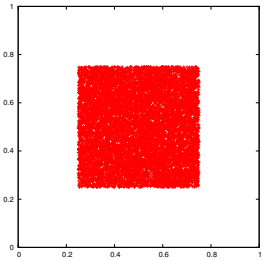

Figure 4: Square ($a = \frac{L}{2}$)

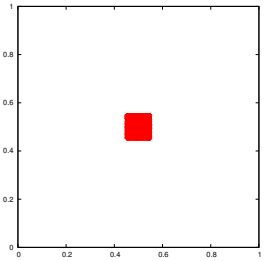

Figure 5: Square ($a = \frac{L}{10}$)

In the actual time evolution of the system, however, new particles are added one by one. Since new particles are put in constantly, the total number of particles increases which makes computation slow down. To cope with increasing particles, we will implement a particle subtraction scheme as described in the next section.

3.4 Background noise and subtraction scheme

Particles eventually get well mixed, and the "older" particles do not contribute the value of the hydrodynamic variance. There is no added value in keeping track of those particles, and we can simply remove them from the system after a sufficiently long time. In fact it is necessary to implement such a particle subtraction scheme so that computation goes on without slowing down.

A subtraction scheme eliminates particles which are "well-mixed", but we need to be careful about the well-mixedness in a particle code. If the system is completely mixed, the hydrodynamic variance $\langle\theta^2\rangle = 0$. But since $\theta(t,\mathbf{x})$ is represented with a finite number of particles and bin of a finite size, $\langle\theta^2\rangle_{\text{measured}}$ is nonzero even when the particles are uniformly distributed: $\langle n^2\rangle$ has the same amount of fluctuations as the error we might have when N_{all} particles are randomly thrown in l^2 bins. Thus when the particles are uniformly distributed $\langle n^2\rangle - \langle n\rangle^2$ is the order of N_{all}/l^2 as illustrated in Figures 6 and 7. There $\overline{\theta^2}_{\text{measured}}(t)$—where the overline now represents the volume average—is plotted in the diffusion-only case with the monochromatic initial condition. Instead of approaching 0, $\overline{\theta^2}_{\text{measured}}(t)$ goes to N_{all}/l^2. We call this departure from 0 a *background fluctuation*, and we refer to the error

due to the use of the finite number of particles and finite size of a bin as error due to *discreteness*.

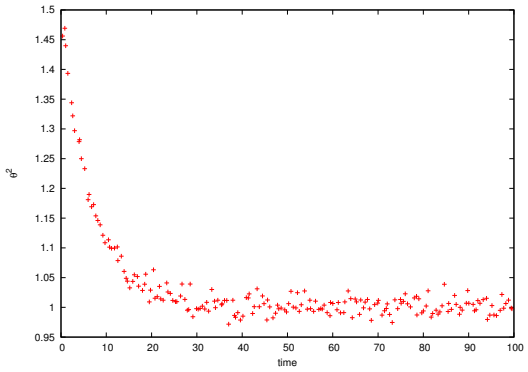

Figure 6: $\kappa = 10^{-3}, N_{\text{all}} = 10^4$, 10^4 bins

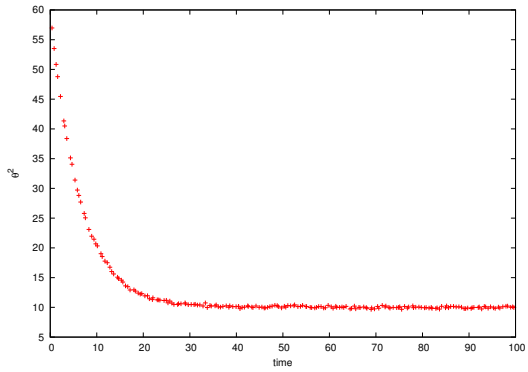

Figure 7: $\kappa = 10^{-3}, N_{\text{all}} = 10^5$, 10^4 bins

These background fluctuations must be removed to obtain hydrodynamic variance that we are interested in. The effect of this subtraction is illustrated in Figures 8 and 9. When the initial density is $\rho(0, \mathbf{x}) = A[1 + \sin(2\pi(x + y)/L)]$,

$$\rho(t, \mathbf{x}) = A\left[1 + e^{-\frac{8\pi^2}{L^2}\kappa t} \sin\left(\frac{2\pi}{L}(x + y)\right)\right], \tag{21}$$

the instantaneous hydrodynamic variance, i.e., $\overline{\theta^2} = \overline{(\rho - \overline{\rho})^2}$, is

$$\overline{\theta^2}(t) = \frac{A^2}{2} e^{-\frac{16\pi^2}{L^2}\kappa t}. \tag{22}$$

As illustrated in Figure 9, after background fluctuations are subtracted off we obtain the correct behavior, i.e., exponential decay. In Figure 8 it might be difficult to tell the difference, but in the log-linear plot in Figure 9 it is obvious that θ^2 shows exponential decay only after background noise is removed. From this point on, this background noise is always removed when $\langle \theta^2 \rangle$ is calculated.

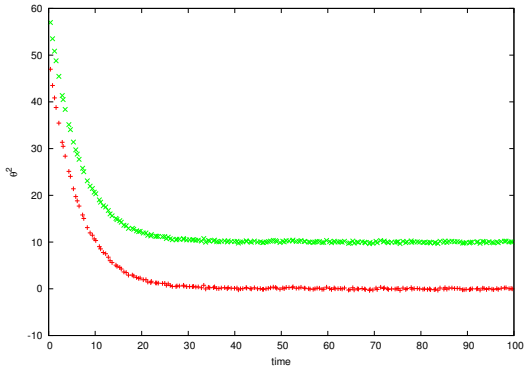

Figure 8: Normal plots

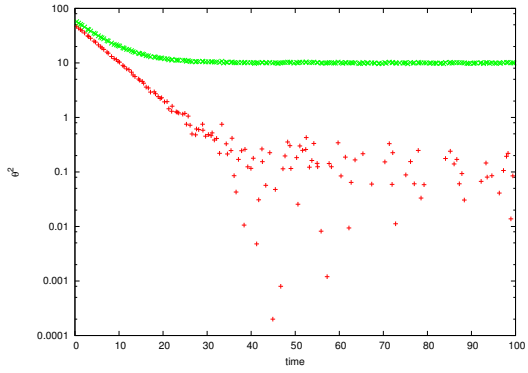

Figure 9: Log-normal plots

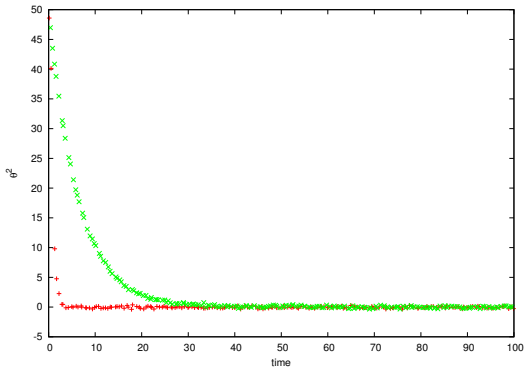

Figure 10: Normal plots

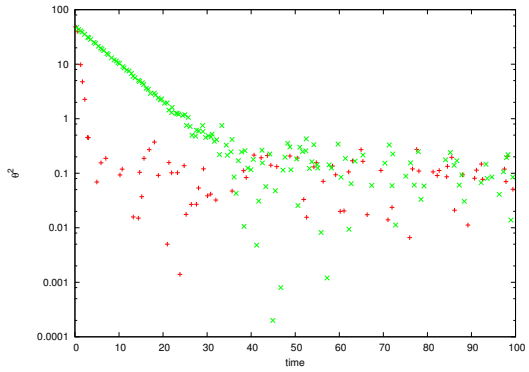

Figure 11: Log-normal plot.

A group of particles can be discounted, removed from further consideration, once their variance reaches the level of background fluctuation. In the actual code, we keep track the Δn particles put in a half period as a group and they are removed after they participate in the time evolution for a time t_*, where t_* is a typical time scale for Δn particles to reach the background fluctuation level. The "lifetime" t_* depends on the number of particles in the cohort we are regarding as "old", the form of the source field, the advection field, and of course the diffusion coefficient κ. This lifetime is estimated by plotting a transient behavior of the variance of Δn particles under the particular advection and/or diffusion conditions of interest. In Figures 10 and 11, 10^5 particles, initially distributed as for the monocromatic source, are mixed by diffusion or advection-diffusion. Eventually both cases end up in a well-mixed state (in the figures, background fluctuations are already removed) and t_* is estimated to be $\sim$10 in the case of advection-diffusion and $\sim$45 in the case of only diffusion. These values of t_* would then be used for long time average measurements for these particular source, stirring and diffusion conditions. For each source, stirring or amplitude of diffusion, such a transient calibration simulation must be repeated to determine the appropriate value of t_* .

3.5 Benchmark Tests

In this section we report the results of specific simulation where the particle code results can be compared to either exact solutions or numerical solutions of the inhomogeneous advection-diffusion partial differential equation. These benchmark tests serve as a check of the code and give some quantitative information about our particle tracking scheme's accuracy.

3.5.1 Simulation parameter independence

First of all, the measures of hydrodynamic variances should be independent of Δn (the number of particles introduced each half-cycle of the stirring) or the bin size that is used to estimate $\rho(t, \mathbf{x})$. The parameter independence can be checked by changing the value of Δn or l with other conditions fixed. Here, we illustrate Δn independence by showing the exponential decays of transient variance for several values of Δn. κ is 0.01, 0.001,

0.0001, respectively, and $\Delta n = 10^4(\text{red}), 10^5(\text{green})$ and 10^6 (blue) (Figures 12, 13 and 14). The plots show exponential decay until the particles are well-mixed, as expected, and

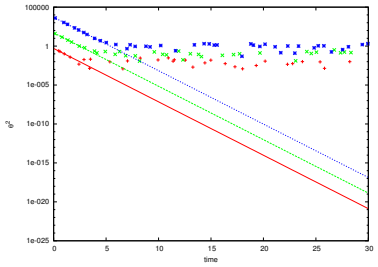

Figure 12: κ=0.01

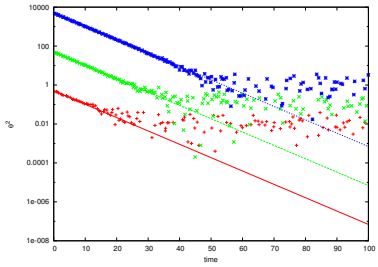

Figure 13: κ=0.001

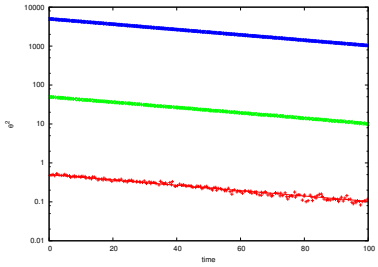

Figure 14: κ=0.0001

the exponential decay rates are independent of Δn (at least in the range $[10^4, 10^6]$ tested). The straight lines fit to the data here are the exact theoretical values with no adjustable parameters.

3.5.2 Diffusion-only with steady source

Secondly we consider diffusion-only with a steady source since $\langle {\theta_0}^2 \rangle$ can be calculated exactly for these cases. It is possible to solve Eq. (9) analytically by Fourier expansion:

$$\theta_0(t, \mathbf{x}) = \sum_{\mathbf{k} \neq \mathbf{0}} \hat{\theta}_0(t, \mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}} \tag{23}$$

$$s(\mathbf{x}) = \sum_{\mathbf{k} \neq \mathbf{0}} \hat{s}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}} \tag{24}$$

Note that $\hat{\theta}(t, \mathbf{k} = \mathbf{0}) = 0$ and that $\hat{s}(t, \mathbf{k} = \mathbf{0}) = 0$, because $\theta(t, \mathbf{x})$ and $s(\mathbf{x})$ have zero spatial means. The inhomogeneous diffusion equation is

$$\frac{\partial \hat{\theta_0}}{\partial t} = -\kappa k^2 \hat{\theta}_0(t, \mathbf{k}) + \hat{s}(\mathbf{k}) \tag{25}$$

so that

$$\hat{\theta}_0(t, \mathbf{k}) = \frac{\hat{s}(\mathbf{k})}{\kappa k^2} + \left(\hat{\theta}_0(0, \mathbf{k}) - \frac{\hat{s}(\mathbf{k})}{\kappa k^2} \right) e^{-\kappa k^2 t} \tag{26}$$

Plugging this expression into the definition of $\langle {\theta_0}^2 \rangle$,

$$\langle {\theta_0}^2 \rangle = \lim_{T\to\infty} \frac{1}{T} \int_0^T dt \, \frac{1}{L^d} \int d^d x \sum_{\mathbf{k},\mathbf{k}' \neq \mathbf{0}} \hat{\theta}_0(t, \mathbf{k}) \hat{\theta}_0(t, \mathbf{k}') e^{i(\mathbf{k}+\mathbf{k}')\cdot\mathbf{x}} \tag{27}$$

$$= \lim_{T\to\infty} \frac{1}{T} \int_0^T dt \sum_{\mathbf{k} \neq \mathbf{0}} |\hat{\theta}_0(t, \mathbf{k})|^2 \tag{28}$$

$$= \frac{1}{\kappa^2} \sum_{\mathbf{k} \neq \mathbf{0}} \frac{|\hat{s}(\mathbf{k})|^2}{k^4} \tag{29}$$

We now check if the particle simulation code produces this value with three kinds of sources: (1) a monochromatic source, (2) a square source and (3) a delta-function source.

For the monochromatic source,

$$\langle {\theta_0}^2 \rangle = \frac{S^2 L^4}{128\pi^4\kappa^2} \tag{30}$$

$$\langle {n_0}^2 \rangle = \langle {\theta_0}^2 \rangle \cdot (\text{bin vol.}) = \frac{(\overline{\frac{\Delta n}{\Delta t}})^2 L^4}{128\pi^4\kappa^2} \tag{31}$$

and the comparison of theory and simulation is presented in tabular form:

κ	$\langle {\theta_0}^2 \rangle_{\text{calculated}}$	theoretical values
0.05	0.1290±0.0076	0.128324
0.02	0.7914±0.0032	0.802029
0.01	3.215±0.010	3.208119
0.005	12.370±0.014	12.8324
0.002	78.836±0.052	80.2029
0.001	312.82±0.13	320.8119
0.0005	1236.39±0.29	1283.24

For the square sources

$$s(\mathbf{x}) = \begin{cases} S & \mathbf{x} \in \left[-\frac{a}{2}+\frac{L}{2}, \frac{a}{2}+\frac{L}{2}\right] \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

the variances are

$$\langle {\theta_0}^2 \rangle = \frac{1}{\kappa^2}\frac{16S^2}{L^4}\sum_{\mathbf{k}\neq\mathbf{0}} \frac{1}{({k_1}^2+{k_2}^2)^2}\left(\frac{\sin(k_1\frac{a}{2})}{k_1}\right)^2\left(\frac{\sin(k_2\frac{a}{2})}{k_2}\right)^2 \tag{33}$$

$$\langle {n_0}^2 \rangle = \frac{1}{\kappa^2}\frac{16}{a^4}\left(\overline{\frac{\Delta n}{\Delta t}}\right)^2\sum_{\mathbf{k}\neq\mathbf{0}} \frac{1}{({k_1}^2+{k_2}^2)^2}\left(\frac{\sin(k_1\frac{a}{2})}{k_1}\right)^2\left(\frac{\sin(k_2\frac{a}{2})}{k_2}\right)^2 \tag{34}$$

and the simulations yield

$a = \frac{L}{2}$

κ	$\langle {\theta_0}^2 \rangle_{\text{calculated}}$	theoretical values
0.01	45.593±0.041	45.862
0.005	177.750±0.092	183.448

$a = \frac{L}{10}$

κ	$\langle {\theta_0}^2 \rangle_{\text{calculated}}$	theoretical values
0.01	141.593±0.075	142.092
0.005	565.99±0.19	568.37

$a = \frac{L}{20}$

κ	$\langle {\theta_0}^2 \rangle_{\text{calculated}}$	theoretical values
0.01	150.502±0.086	150.615
0.005	599.79±0.20	602.46

Finally, for the delta-function source

$$s(\mathbf{x}) = S\delta\left(x - \frac{L}{2}\right)\delta\left(y - \frac{L}{2}\right), \tag{35}$$

we have

$$\langle {\theta_0}^2 \rangle = \frac{1}{\kappa^2}\frac{S^2}{L^4}\sum_{\mathbf{k}\neq\mathbf{0}}\frac{1}{({k_1}^2 + {k_2}^2)^2} \tag{36}$$

$$\langle {n_0}^2 \rangle = \frac{1}{\kappa^2}\left(\overline{\frac{\Delta n}{\Delta t}}\right)^2\sum_{\mathbf{k}\neq\mathbf{0}}\frac{1}{({k_1}^2 + {k_2}^2)^2} = \frac{1}{\kappa^2}\left(\overline{\frac{\Delta n}{\Delta t}}\right)^2 L^4 \cdot 3.8669 \times 10^{-3}. \tag{37}$$

(Note that (37) can be obtained by letting $a \to 0$ in (34).) We can check if the code outputs the same value:

κ	l	$\langle {\theta_0}^2 \rangle_{\text{measured}}$	theoretical values
0.02	100	37.772±0.023	38.669
0.01	100	142.673±0.064	154.678
0.02	200	2.4127±0.0025	2.4168
0.01	200	9.4448±0.0044	9.6674
0.02	400	0.15116±0.00024	0.15105
0.01	400	0.59042±0.00034	0.60421
0.005	400	2.36000±0.00085	2.41684

In those results, $\langle {\theta_0}^2 \rangle_{\text{measured}}$ tends to be smaller than the theoretical values because variance calculation is based on bins of a finite size and the contribution from smaller scales is not included. If $\langle {\theta_0}^2 \rangle_{\text{measured}}$ is compared with, say,

$$\frac{1}{\kappa^2}\sum_{\mathbf{k}\neq\mathbf{0}}^{|\mathbf{k}|\leq\frac{2\pi}{\Delta l}}\frac{|\hat{s}(\mathbf{k})|^2}{k^4}, \tag{38}$$

the discrepancies would be smaller. Also, note that in the case of a delta function source, the bin size needs to be very small—at least in the neighborhood of the source—in order to obtain accurate values.

3.5.3 Advection, diffusion and a steady source

Finally, we compare the full advection-diffusion-source code with the results of a spectral method applied to the inhomogeneous advection-diffusion partial differential equation. In Figure 15, the mixing efficiency is plotted against Pe for the case of the monochromatic source stirred by the random sine flow. The green curve shows the theoretical upper bound and the red curve is calculated by spectral method [4, 5]. The blue points are from the particle code. This comparison shows that the code accurately calculates mixing efficiencies and that it can be effectively as accurate as spectral method even with Δn is as small as 10^4.

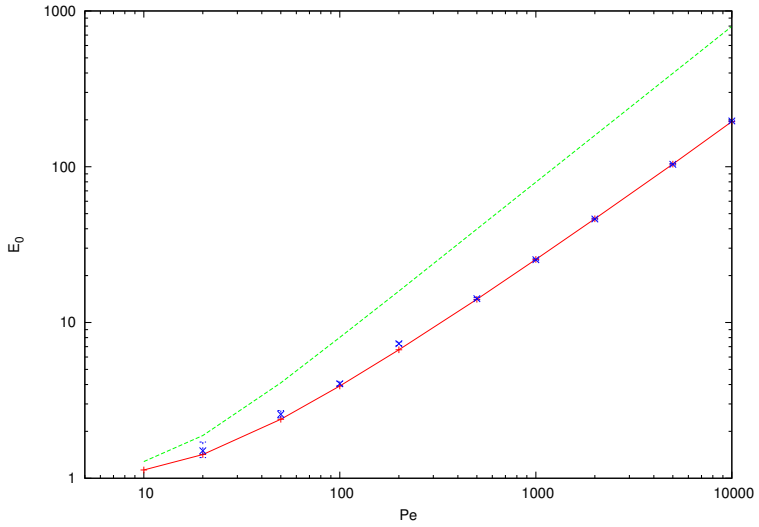

Figure 15: Benchmark test for the mixing efficiecy with a monochromatic source.

4 New applications

The particle code is applicable for small-scale sources, as shown in the diffusion-only case in the previous section. Figures 16, 17 show, repectively, the upper bounds on the mixing efficiencies and the measured values of the mixing efficiencies for square and delta function sources. The theoretical upper bounds and data plots are for source sizes $L/2, L/10, L/50$ and a delta function source (from top to bottom in each plot). The upper bound analysis predicted that as the source gets smaller, the $\mathcal{E}_0(\mathrm{Pe})$ curves are lowered. While the upper bound for any finite-size source is asymptotically $\sim$Pe, the delta function source behaves $\sim \frac{\mathrm{Pe}}{\sqrt{\ln \mathrm{Pe}}}$ in the large Pe limit.

As the source gets smaller, the measured mixing efficiencies get smaller in a way that is qualitatively remarkably similar to that shown by the bounds. That is, Figures 16 and 17 show that the observed mixing efficiencies qualitatively display the same features as that of upper bounds as far as source-size is concerned. The bounds and simulation data are plotted together for comparison in Figure 18.

5 Future Works and Conclusions

We have confirmed that we can use a particle code to study hydrodynamic mixing efficiencies. The particle method reproduces theoretical values and previous numerical simulation correctly. As we saw, the outputs may be as accurate as a PDE code. Moreover, the number of particles used to represent the passive scalar field can be as small as 10^4. The particle method is particularly useful for simulations at high P*é*clet number and with a wide variety, including singular measure-valued source distributions. In this project, the same code was used effectively for a monochromatic source, square sources and a delta-function source. The code efficiently produced reliable results for all these cases.

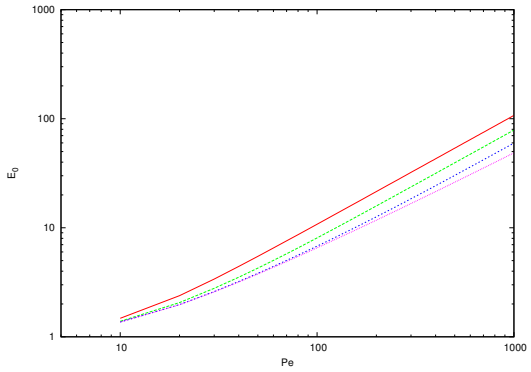

Figure 16: Upper bounds

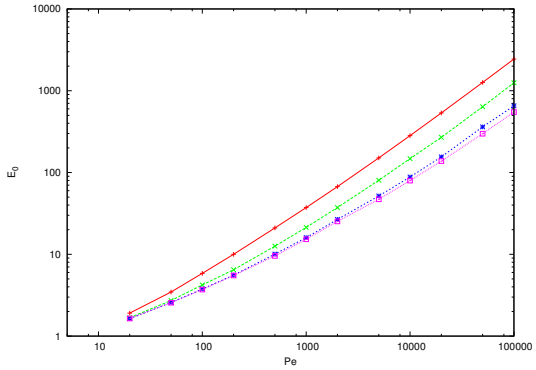

Figure 17: The plot of mixing efficiencies

What we have done here is just the very beginning of investigations exploiting the particle code. There are lots of problems to be explored. Firstly, we can adapt the method to other stirring fields. In this paper only the random sine flow was used, but it is possible to extend this approach to other advection fields such as sine flows with a variety of wave numbers or turbulent flows. Secondly, simulations in three dimensions important. The distinction between mixing efficiencies for the finite-size square sources and a delta function source is predicted to be much more apparent in 3D. The extension of the particle method to 3D is straightforward although simulations will require much more computation power (more particles and more bins will be necessary). Thirdly, the mixing efficiency on large length scales, $\mathcal{E}_{-1}$, can be calculated in principle even though only $\mathcal{E}_0$ was calculated in this project. Fourthly, we would like to see the results of the small-scale sources reported here reproduced by another numerical scheme. Then the results presented in this report can serves as a benchmark test for new codes.

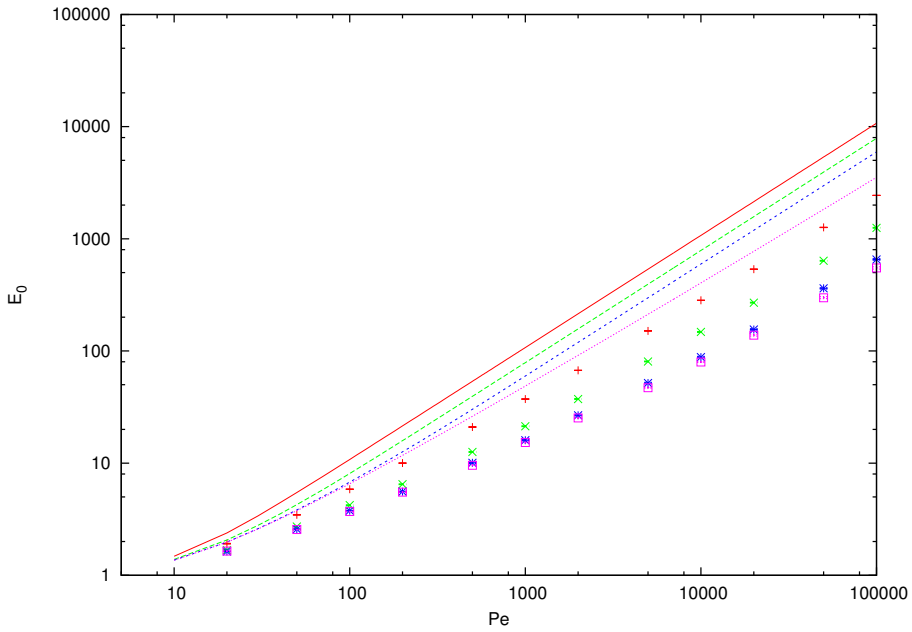

Figure 18: Upper bounds (solid lines) and simulation data (points). The different colors represent the different source sizes.

6 Acknowledgements

The author is grateful for the wonderful summer. I learned much by studying with the intelligent fellow Fellows. The principal lecture by Dr. Worster was full of intellectual excitement even though I knew nothing about the theory of sea ice before. The author sincerely appreciates my advisor, Dr. Doering. He taught me the mixing efficiency problem from scratch, and when I was having difficulties with the code, he helped me by suggesting possible solutions. I could not have accomplished the project without his patient guidance and encouragement. It has been one of my greatest honors to complete research under the supervision of Dr. Doering.

References

[1] J. -L. Thiffeault, C. R. Doering, and J. D. Gibbon, *J. Fluid Mech.* **521** (2004), 105-114.

[2] S. Plasting and W. R. Young, *J. Fluid Mech.* **552** (2006), 289-298.

[3] Charles R. Doering and Jean-Luc Thiffeault, *Phys. Rev. E* Vol. 74, 025301(R) (2006).

[4] T. A. Shaw, J. -L. Thiffeault and C. R. Doering, Stirring up trouble, *Physica D* (submitted, 2006).

[5] T. A. Shaw, Bounds on Multiscale Mixing Efficiencies. *Proc. 2005 Summer Program in Geophysical Fluid Dynamics*. Woods Hole Oceanographic Institution, Woods Hole, MA, USA. http://gfd.whoi.edu/proceedings/2005/PDFvol2005.html.

Appendices

In these appendices we explain the detail of the code, including the actual implementation.

A Calculation of Variance

The domain is divided into `l*l` bins (i.e. bin size is `deltal`=L/l) in order to calculate instantaneous variances. Given positions of all the particles, (`p[n].x` and `p[n].y` (`n`=1, 2, $\cdots$, `N_all`), the number of particles in each bin (`bin[c]`, `c`=1,2, $\cdots$, `l*l`) is counted as follows.

```
for(n=1; n<=N_all; n++){
u=(int)(p[n].x/deltal);
v=(int)(p[n].y/deltal);
bin[v*l+u+1]=bin[v*l+u+1]+1;
}
```

`u` and `v` are horizontal and vertical positions of a bin. Bins are labeled from bottom left to top right, and if a bin is located at (`u`, `v`), the label is `v*l+u+1`. Variance calculation follows.

```
nbar=(double)N_all/(double)(l*l);
a=0;
for(c=1;c<=l*l;c++)
{
a=a+(bin[c]-nbar)*(bin[c]-nbar)/((double)l*(double)l);
}
var=a-nbar;
```

In the last line, background fluctuation is removed.

B Miscellaneous components on time evolution

B.1 Periodic boundary condition

```
for(n=1; n<=N_all; n++) {
while(p[n].x>L){p[n].x=p[n].x-L;}
while(p[n].x<0.0){p[n].x=p[n].x+L;}
while(p[n].y>L){p[n].y=p[n].y-L;}
while(p[n].y<0.0){p[n].y=p[n].y+L;}
```

B.2 Source term

Source term is realized by adding a new particle one by one in accordance with $S(\mathbf{x})$. In the case of a square source, a new particle is generated every $\Delta n/\Delta t$, using a probability distribution function

$$p(x,y) = \begin{cases} \frac{1}{a^2} & \mathbf{x} \in \left[-\frac{a}{2}+\frac{L}{2}, \frac{a}{2}+\frac{L}{2}\right]^2 \\ 0 & \text{otherwise} \end{cases} \tag{39}$$

The corresponding code is

```
void source(double a, double *xo, double *yo){
double x1,x2;
x1=a*(double)rand()/(double)RAND_MAX+0.5-0.5*a;
x2=a*(double)rand()/(double)RAND_MAX+0.5-0.5*a;
*xo=x1;
*yo=x2;
}
```

A monochromatic source is a little bit difficult. We need to generate a pair of random numbers which follow

$$p(x,y) = 1 + \sin\left(\frac{2\pi}{L}(x+y)\right) \quad \mathbf{x} \in [0,L]^2 \tag{40}$$

Because this probability distribution function is tilted, let us consider it in the new coordinates $(\xi,\eta) = \left(\frac{1}{\sqrt{2}}(x+y), \frac{1}{\sqrt{2}}(-x+y)\right)$. Then, the probability distribution funciton becomes

$$p(\xi) = \frac{1}{\sqrt{2}}\left(1+\sin(\frac{2\pi}{L}\sqrt{2}\xi)\right) \quad \xi \in [0,\sqrt{2}], \eta \in [0,\sqrt{2}]. \tag{41}$$

η is given by uniform random numbers, and ξ is given by mapping uniform random numbers by the inverse of the cumulative distribution function of (41):

$$\frac{\xi}{\sqrt{2}} + \frac{L}{4\pi} - \frac{L}{4\pi}\cos\left(\frac{4\pi}{\sqrt{2}L}\xi\right) \tag{42}$$

The subroutine `inv` maps a uniform random number `y` with the function above by using bisection method up to the accuracy of `0.01*deltal`. Then `source` subroutine rotates the frame by $\frac{\pi}{4}$ and imposes periodic boundary conditions.

```
void source(double deltal, double *xo, double *yo){
double x1,x2,y,z1,z2;
y=(double)rand()/(double)RAND_MAX;
x1=inv(y,deltal);
x2=sqrt(2.0)*(double)rand()/(double)RAND_MAX;
```

```
z1=(x1-x2)/sqrt(2.0)+0.5;
z2=(x1+x2)/sqrt(2.0)-0.5;
while(z1>1.0){z1=z1-1.0;}
while(z1<0.0){z1=z1+1.0;}
while(z2>1.0){z2=z2-1.0;}
while(z2<0.0){z2=z2+1.0;}
*xo=z1;
*yo=z2;
}

double inv(double y, double deltal)
{
double small, mid, large;
int i;
small=0.0;
large=sqrt(2.0);
while(large-small>0.01*deltal)
{
mid=0.5*(large+small);
if(func(large,y)*func(mid,y)<0.0){small=mid;}
else{large=mid;}
}
return(mid);
}

double func(double x,double y)
{
double z;
z=-y+x/sqrt(2.0)+1.0/(4.0*pi)-cos(4.0*pi*x/sqrt(2.0))/(4.0*pi);
return(z);
}
```

B.3 distorted Gaussian profile: (noise) subroutine

The following code is just the noise parts of (19) and (20). Normal Gaussian noises (y1, y2) are generated by using Box-Muller method.

```
void noise(double kappa, double S, double dt, double *rand1, double
*rand2) {

double x1,x2,y1,y2, a,b,c;
x1=((double)rand()+0.01)/((double)RAND_MAX+0.01);
x2=((double)rand()+0.01)/((double)RAND_MAX+0.01);
```

```
y1=sqrt(-2.0*log(x1))*cos(2.0*pi*x2);
y2=sqrt(-2.0*log(x1))*sin(2.0*pi*x2);
a=S*sqrt(0.5*kappa)*dt*sqrt(dt);
b=sqrt(kappa*S*S*dt*dt*dt/6.0+2.0*kappa*dt);
c=sqrt(2.0*kappa*dt);
*rand1=a*y1+b*y2;
*rand2=c*y1;
}
```

B.4 Subtraction scheme

It is easy to implement subtraction scheme. Subtraction of $N_{\rm old}$ older particles is done by re-labeling particles.$(N_{\rm old}+i)$th particle becomes ith particle, and the number of all the particles gets smaller by $N_{\rm old}$. In this code, $N_{\rm old}=\Delta n$.

```
if(j>=cutoff){
N_all=N_all-deltan;
for(n=1;n<=N_all;n++){
p[n].x=p[n+deltan].x;
p[n].y=p[n+deltan].y;
}
}
```

C Time evolution

The following code is time evolution from the beginning of the period to variance calculation.

```
phi=2.0*pi*rand()/(double)RAND_MAX;
randt=(double)rand()/(double)RAND_MAX;
for(c=1;c<=l*l;c++)
{
bin[c]=0;
}
a=0.0;
t_obs=deltat*randt;
N_obs=(int)(deltan*randt);
for(n=1;n<=N_all;n++)
{
noise(kappa, -w*2.0*pi*cos(2.0*pi*p[n].y/L+phi)/L, t_obs, &rand1,
&rand2);
p[n].x=p[n].x-t_obs*w*sin(2.0*pi*p[n].y/L+phi)+rand1;
p[n].y=p[n].y+rand2;
while(p[n].x>L){p[n].x=p[n].x-L;}
```

```
while(p[n].x<0.0){p[n].x=p[n].x+L;}
while(p[n].y>L){p[n].y=p[n].y-L;}
while(p[n].y<0.0){p[n].y=p[n].y+L;}
u=(int)(p[n].x/deltal);
v=(int)(p[n].y/deltal);
bin[v*l+u+1]=bin[v*l+u+1]+1;
}
for(n=1;n<=N_obs;n++)
{
dt=t_obs-deltat*(double)n/(double)deltan;
source(deltal, &xo, &yo);
noise(kappa, -w*2.0*pi*cos(2.0*pi*yo/L+phi)/L, dt, &rand1, &rand2);
p[n+N_all].x=xo-dt*w*sin(2.0*pi*yo/L+phi)+rand1;
p[n+N_all].y=yo+rand2;
while(p[n+N_all].x>L){p[n+N_all].x=p[n+N_all].x-L;}
while(p[n+N_all].x<0.0){p[n+N_all].x=p[n+N_all].x+L;}
while(p[n+N_all].y>L){p[n+N_all].y=p[n+N_all].y-L;}
while(p[n+N_all].y<0.0){p[n+N_all].y=p[n+N_all].y+L;}
u=(int)(p[n+N_all].x/deltal);
v=(int)(p[n+N_all].y/deltal);
bin[v*l+u+1]=bin[v*l+u+1]+1;
}
N_all=N_all+N_obs;
nbar=(double)N_all/(double)(l*l);
```

First, random phase (`phi`), the time to calculate variance (`t_obs`) and the number of particles added into the domain from the beginning of the period to the variance calculation (`N_obs`) are calculated. In the first `for`-loop, the particles which already existed at the beginning of the period are evolved based on (19) and (20), boundary conditions are imposed and (coarse-grained) concentration field is recovered by binning. In the following `for`-loop, new particles from a source term are added with `source` subroutine. `dt` is a time from particle creation to the variance calculation, which is different from particle to particle. Those new particles are evolved in the same way as existing particles. The time evolution after the variance calculation is implemented in the same way. The time evolution of the second half of the period is similar except that the random sine flow is vertical.