The shadow position mixing model tested for two turbulent flows

by

S.C. Ruiter

to obtain the Bachelor degrees Applied Physics and Applied Mathematics,
at the Delft University of Technology,
to be defended publicly on August 30, 2018 at 15:00.

Student number:	4311582	
Thesis committee:	Prof. Dr. D.J.E.M. Roekaerts	TU Delft, supervisor
	Prof. Dr. Ir. A.W. Heemink	TU Delft, supervisor
	Dr. Ir. F.H. van der Meulen	TU Delft
	Dr. S.R. de Roode	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Abstract

Predicting the behavior of turbulent flows and how thermochemical properties develop such as temperature, enthalpy or concentrations is an import aspect in various fields. However, numerical calculations have a high time complexity. Therefore, models were constructed to approximate these flows. This thesis focuses on using probability density functions (pdf) to predict scalar probabilities in these flows.

Since the pdf does not contain spatial information, small scaled mixing has a closure problem and has to be modeled using micromixing. A micromixing model, called the Shadow Position Mixing Model (SPMM) was introduced previously to calculate this mixing. In this project, its performance is determined for the mixing in two different flows.

The model shows promising results for mixing of two passive scalar, initially mainly in one of three states. Comparing the results of various mixing models to the provided DNS, the SPMM predicts the evolution of the pdf over time adequately, but still has visible differences compared to the DNS.

In the second test, the model is tested using an imposed mean scalar gradient in order to check the convergence of the mixing model. The SPMM does show promising results, but can not reliably predict a correct statistically stationary state. This is due to the fact that varying number of scalars and particles lead to significant inconsistencies. Under the current conditions, the SPMM cannot be reliably applied to flows with such an additional scalar force.

The SPMM uses a near-neighbor sorting algorithm which has a high algorithmic complexity (order $O(n^2)$). For practical purposes, a lower complexity algorithm has to be developed in order to use this model for high number of particles.

Contents

Declaration of variables

Abbrevations	
pdf	probability density function
jpdf	joint pdf
DNS	Direct Numerical Simulation
SPMM	Shadow Position Mixing Model
IEM	Interaction by Exchange with the Mean mixing model
CD	Coalescence-Dispersion mixing model
MCD	Modified Coalescence-Dispersion mixing model

Symbols	
a, b	Shadow displacement coefficients
C_ϕ	Micromixing model coefficient
C_0, C_t	Langevin model coefficients
f	Joint pdf
i	Indicator for particle i
k	Turbulent kinetic energy
n_p	Number of particles in simulation
n_s	Number of scalars
R	Shadow displacement
t	Time
Δt	Discrete timestep
V	Sample space variable of U
w	Weight of a particle
W_t	A Wiener process
$\mathrm{d}W_t$	A increment of the W_t on interval $[t-\Delta t, t]$
X	Cartesian vector in physical space
U	Velocity
ϕ	Composition scalar
ψ	Sample space variable of ϕ
ω	Turbulent frequency
θ	Sample space variable of ω
ϕ'	Fluctuations in ϕ, i.e. $\phi' = \phi - \langle\phi\rangle$
ϕ''	Variance in ϕ, i.e. $\phi'' = \langle\phi'^2\rangle$
ϵ	Rate of turbulence energy dissipation

1

Introduction

Turbulence is present in almost every fluid. An obvious example of this, is a vigorously flowing river, but also a person walking creates turbulent air flows. Even the most constant looking flow has turbulent parts near the edges, due to friction of these edges. Predicting the behavior of turbulent flows and how thermochemical scalars develop such as temperature, enthalpy or concentration, has therefore numerous applications.

The most accurate way to calculate the scalars exactly is using Direct Numerical Simulations (DNS), which doesn't use any turbulence model, but directly solves the Navier-Stokes equation in space and time. However, these calculations are very time consuming and are therefore computationally infeasible for complex and industrial applicable flows.

Alternatively, turbulence can be described as a stochastic process, and statistical properties can be calculated. One of these methods is the pdf method. In this instance, probability density functions (pdf) are used to predict the probability of a thermochemical scalars in a certain place and time.

Due to diffusion, convection, reaction and other physical phenomena, scalars change and mix over time. However, this mixing is not closed in pdf methods. A closure problem present itself in a equation, when a term cannot be solved directly using the equation, but has to be modelled. For the mixing closure problem, *micromixing* modeling is introduced to calculate this mixing. Unfortunately, all micromixing modeled present today are not perfect and thus have deficiencies.

A new micromixing model is introduced by Pope[2]. The model is called the ”Shadow position mixing model” (SPMM). To further investigate the performance of this model, it is applied for mixing in two different flows.

2

Micromixing

2.1. Turbulence

Turbulent flows occur in a fluid, when parts of the flow have different velocities. Due to these differences, swirls in the fluid are created and are in fluid dynamics also called eddies. They can be created with different sizes and energies depending on the fluid properties and energy that is inserted in it. These swirls are again disturbances in the flow, and thus create other smaller or equal sized eddies of there own. There is a minimum length to this where the rate at which these eddies are created is smaller then that they are dampened. This scale is called the Komogorov length scale. This is a important value of numerical simulations.

Numerical simulations are an important method to predict the behavior of flows in industrial, biochemical or other applications. Especially when reactions, mass transfer and heat transfer are considered, turbulence has a big effect and the simulations give more insight into this and thus help optimizing the applications. They are in most cases less costly then empirical experiments.

2.2. Numerical simulations

One of these numerical methods is Direct Numerical Simulations (DNS), where the Navier-Stokes equations are solved directly. The Navier-Stokes equations are partial differential transport equations that describe the flow of fluids. To solve the turbulent flow, it is divided in a grid and these equations are computed for each grid cell. However, to incorporate all eddies, the Komogorov length scale is used to determine the grid size. The length scale could be in de order of millimeters, while the largest eddies could be kilometers long.

Since this scale is very small in comparison to the entire flow, the DNS calculations have a high computational cost and are therefore not a great option for complex flows. Instead, models are used to predict the behavior, one of which is to use time averaged equations. These equations give time-averaged solutions of the Navier-Stokes equations.

However, due to the nonlinear nature of the Navier-Stokes equations, averaging them still leads to a closure problem where one term cannot be resolved from the model alone and a turbulence model is needed to resolve this.

2.3. PDF methods

One of the models to give averaged statistics are the pdf transport models, which use joint probability density functions (pdf) to solve the turbulent flow. In the pdf methods, a joint pdf represents the one point statistics of thermochemical scalar variables in combination with other characteristics of the flow. The evolution of these properties can then be predicted by a transport equation.

The transport equation for the joint pdf $f(x,t;V,\psi,\theta)$ can be derived, as shown in appendix A.1. The jpdf contains all information about V, ψ and θ given a certain position x and time t. An expression

for the joint velocity-composition-turbulent frequency pdf is as follows,

$$\frac{\partial f}{\partial t} + \vec{U}\frac{\partial f}{\partial \vec{x}} = -\frac{\partial}{\partial \psi}\left(f\left\langle \frac{\mathrm{d}\phi}{\mathrm{d}t}\middle|\vec{V},\psi,\theta\right\rangle\right) - \frac{\partial}{\partial \theta}\left(f\left\langle \frac{\mathrm{d}\omega}{\mathrm{d}t}\middle|\vec{V},\psi,\theta\right\rangle\right) - \frac{\partial}{\partial \vec{V}}\left(f\left\langle \frac{\mathrm{d}\vec{U}}{\mathrm{d}t}\middle|\vec{V},\psi,\theta\right\rangle\right). \tag{2.1}$$

Incorporating the turbulent frequency ω is a way to describe how rapid the thermochemical scalars and velocity change over time. In this project, the turbulent frequency is defined by $\langle\omega\rangle = \frac{\epsilon}{k}$, where k is the turbulence kinetic energy and ϵ the rate of turbulence energy dissipation. In most cases, these variables are deduced form the $k-\epsilon$ turbulence model.

However, since the joint pdf does not provide spatial information for the particles, this method is not closed for small scaled mixing. Therefore, the term $\frac{\mathrm{d}\phi}{\mathrm{d}t}$ has to be solved using an additional *micromixing* model. In general, the term can be written as

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = \nabla\cdot(D\nabla\phi) + S, \tag{2.2}$$

where $\nabla\cdot(D\nabla\phi)$ will be solved using the micromixing models, and S a closed scalar source term, e.g. as a result of chemical reaction. D is a scalar diffusion coefficient.

To solve eq. 2.1, a particle model is used. During this project, the Lagrangian approach is used, in which a particle represents a pocket of fluid and is tracked over time. This particle has all the thermochemical scalar properties of significance, as well as a position, velocity and turbulent frequency. In this approach, stochastic methods are used to simulate the turbulence, in this case using the Langevin equation for the velocity. The thermochemical scalar properties will be solved using different micromixing models.

In the initial state, the particle is described by the initial pdf and evolves according to the stochastic and micromixing models.

2.4. IEM model

A most basic and well-known deterministic micromixing model is the Interaction by Exchange with the Mean (IEM). The essence of this model is eq. 2.3. Here, C_ϕ is a model constant which determines the speed of the mixing. This constant has to be referenced to experimental or DNS data. A term $S(\phi_i)$ could be added, to describe a source term, due to various reasons, such as chemical reaction.

$$\frac{\mathrm{d}\phi_i}{\mathrm{d}t} = -\frac{1}{2}C_\phi\langle\omega\rangle(\psi_i - \langle\phi\rangle) + S(\phi_i) \tag{2.3}$$

This equation describes the change of scalar ϕ over time of particle with index i. To give a physical meaning to this model, combining 2.2 and 2.3 results in

$$\langle\nabla\cdot(D\nabla\phi_i)|\psi,\theta\rangle = -\frac{1}{2}C_\phi\langle\omega\rangle(\psi_i - \langle\phi\rangle) \tag{2.4}$$

This means, that the ϕ will always relax to same mean in a computational cell. The means are calculated as follows, where ψ_i and w_i are the scalar value and weight respectively of particle i,

$$\langle\phi\rangle = \frac{1}{W}\sum_{i=1}^{N} w_i\psi_i \text{ and } W = \sum_{i=1}^{N} w_i \tag{2.5}$$

Since all particles in a computational cell are used to calculate the mean, and the spacial coordinate of each particle is ignored, the mixing in the IEM model is not a local process while mixing is considered to be local.

Another drawback of this mean, is that all particles in a cell converge to the same mean at the same rate. Due to this, the IEM model can predict the decaying variance of a scalar quite well, but does not change the initial shape of the joint pdf[14]. Therefore, the IEM will not give promising results in a lot of test cases were pdf evolutions are studied.

Another deficiency of the IEM model, is the inconsistency with the dispersion theory[15]. For high Reynolds number, $\langle\phi\rangle$ should only be affected by the fluid's motion and not by diffusion, i.e.

$$\left\langle U\frac{\mathrm{d}\phi}{\mathrm{d}t}\right\rangle = 0. \tag{2.6}$$

For the IEM model, this condition does not hold.

2.5. CD model

The next mixing model to be evaluated, is the Coalescence-Dispersion mixing model (CD), which still has a similar non-localness problem as the IEM model, but is not shape conserving.

In this model, two random picked particles i, j mix completely to their combined mean, resulting in the following mixing equation.

$$\psi_i = \psi_j = \frac{w_i\psi_i + w_j\psi_j}{w_i + w_j}. \tag{2.7}$$

To obtain the correct speed of the mixing, the amount of randomly selected mixing pairs per time step is

$$N_p = ceil(2C_\phi W\Delta t\,\langle\omega\rangle). \tag{2.8}$$

The drawback of this model, is it's physical interpretation. Each timestep, the selected mixing pairs mix completely, which does not happen in nature. Also, only a number of particles mix and the remaining particles do not change at all. Therefore, the Modified Coalescence-Dispersion mixing model (MCD) is introduced, removing the discreteness of mixing. For a uniformly distributed random variable $h \in [0, 1]$, the model becomes

$$\psi_i = \psi_i + h\left(\frac{w_i\psi_i + w_j\psi_j}{w_i + w_j} - \psi_i\right) \text{ and } \psi_j = \psi_j + h\left(\frac{w_i\psi_i + w_j\psi_j}{w_i + w_j} - \psi_j\right). \tag{2.9}$$

This gives the particle i and j the ability to assume all values between the to scalar ψ_i and ψ_j. However, still not all particles mix simultaneously.

As studied in [3], to obtain equal variance decay rates for the CD model as the IEM model, equal C_ϕ coefficients have to be used. For the MCD coefficient, this is $C_{\phi,mcd} = \frac{3}{2}C_{\phi,CD}$.

2.6. SPMM

A new model to fix some of the deficiencies present in other models is introduced by Pope[2], called the shadow-position mixing model (SPMM). In this model, each particle i has a position X_i, similar to the other mixing models. In the SPMM, an *shadow position* Z_i is also attributed to each particle i. The resulting *shadow displacement* is defined as

$$R_i(t) \equiv Z_i(t) - X_i(t). \tag{2.10}$$

De evolution of the Z_i is defined by a stochastic differential equation, and as a result the R_i evolves according to eq. 2.11,

$$\mathrm{d}R_i = -aR_i\frac{\mathrm{d}t}{T_L} - (U - \langle U|X\rangle)\mathrm{d}t + b\sqrt{2\sigma^2 T_L}\mathrm{d}W_t, \tag{2.11}$$

where T_L can be modeled as $T_L = \frac{C_T}{\langle\omega\rangle}$ and $\sigma^2 = \frac{2}{3}k$. a and b are model coefficients. In order to make the SPMM consistent with the turbulent dispersion theory, eq. 2.6 has to hold. As a result, there is a relation between the model coefficients,

$$b = \frac{1}{1+a}. \tag{2.12}$$

Using the shadow displacement R_i, the SPMM is defined as

$$\frac{\mathrm{d}\phi_i}{\mathrm{d}t} = -\frac{C_\phi}{T_L}(\psi_i - \langle\phi|X, R\rangle). \tag{2.13}$$

This equation looks fairly similar to the IEM model, but here the mean ϕ is replaced by a mean conditioned on R_i. The source term is omitted without loss of generality.

This last model coefficient C_ϕ also has a relation with a. As explained in III.D of [2], there is a minimum value for a. Using DNS data, $a_{min} \approx 0.87$ and the resulting condition for C_ϕ becomes

$$\frac{1}{C_\phi} = 2\left(\frac{1}{a_{min}} - \frac{1}{a}\right). \tag{2.14}$$

To calculate the conditional mean in eq. 2.13, the suggested numerical implementation by Pope[2] is used. In this calculation, the mean is calculated by

$$\langle \phi_i | X, R \rangle = \frac{w_{i-1}\psi_{i-1} + w_{i+1}\psi_{i+1}}{w_{i-1} + w_{i+1}}. \tag{2.15}$$

In this expression, ψ_{i-1}, ψ_{i+1} are the near-neighbors of ψ_i in the (X, R) sample space. An algorithm to implement this is included in appendix A.3. An example of algorithm is displayed in fig 2.1. In this example, $n_p = 20$ particles are simulated in a 1D sample space using a uniform distribution. The lines indicate the near-neighbor particles according to the algorithm.

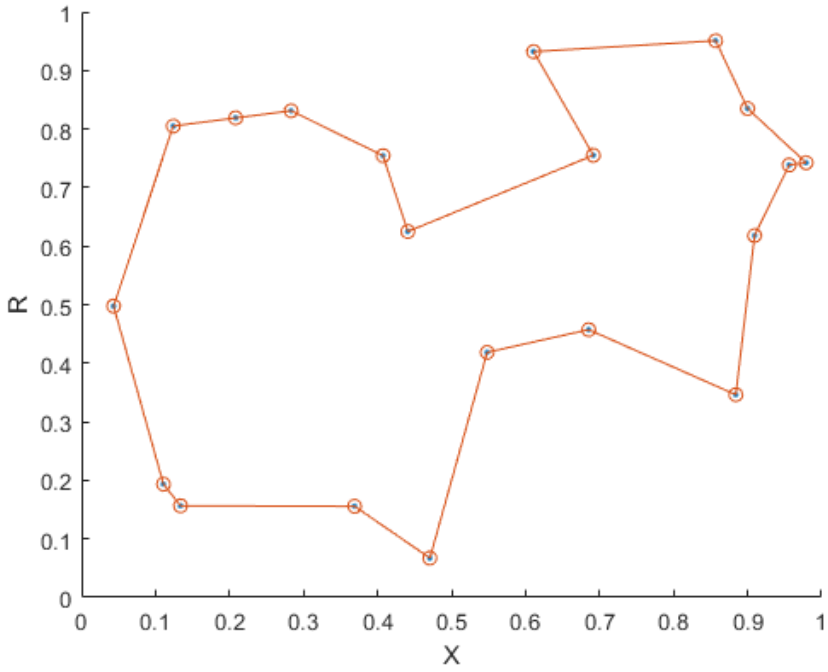


Figure 2.1: An example of the near-neighbor sorting algorithm given in A.3, each point represents a particle with a X and R value. The lines indicate which are the near-neighbors pairs.

As can be seen, the selected set of near-neighbor pairs are dependent on the starting particle, and is not unique. In general, particle relatively far away can still be near neighbor particles according to the algorithm.

The idea behind the SPMM is to lift non-localness deficiency of the other mixing model. By calculating the mean using the near-neighbor principle, this is clearly the case.

Other models have already been introduced to tackle the non-localness principle, one of which is by using a Euclidean Minimum Spanning Tree (EMST)[9] to determine near-neighbors in scalar space. Moreover, this model leads to stranding, since is for the main part unchanged per time step which leads to undesirable results.

The SPMM does not have this deficiency. As the shadow displacement R_i fluctuates each time step around $\langle R \rangle = 0$, there is a high chance that new mixing pairs are selected each time step, especially when n_p is high.

To validate the accuracy of this model, it has to be tested against empirical or previously calculated data. In the following two test cases, a turbulent flow is considered for which DNS data is already provided. During this project, the SPMM model will be tested on these flows as well as the other mixing models, in order to get a better understanding of the performance of this model.

3

Mixing two passive scalars

The first test case on which the SPMM will be tested, is the three stream problem. In this test case, a stationary homogeneous turbulent flow is considered in which two passive scalars mix. A homogeneous stationary turbulent flow means that it is unaffected by translations in space and time. In other words, the flow is statistically the same for every point in space and in time.

For this problem, the DNS results have already been provided[5] and the scalar mixing has been computed using the IEM, CD and MCD model[3]. In this thesis, this will be repeated using SPMM. For better comparing the mixing models, this mixing is also calculated with the IEM, CD and MCD.

In this test case, no external source will be added, which reduces eq. 2.2 to

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = D\nabla^2\phi. \tag{3.1}$$

This results in the following jpdf equation, where $\psi = (\psi_1, \psi_2)$,

$$\frac{\partial f}{\partial t} + \vec{U}\frac{\partial f}{\partial \vec{x}} = -\frac{\partial}{\partial \psi}\left(f\left\langle D\nabla^2\phi \middle| \vec{V}, \psi, \theta \right\rangle\right) - \frac{\partial}{\partial \theta}\left(f\left\langle \frac{\mathrm{d}\omega}{\mathrm{d}t} \middle| \vec{V}\psi, \theta \right\rangle\right) - \frac{\partial}{\partial \vec{V}}\left(f\left\langle \frac{\mathrm{d}\vec{U}}{\mathrm{d}t} \middle| \vec{V}, \psi, \theta \right\rangle\right) \tag{3.2}$$

To solve eq. 3.2, it will be written as a set of equations using a Lagrangian reference frame and a $k-\epsilon$ turbulence model.

For relatively high Reynolds numbers, the Langevin equation approximately describes the development of speed, and thus of a sample particle's position. The Langevin equations for the particle position and velocity are

$$\mathrm{d}x_i = U\mathrm{d}t \tag{3.3}$$

$$dU_i = -U_i\frac{\mathrm{d}t}{T_L} + \sqrt{\frac{2\sigma^2}{T_L}}\mathrm{d}W_t. \tag{3.4}$$

In eq. 3.4, W_t is an independent Wiener process, different for each particle and each velocity component, $\sigma = \mathrm{rms}(U)$ and $T_L = \frac{C_T}{\langle\omega\rangle}$. Next, the turbulent frequency, a rate for the speed of the mixing, will also develop using the Langevin equation where $\sigma_\omega = \mathrm{rms}(\omega)$.

$$\mathrm{d}\omega_i = -(\omega_i - \langle\omega\rangle)\frac{\mathrm{d}t}{T_\omega} + \sqrt{\frac{2\sigma_\omega^2\langle\omega\rangle\omega_i}{T_\omega}}\mathrm{d}W_t. \tag{3.5}$$

Again, W_t is an independent wiener process and a different from W_t in 3.4 and different for each particle, and $T_\omega = \frac{1}{\langle\omega\rangle}$. To actually calculate the evolution of the scalar, this will be calculated using the mixing models, thus the SPMM model is following 2.13

$$\mathrm{d}\phi_i = -\frac{C_\phi}{T_L}(\psi_i - \langle\phi|X, R\rangle)\mathrm{d}t. \tag{3.6}$$

3.1. Numerical integration

Since (stochastic) differential equations are numerically integrated, choosing the correct time step is of utmost importance. If the integrating time step is too large, integration becomes unstable.

At first, we consider deterministic equation 3.3. Using the Euler forward approximation, we get

$$X_{n+1} = X_n + U\Delta t \tag{3.7}$$

where $X_0 = x(0)$ and $X_n = x(t_n)$ with $t_n = n\Delta t$. This is an Euler forward integration where a constant value U is integrated. Since Euler forward is a first-order numerical procedure and integrating a constant is also first order, this numerical approximation is stable for all time steps. Thus for $\Delta t > 0$, this equation is stable[1].

Next, we consider equation 3.5. Since this equation is hard to solve on itself, with statistical properties in each term, we first consider $\langle\omega\rangle$ and σ^2 as constant. This is satisfied for this application, where $\langle \mathrm{d}\omega\rangle = 0$. This reduces equation 3.5 to equation 3.8, which is a mean reverting square root process.

$$\mathrm{d}\omega_i = \frac{1}{T_\omega}(\langle\omega\rangle - \omega_i)\mathrm{d}t + \alpha\sqrt{\omega_i}\mathrm{d}W_t \text{ with } \alpha = \sqrt{\frac{2\sigma^2\langle\omega\rangle}{T_\omega}} \tag{3.8}$$

For this SDE, we get $\lim_{t\to\infty}\langle\omega(t)\rangle = \langle\omega\rangle$ and $\lim_{t\to\infty}\mathrm{var}(\omega_t) = \frac{\alpha^2\langle\omega\rangle T_\omega}{2}$[12].

As this process is mathematically equivalent to an Itô process, we will use the Euler-Maruyama (EM) method to numerically integrate the equation. Using the EM method, we approximate this stochastic process.

$$\Omega_{n+1} = \Omega_n + \frac{1}{T_\omega}(\langle\omega\rangle - \Omega_n)\Delta t + \alpha\sqrt{\Omega_n\Delta t}Z \tag{3.9}$$

where $\Omega_0 = \omega(0)$, $\Omega_n = \omega(t_n)$ and Z a random number picked from a normal distribution $N(0,1)$. This approximation is stable for

$$\left|1 - \frac{\Delta t}{T_\omega}\right| < 1. \tag{3.10}$$

Choosing $\Delta t < 2T_\omega = \frac{2}{C\langle\omega\rangle}$, we know that the approximation 3.9 converges in the first two moments[12].

Since the $\langle\omega\rangle$ and σ^2 were assumed constants, the expected stationary mean and variance might fluctuate per time step. Choosing a value for Δt such that the condition for Δt is never exceeded is necessary.

Next, we consider equation 2.11. Again, using the EM method, we get for R the following approximation 3.11,

$$r_{n+1} = r_n - \frac{a}{T_L}\left(r_n - \frac{T_L}{a}(U_n - \langle U_n|X_n\rangle)\right)\Delta t + b\sqrt{2\sigma^2 T_L\Delta t}Z \tag{3.11}$$

where $r_0 = R(0)$, $r_n = R(t_n)$, $t_n = n\Delta t$ and Z again a $N(0,1)$ distributed random selected number. Here, the first and second moment converges if

$$\left|1 - \frac{a}{T_L}\Delta t\right| < 1$$

thus if $\Delta t < \frac{2T_L}{a} = \frac{2C_T}{a\langle\omega\rangle}$.

At last, we check equation stability for the SPMM and IEM model, eq. 2.3,2.13. Again, here we take the means to be constant. Using the numerical approximation derived in appendix A.2, these equations become

$$\phi_{n+1} = \phi_n + \left[\exp\left(-\frac{1}{2}C_\phi\langle\omega\rangle\Delta t\right) - 1\right](\phi_n - \langle\phi\rangle), \tag{3.12}$$

which is derived in the appendix. This approximation is stable if

$$\left|\exp\left(-\frac{1}{2}C_\phi\langle\omega\rangle\Delta t\right)\right| < 0 \Rightarrow \Delta t > 0. \tag{3.13}$$

Since the evolution over time is also quite important in this project, the timestep needs to be sufficiently small. For this flow, we take

$$\Delta t = \min\left(\frac{2}{C\langle\omega\rangle}; \frac{2C_T}{a\langle\omega\rangle}; \frac{0.01}{\langle\omega\rangle}\right) = \frac{0.01}{\langle\omega\rangle} \tag{3.14}$$

3.2. Initial condition

To start off the experiment, the initial conditions are shown below in table 3.1.

Table 3.1: Initial conditions for the three stream flow

$\Delta t = \frac{0.01}{\langle\omega\rangle}$	$U_i(0) \sim N(0, \frac{2}{3}k)$
$k = 577m^2/s^2$	$R_i(0) \sim N(0,1)$
$\epsilon = 2641J/(kg \cdot s)$	$X_i(0) \sim N(0,1)$
$N = 10000$	$\omega_i(0) \sim N\left(\frac{\epsilon}{k}, \frac{\epsilon}{2k}\right)$
$C_T = 0.5$	

The main focus of this experiment is the development of the scalar pdf over time. The scalar pdf initial condition is shown in fig. 3.1. Most sample particles are at or close to one of three states. The pdf then has three clear peaks.

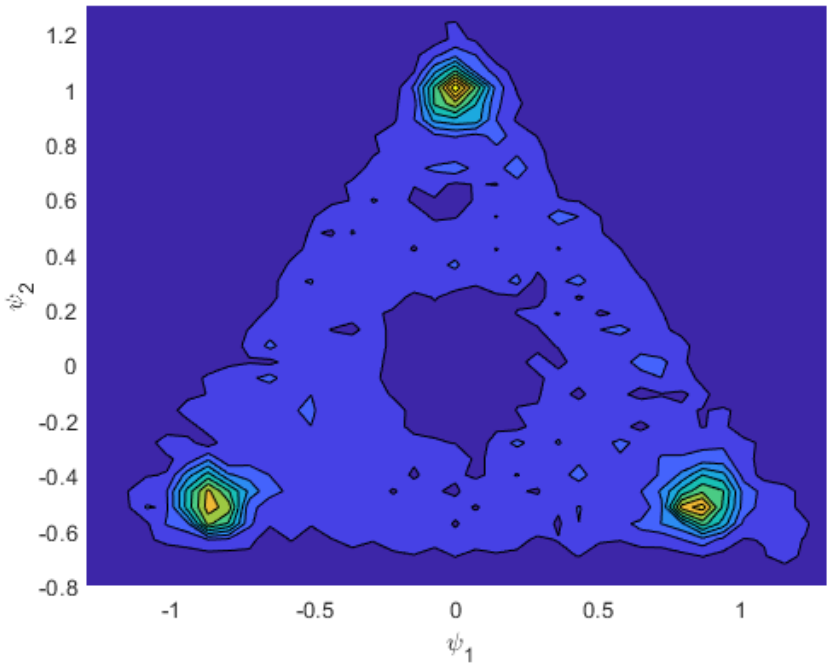


Figure 3.1: Initial condition of the joint scalar pdf. In this figure, the blue color represents the minimum value of the jpdf, and orange the maximums. The pdf clearly has 3 distinct peaks. A particle has a high change to be in either one of 3 states.

In order to better compare the SPMM model to the other models, the model coefficients have to be chosen such that the rms-decay is similar to the DNS data, fig 12[5]. Doing this, the rms-decay rate of ϕ_1 are similar as shown in fig. 3.2.

As mentioned in section 2, the model coefficient for the IEM, CD and MCD model are $C_\phi = 2$, $C_\phi = 2$ and $C_\phi = 3$ respectively. For the SPMM to have the same rms decay rate, $a = 6$ has to be selected such that $C_\phi \approx 0.5$. Note that, with the used coefficient of $C_T = 0.5$, the mixing rate of the IEM model

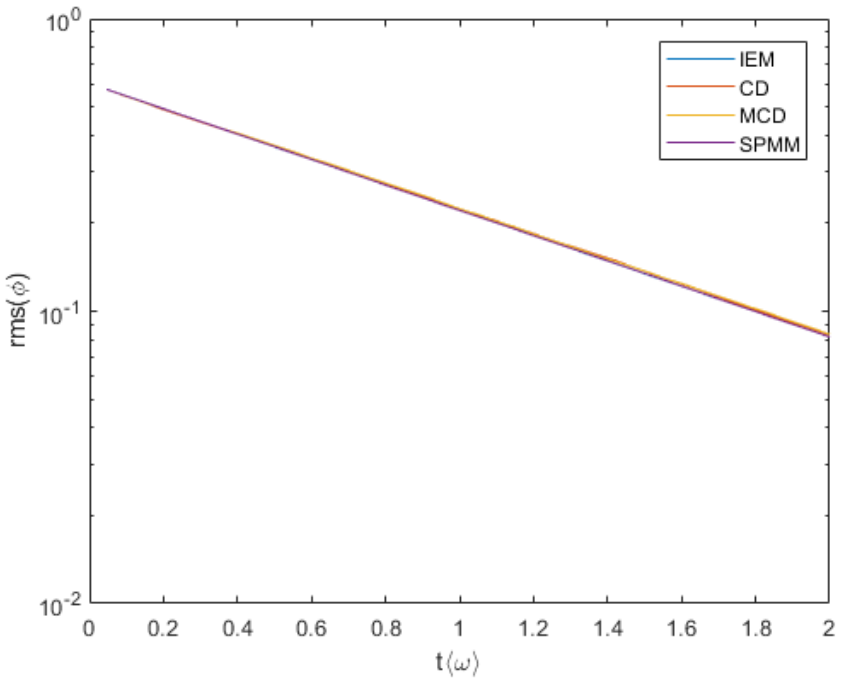


Figure 3.2: The rms decay of different mixing models over time. The model coefficients are $C_{\phi,IEM} = 2$, $C_{\phi,CD} = 2$, $C_{\phi,MCD} = 3$ where the subscript indicates the mixing model. For the SPMM, $a = 6$, $b = 0.14$, $c = 0.51$ are used.

and SPMM have similar values, i.e.

$$\frac{1}{2}C_{\phi,IEM}\langle\omega\rangle \approx \frac{C_{\phi,SPMM}\langle\omega\rangle}{C_T}. \tag{3.15}$$

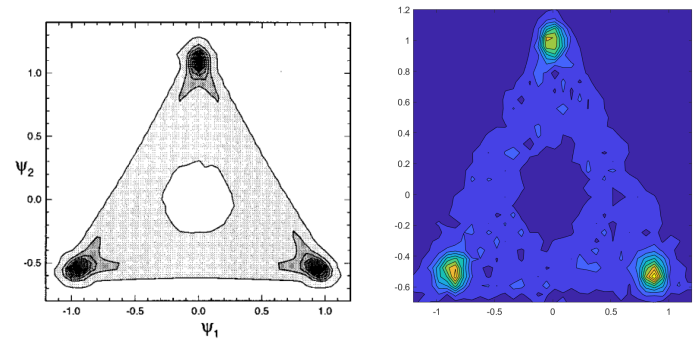
3.3. Results

In fig. 3.3, the jpdf f at different moments in time are shown, developed using SPMM model with a coefficient $a = 6$, as well as the DNS data[5]. In fig. 3.4, the same jpdf evolution is displayed for the IEM, CD and MCD model, in order to better compare the models.

The jpdf is calculated using the initial conditions in table 3.1. Each time step, the particles are divided in 100 computational cell to calculate the local mean. Note that each cell has approximately 100 particles inside it.

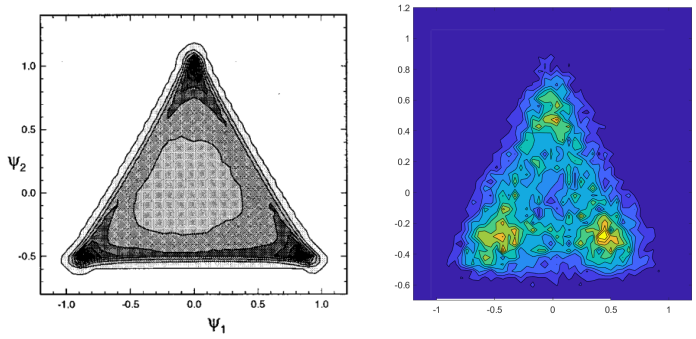
In order to display the pdf over time, additional data processing is needed to convert the data particle set to a clear pdf. This is done in Matlab[8], using a 2D histogram (Matlab function *hist3*) to calculate the number of particles in a grid. For a smooth figure, an 40x40 grid is used. Using this histogram, the *contourf* function is used, to display the calculated histogram as a filled contour plot, using 10 different levels.

The calculated time evolution of the joint pdf using the SPMM is displayed in 3.3. The evolution calculated with the other models is displayed in fig. 3.4. A more detailed evolution of the jpdf using the SPMM can be seen in Appendix A.4.



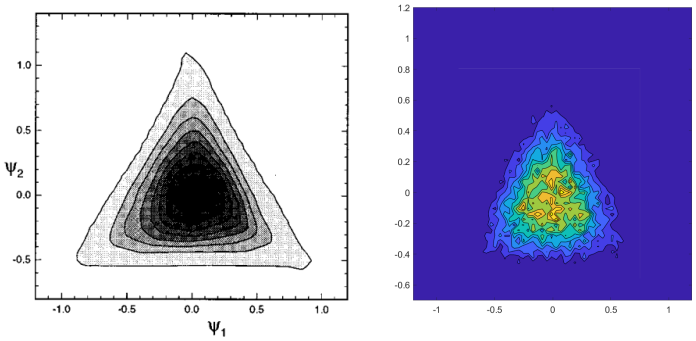
(a) DNS, $\frac{\phi''(t)}{\phi''(0)} = 1.0$

(b) a=6, $\frac{\phi''(t)}{\phi''(0)} = 1.0$



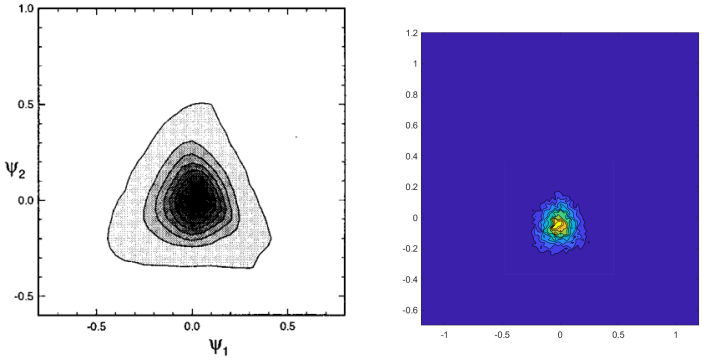
(c) DNS, $\frac{\phi''(t)}{\phi''(0)} = 0.7$

(d) a=6, $\frac{\phi''(t)}{\phi''(0)} = 0.7$



(e) DNS, $\frac{\phi''(t)}{\phi''(0)} = 0.4$

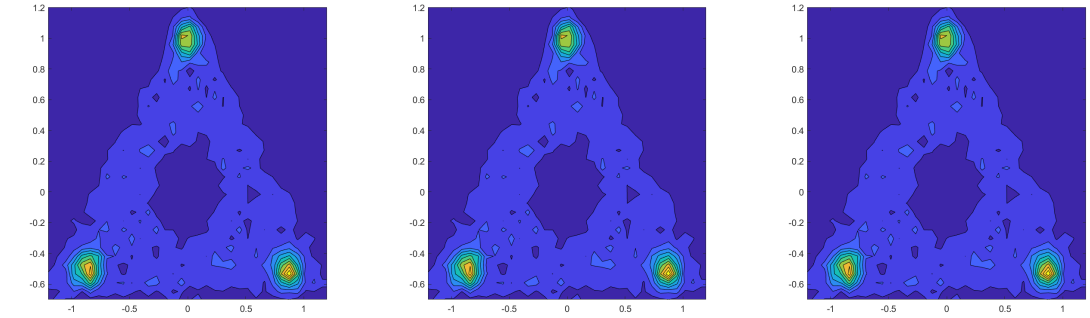
(f) a=6, $\frac{\phi''(t)}{\phi''(0)} = 0.4$



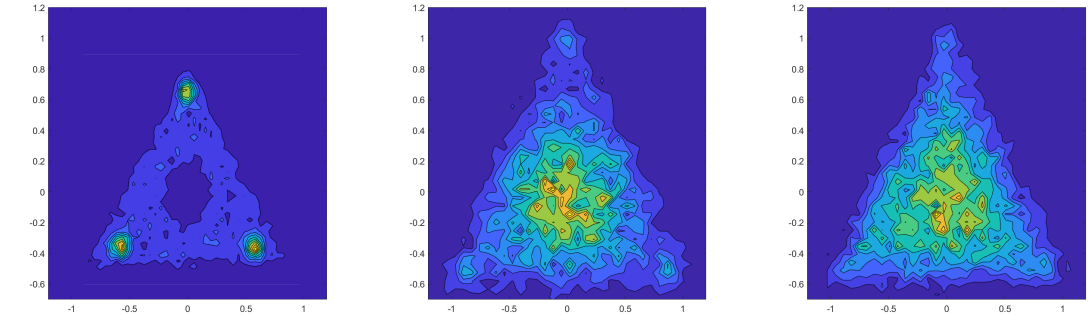
(g) DNS, $\frac{\phi''(t)}{\phi''(0)} = 0.2$

(h) a=6, $\frac{\phi''(t)}{\phi''(0)} = 0.2$

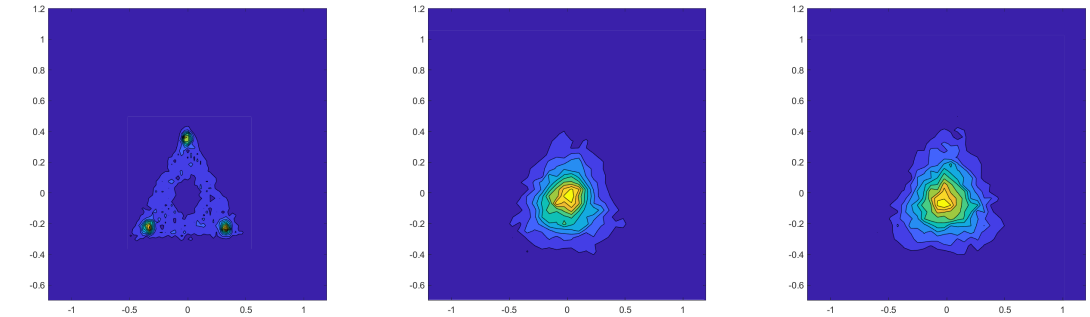
Figure 3.3: The jpdf f for different rms values. The left column (3.3a, 3.3c, 3.3e, 3.3g) displays the DNS data[5] for values $\frac{\phi''(t)}{\phi''(0)} = 1.0, 0.7, 0.4, 0.2$ respectively. The right column (3.3b,3.3d, 3.3f, 3.3h) displays the SPMM model with $a = 6$ for the same rms values.



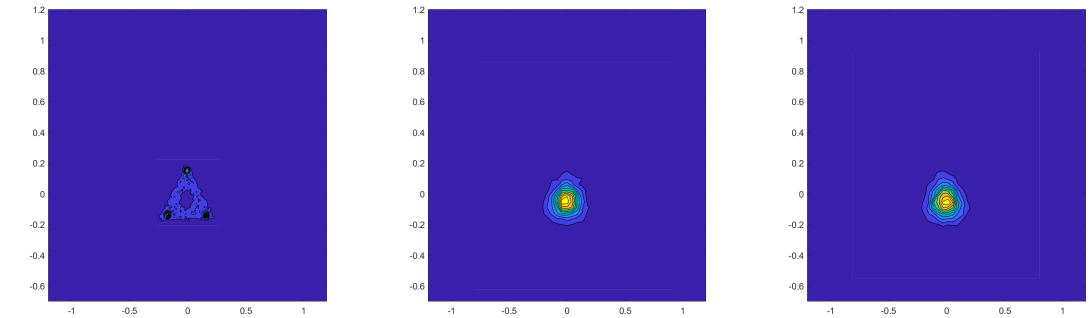
(a) IEM, $\frac{\phi''(t)}{\phi''(0)} = 1.0$ (b) CD, $\frac{\phi''(t)}{\phi''(0)} = 1.0$ (c) MCD, $\frac{\phi''(t)}{\phi''(0)} = 1.0$



(d) IEM, $\frac{\phi''(t)}{\phi''(0)} = 0.7$ (e) CD, $\frac{\phi''(t)}{\phi''(0)} = 0.7$ (f) MCD, $\frac{\phi''(t)}{\phi''(0)} = 0.7$



(g) IEM, $\frac{\phi''(t)}{\phi''(0)} = 0.4$ (h) CD, $\frac{\phi''(t)}{\phi''(0)} = 0.4$ (i) MCD, $\frac{\phi''(t)}{\phi''(0)} = 0.4$



(j) IEM, $\frac{\phi''(t)}{\phi''(0)} = 0.2$ (k) CD, $\frac{\phi''(t)}{\phi''(0)} = 0.2$ (l) MCD, $\frac{\phi''(t)}{\phi''(0)} = 0.2$

Figure 3.4: The jpdf f for different rms values. Column 1 displays the IEM model, column 2 the CD model and column 3 the MCD model for values $\frac{\phi''(t)}{\phi''(0)} = 1.0, 0.7, 0.4, 0.2$.

3.4. Discussion

Knowing that the decaying rate is similar, a comparison can be made between the joint pdf of the DNS and de micromixing models. Using the correct coefficient for the micromixing models, the rms decay of the scalar in fig. 3.2 is the same for each model and is the same in the DNS data.

As can be seen in fig. 3.3, the SPMM model approximates the mixing of two scalar systems adequately, and much better than the IEM, CD and MCD model, but still is not the same as the provided DNS data.

The IEM model never performs well in flows where the jpdf evolution is studied. As stated before, the IEM model has the characteristic to be shape preserving compared to the initial pdf. This is evident in fig 3.4, where the variance is indeed decreasing but the initial shape remains. As this project studies the evolution of the joint pdf, the IEM model does not function well for this case.

The CD and MCD do not have this problem, but does tend to go faster to the mean then the DNS, as can be seen in fig. 3.4e,3.4f. It is predicted that main body of particles first mixes along the edges of the triangle, while the CD and MCD tent to go towards the total mean, i.e. the center of the triangle quicker.

The SPMM model performs in this case better. Thanks to it's conditional mean, and thus not entirely randomly chosen mixing pairs, the peaks are noticeable longer, and as a result approximate the DNS data much better. This can also be seen at the lower rms values, i.e. fig. 3.3f where the SPMM model has the triangle shape similar to the DNS data in contrast to the CD/MCD dataset.

Especially in the early stages of the jpdf, the SPMM model performs good. Eventually the model tends to converge a bit too much to the center, while the DNS data is still spread out in the (ψ_1, ψ_2) plane. A possible explanation for this problem is the initial pdf. In the used pdf, there are no particles outside the triangle. The pdf is thus discrete around the edges, while a pdf should be continuous i.e. particles can still be outside the displayed triangle but with a significantly lower chance.

4

Micromixing with mean scalar gradients

The next test case is oriented to test the rate of convergence and computational cost of mixing models. In this test case, a imposed mean scalar gradient $\beta = \frac{\partial \langle \phi \rangle}{\partial x} = 1$ is considered in a stationary, homogeneous turbulent flow.

This test case is a good flow to test the convergence of statistical moments of the thermochemical scalars. Since the additional gradient term will push the scalar away from the mean, it can be calculated how to micromixing models can reach a statistically steady state. This flow has already been studied using a variety of numerical methods, and now the SPMM will be considered. This flow has also been studied using DNS[6]. For a better comparison, also the IEM and MCD models are calculated

Due to this imposed mean scalar gradient, eq. 2.2 becomes

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = D\nabla^2\phi - \beta U. \tag{4.1}$$

Substitute this in eq. 2.1, the transport equation for the jpdf f becomes

$$\begin{aligned}\frac{\partial f}{\partial t} + \vec{U}\frac{\partial f}{\partial \vec{x}} = -\frac{\partial}{\partial \psi}\left(f\left\langle D\nabla^2\phi \middle| \vec{V}, \psi, \theta \right\rangle\right) - \frac{\partial}{\partial \theta}\left(f\left\langle \frac{\mathrm{d}\omega}{\mathrm{d}t} \middle| \vec{V}, \psi, \theta \right\rangle\right) \\ -\frac{\partial}{\partial \vec{V}}\left(f\left\langle \frac{\mathrm{d}\vec{U}}{\mathrm{d}t} \middle| \vec{V}, \psi, \theta \right\rangle\right) + \beta\vec{V}\frac{\partial f}{\partial \psi}.\end{aligned} \tag{4.2}$$

To check the computational cost for both the number of particles n_p and scalars n_s, the scalar gradient is extended in n_s dimensions. For particle i and dimension α, the stochastic differential equations become

$$\mathrm{d}x_i^\alpha = U_i^\alpha \mathrm{d}t \tag{4.3}$$

$$\mathrm{d}U_i^\alpha = -\frac{3}{4}C_0\langle\omega\rangle U_i^\alpha \mathrm{d}t + \sqrt{C_0\langle\omega\rangle k}\mathrm{d}W_t \tag{4.4}$$

$$\mathrm{d}\omega_i = -(\omega_i - \langle\omega\rangle)\frac{\mathrm{d}t}{T_\omega} + \sqrt{\frac{2\sigma_\omega^2\langle\omega\rangle\omega_i}{T_\omega}}\mathrm{d}W_t \tag{4.5}$$

$$\mathrm{d}\phi_i^\alpha = -\frac{C_\phi}{T_L}(\psi_i^\alpha - \langle\phi^\alpha|X,R\rangle)\mathrm{d}t - U_i^\alpha\beta\mathrm{d}t. \tag{4.6}$$

4.1. Numerical integration

Since the (stochastic) differential equation are mostly covered in section 3.1, we only consider the new equation 4.4. Again, assuming $\langle\omega\rangle$ is constant, we get the following numerical approximation using the Euler-Maruyama method.

$$u_{n+1} = u_n - \frac{3}{4}C_0\langle\omega\rangle u_n\Delta t + \sqrt{C_0\langle\omega\rangle k\Delta t}Z(0,1) \tag{4.7}$$

where $u_0 = U(0)$, $u_n = U(t_n)$, $t_n = n\Delta t$ and $Z(0,1)$ a randomly picked number with a normal distribution $N(0,1)$. Derived similar to eq. 3.10, $\Delta t < \frac{8}{4C_0\langle\omega\rangle} = \frac{2}{C_0\langle\omega\rangle}$. Similar to equation 3.14, we get $\Delta t = 10^{-2}s$.

4.2. Initial condition

For this test case, only 1 computational grid cell is considered. The initial conditions for this flow are displayed in table 4.1.

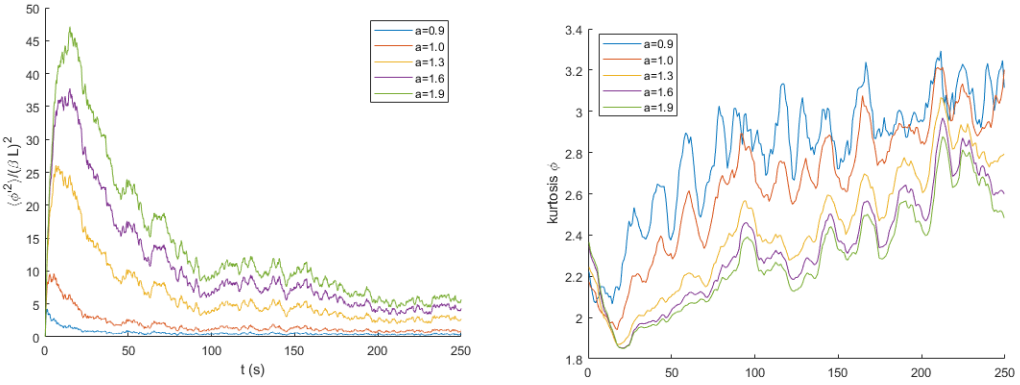
Table 4.1: Initial conditions for the flow with imposed mean gradient

$\Delta t = 10^{-2}s$	$U_i^\alpha(0) \sim N(0,1)$
$k = 577m^2/s^2$	$R_i^\alpha(0) \sim N(0,1)$
$\epsilon = 2641J/(kg \cdot s)$	$X_i^\alpha(0) \sim N(0,1)$
$\beta = 1m^{-1}$	$\omega_i(0) \sim N\left(\frac{\epsilon}{k}, \frac{\epsilon}{2k}\right) \sim N(4.577, 2.289)$
	$\phi_i^\alpha(0) \sim N(0, 10^{-6})$

4.3. Results

4.3.1. Statistics

At first, the flow is solved with different values of a and a fixed $n_s = 2$ and $n_p = 512$. The results are shown in fig. 4.1b.



(a) Normalized variance of ϕ with fixed $n_s = 2$, $n_p = 512$. The lowest line has $a = 0.9$.

(b) The kurtosis of ϕ with fixed $n_s = 2$, $n_p = 512$.

Figure 4.1: Normalized variance and kurtosis of ϕ over time, calculated by the SPMM model using different values for a ($a = 0.9, 1.0, 1.3, 1.6, 1.9$) with a fixed $n_p = 512, n_s = 2$. After $t = 200s$, a statistically stationary state is reached.

The coefficient a has an influence on the scalar statistics. The fig. 4.1a shows that higher values of a result in a higher scalar variance. This can be explained by the definition of C_ϕ in the model coefficient, eq. 2.14. As the a increases, C_ϕ decreases.

Note that C_ϕ determines the variance decay rate of a scalar, and with a lower C_ϕ the variance decay is smaller. To reach a statistically stationary state, the variance decay rate has to be equal to the variance production rate by the mean scalar gradient term βU. A higher value for a results in a lower variance decay rate, and thus results in a higher statistically stationary variance.

In fig. 4.1b, the scalar kurtosis is shown. The kurtosis is a fourth standardized moment, which gives information about the shape of the distribution. For intuition, the normal distribution has a kurtosis of 3. All odd moments (i.e. mean and skewness) are not considered here, since they remain zero over time.

Note that the mean scalar gradient ”pushes” the scalar values away from the general mean, while the micromixing model ”pulls” the scalars back to their mean. A lower C_ϕ results in less ”pull” towards

the mean, and thus a less accumulated scalar distribution. Thus lower C_ϕ leads to a lower scalar kurtosis.

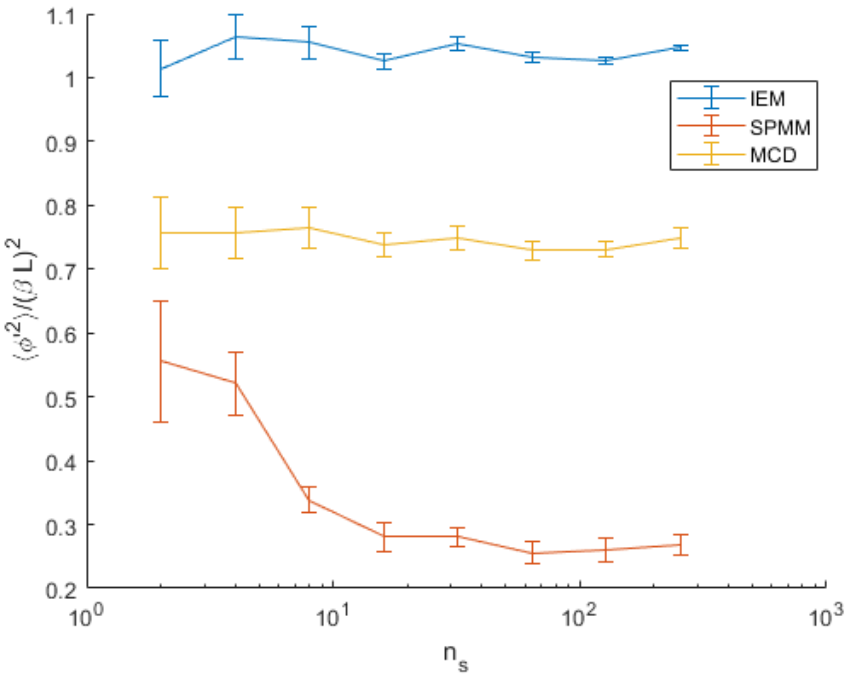


Figure 4.2: Normalized variance of ϕ as function of n_s with fixed $n_p = 512$. SPMM has a parameter $a = 1.0$.

Next, the statistically stationary variance and kurtosis are displayed as function of n_s, with a fixed $n_p = 512$. In fig. 4.2 and 4.3, the scalar variance and kurtosis calculated by the IEM or MCD model are unaffected by number of scalars. Since all scalar distribution are statistically equal, averaging of all scalars is used to give better statistical results. This is apparent in the errorbars, where the size decreases as n_s increases.

However, the SPMM has no constant variance, but has a decreasing variance for higher n_s, while the kurtosis increases. These values do tend to converge. The SPMM is affected by the number of dimensions in X and R and by construction, as n_s increases, so do the dimensions. A possible explanation for the decreasing variance, is that the chances of creating new mixing pairs increases by the number of dimensions. As the number of dimensions increases, so do the number of stochastic equations, and thus increasing the randomness of a particles positions. This increased randomness results in an a higher number of unique mixing pairs per time step. Particles which are not near neighbors in one dimension, can still become mixing pairs and mix to the mean. As a result, particles are pulled towards the mean faster, and thus decreases the variance.

The statistically stationary variance and kurtosis is also plotted as function of n_p with a fixed $n_s = 2$. As the number of particles increases, the variance increases drastically. This can be explained by the way the mean is calculated in the SPMM.

As n_p increases, the localness of the model becomes apparent. Higher n_p leads to particles being closer together in (X, R). Particles close in (X, R) have had similar velocities and thus similar mean gradients terms. As a result, the scalar values are similar and are not necessarily pulled towards the general mean, and thus contributes to a higher variance.

Note that, when n_p increases, the time it takes to reach a statistically stationary state also increases.

4.3.2. Computational cost

During the calculations, the computational cost was also determined. In the first calculation, the number of sample particles n_p increase, and a number of scalar $n_s = 2$, fig. 4.6. These calculations are perform using a unoptimized Matlab script (Appendix A.3), on a Lenovo IdeaPad G700 with Intel i7 processor.

For the IEM, CD and MCD model, the time increases linearly, since these processes have a time

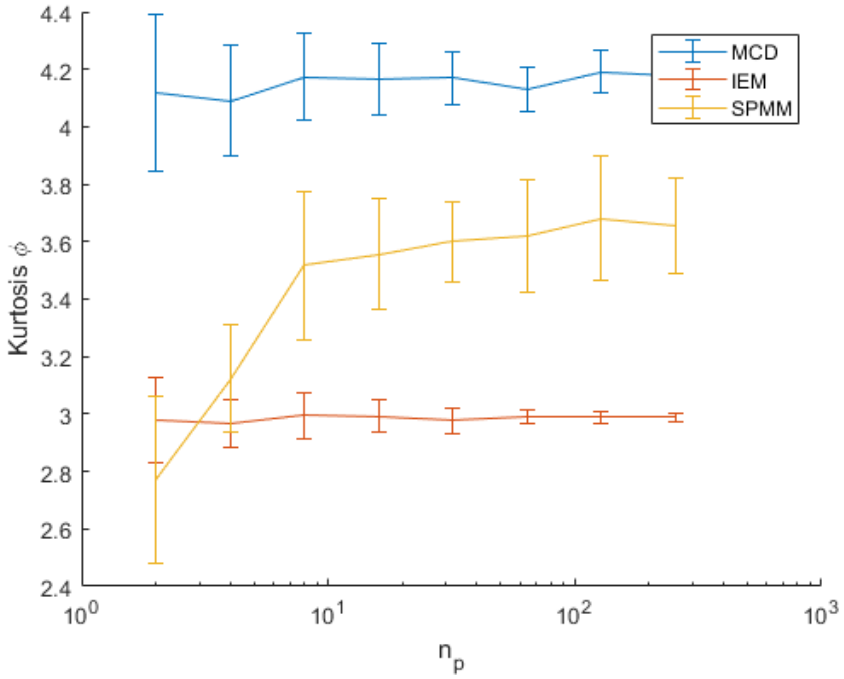


Figure 4.3: Kurtosis of ϕ as function of n_s with fixed $n_p = 512$. SPMM has a parameter $a = 1.0$.

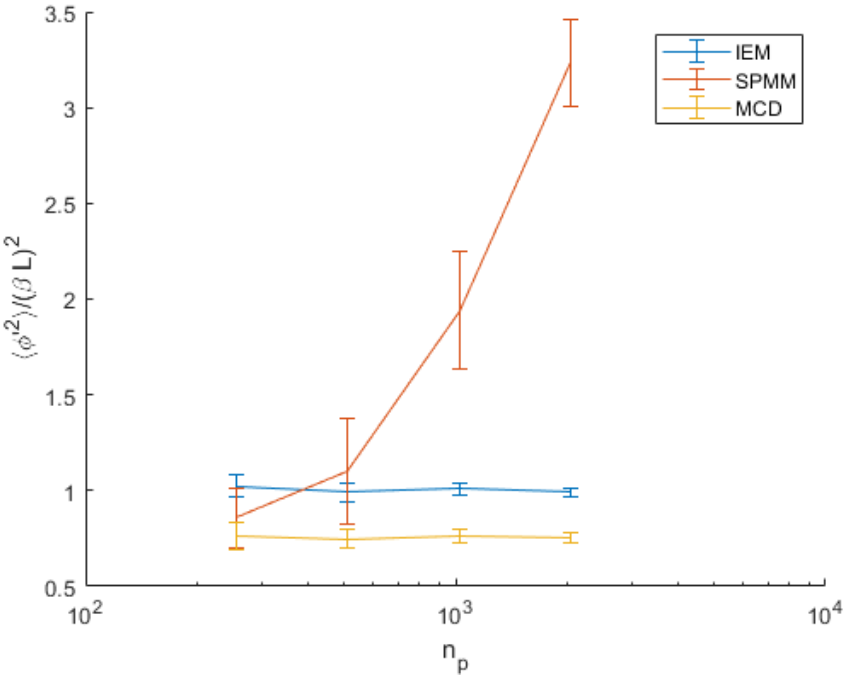


Figure 4.4: The normalized variance of ϕ as function of n_p with fixed $n_s = 2$. The SPMM coefficient is $a = 1.0$.

complexity of $O(n_p)$. However, due to the near-neighbor sorting algorithm, which has a time complexity of $O(n_p^2)$, the computational cost of the SPMM increases quadratically as the number of particles increases. This is due to the nested loop used in this sorting algorithm.

In the next calculation, the number of scalars n_s increases (and thus also the number of dimensions), and a number of particles is fixed at $n_p = 2^9$. As can be seen in fig. 4.7, the computational cost of

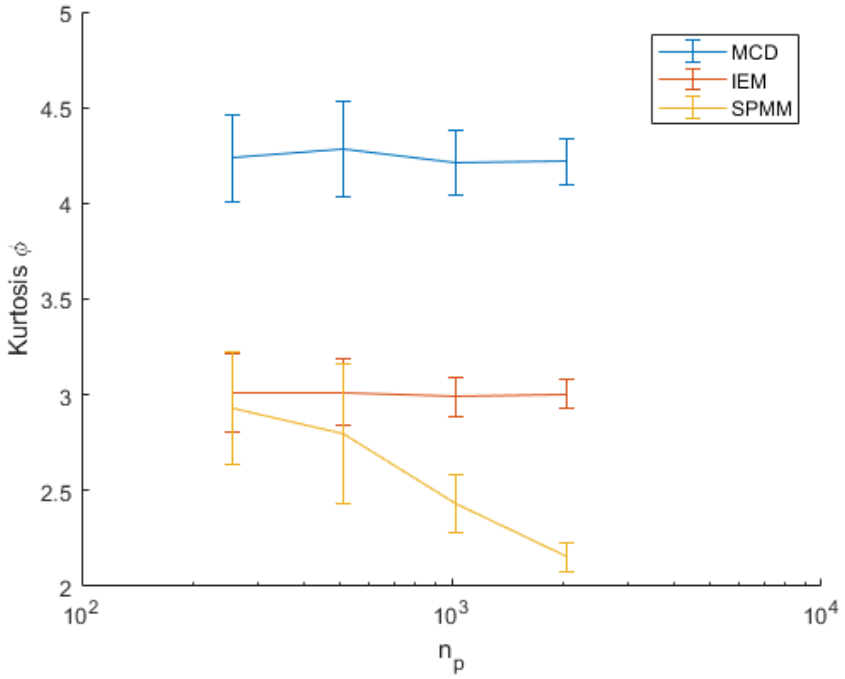


Figure 4.5: Kurtosis ϕ as function of n_p with fixed $n_s = 2$. The SPMM has a coefficient of $a = 1.0$.

each model has a dependency on n_s, but significantly lower than n_p, i.e. $O(\sqrt{n_s})$ time complexity. The only difference is the SPMM calculation time which is still significantly larger. This is mainly due to the near-neighbor sorting algorithm.

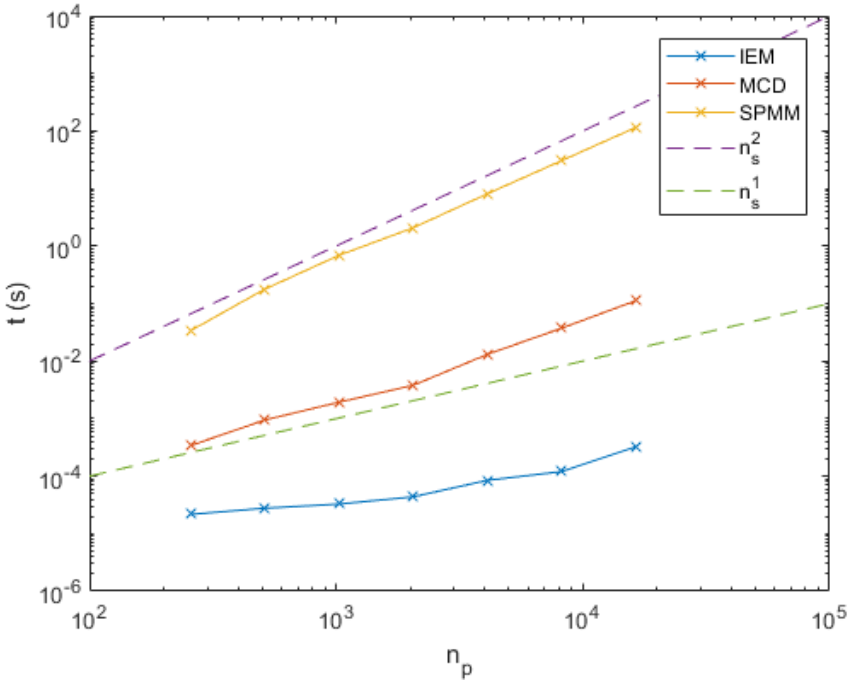


Figure 4.6: Computational cost as function of n_p, with fixed $n_s = 2$. The SPMM has a coefficient of $a = 1.0$.

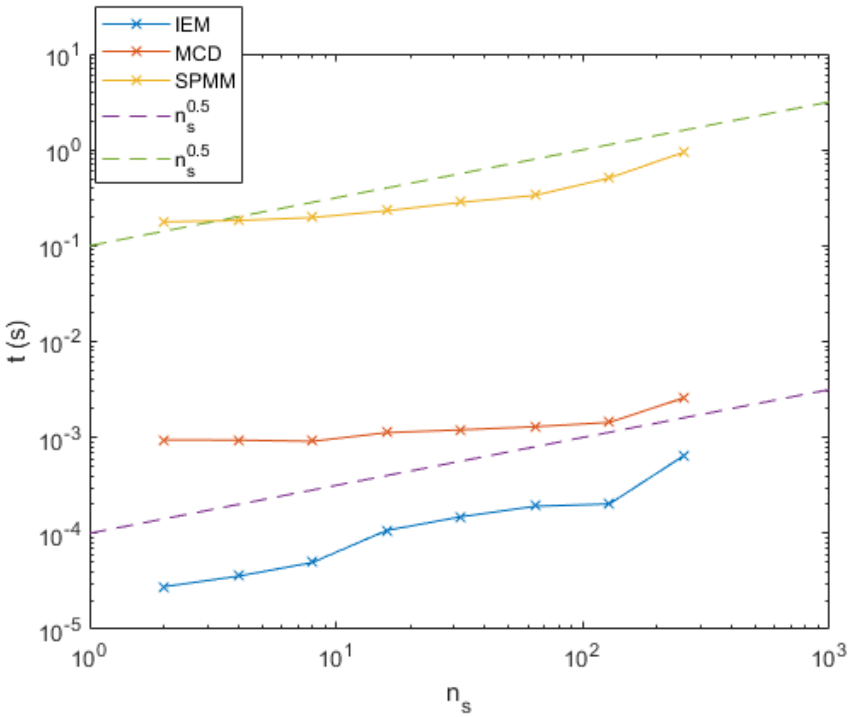


Figure 4.7: Computational cost as function of n_s with fixed $n_p = 512$. The SPMM has a coefficient of $a = 1.0$.

4.4. Discussion

From DNS calculations[6], the statistically stationary normalized variance has a value of $\langle \phi'^2 \rangle / (\beta L)^2 = 0.307$ and kurtosis of 3.171.

Comparing these values to the different micromixing models, the SPMM shows promising results. Especially as the number of scalars increases, the variance becomes similar to the DNS data and a significantly better prediction than the other evaluated models. The kurtosis is still much higher than the DNS values, but again shows better results then the other models. Note that the IEM model will always have a kurtosis of 3, since eq. 4.6 reduces to a Ornstein-Uhlbeck process which has a Gaussian solution.

However, the SPMM results are still containing flaws. This is especially apparent in fig. 4.4, where the number of particles has a big influence on the stationary variance. During this project, the statistical values can be referenced to DNS data, but n_p makes this model less reliable for stand alone predictions. For this, it is required that a model has consistent results for variables n_s and n_p. Further research on the influence of these variables has to be done, in order to reliably use this model in future projects.

In order to do practicable simulations with the SPMM model, the near-neighbor sorting algorithm has to be improved complexity wise. For larger n_p, the model becomes more reliable and physically more accurate, because the mixing time of a pair decreases. However, because of the $O(n^2)$ algorithm currently used, the computational cost for larger data sets becomes enormous. Possible improvements might be to utilize a variation of a fast EMST algorithm or sorting algorithm with $O(n \log n)$ complexity or lower.

This sorting algorithm does not have an unique solution and is dependent on the first chosen particle to calculate the remaining near neighbors. As a result, the solution of this algorithm is not unique and different mixing results are acquired. If this results in significantly different simulations is unlikely, but has to be tested qualitatively in order to exclude this.

5

Conclusion

The SPMM model approximates the three stream problem for two passive scalars qualitatively well. The jpdf evolution calculated by the SPMM evolves similarly in time as the DNS data, in contrast to the IEM, CD and MCD model. The SPMM has become a local mixing process thanks to the conditional mean in contrast to the IEM, CD and MCD model and as a result is physically more accurate.

The micromixing models, as well as the Langevin equations are numerically solved using Euler forward and the Euler-Maruyama method. Additionally, the stability condition for these integrating methods on Δt is determined.

There is however a drawback for the second case with imposed mean scalar gradient, in which closeness in (X,R) space result in unsatisfying results. As the number of particles increases, the variance and kurtosis diverge from the expected results and as a result does not reliably predict the behavior of scalar evolution in flows with an external source or gradient term. The model does show promising results compared to the provided DNS data, but clear overview of the conditions on applying the model has to be devised.

Computational costwise, the near-neighbor sorting algorithm used during these experiments is inadequate for large particle ensembles (time complexity $O(n^2)$). A better near-neighbor sorting algorithm has to be constructed to reduce complexity, such as using fast minimum spanning trees or by using a insertion or selection sorting algorithm.

A

Appendix

A.1. Joint PDF

This project concerns in the joint velocity-composition-turbulence frequency pdf $f_{U,\phi,\omega}(V,\psi,\theta;x,t)$. The derivation of this jpdf is similar to the joint velocity-composition pdf[4].

Recall that by definition, for a given quantity $Q(U,\phi,\omega)$, we have

$$\langle Q(U,\phi,\omega)\rangle = \iiint_{-\infty}^{\infty} Q(V,\psi,\theta) f_{U\phi\omega}(V,\psi,\theta;x,t)\mathrm{d}\psi\mathrm{d}V\mathrm{d}\theta \tag{A.1}$$

thus

$$\begin{aligned}\left\langle \frac{\mathrm{d}Q}{\mathrm{d}t}\right\rangle &= \frac{\partial\langle Q\rangle}{\partial t} + \frac{\partial\langle UQ\rangle}{\partial x} \\ &= \iiint_{-\infty}^{\infty} Q(V,\psi,\theta)\left[\frac{\partial f}{\partial t} + V\frac{\partial f}{\partial x}\right]\mathrm{d}\psi\mathrm{d}V\mathrm{d}\theta\end{aligned} \tag{A.2}$$

This expression can also be written with the other set of variables. Recall that

$$\frac{\mathrm{d}Q}{\mathrm{d}t} = \frac{\partial Q}{\partial V}\frac{\mathrm{d}U}{\mathrm{d}t} + \frac{\partial Q}{\partial \psi}\frac{\mathrm{d}\phi}{\mathrm{d}t} + \frac{\partial Q}{\partial \theta}\frac{\mathrm{d}\omega}{\mathrm{d}t} \tag{A.3}$$

By definition, we know that [4],[13]

$$f_{U,\phi,\omega,A}(V,\psi,\theta,a) = f_{A|U,\phi,\omega}(a|V,\psi,\theta) f_{U,\phi,\omega}(V,\psi,\theta) \tag{A.4}$$

$$\left\langle \frac{\mathrm{d}Q}{\mathrm{d}t}\right\rangle = \left\langle \frac{\partial Q}{\partial V}\frac{\mathrm{d}U}{\mathrm{d}t}\right\rangle + \left\langle \frac{\partial Q}{\partial \psi}\frac{\mathrm{d}\phi}{\mathrm{d}t}\right\rangle + \left\langle \frac{\partial Q}{\partial \theta}\frac{\mathrm{d}\omega}{\mathrm{d}t}\right\rangle \tag{A.5}$$

$$\left\langle \frac{\partial Q}{\partial V}A\right\rangle = -\iiint_{-\infty}^{\infty} Q\frac{\partial}{\partial V}\left(\langle a|V,\psi,\theta\rangle f_{U,\phi,\omega}\right)\mathrm{d}V\mathrm{d}\psi\mathrm{d}\theta \tag{A.6}$$

The second and third terms in eq. A.5 are similar to eq. A.6. Combining the equation, the new expression becomes

$$\begin{aligned}\left\langle \frac{\mathrm{d}Q}{\mathrm{d}t}\right\rangle = \iiint_{-\infty}^{\infty} Q\Bigg[&-\frac{\partial}{\partial V}\left(f\left\langle \frac{\mathrm{d}U}{\mathrm{d}t}|V,\psi,\theta\right\rangle\right) - \frac{\partial}{\partial \psi}\left(f\left\langle \frac{\mathrm{d}\phi}{\mathrm{d}t}|V,\psi,\theta\right\rangle\right) \\ &-\frac{\partial}{\partial \theta}\left(f\left\langle \frac{\mathrm{d}\omega}{\mathrm{d}t}|V,\psi,\theta\right\rangle\right)\Bigg]\mathrm{d}V\mathrm{d}\psi\mathrm{d}\theta.\end{aligned} \tag{A.7}$$

Since we have two expression for $\left\langle \frac{\mathrm{d}Q}{\mathrm{d}t} \right\rangle$ which have to be equal to each other, we get an partial differential equation for jpdf $f_{U,\phi,\omega}(V,\psi,\theta)$, which is

$$\begin{aligned}\frac{\partial f}{\partial t} + U\frac{\partial f}{\partial x} = -\frac{\partial}{\partial \psi}\left(f\left\langle \frac{\mathrm{d}\phi}{\mathrm{d}t}|V,\psi,\theta \right\rangle\right) - \frac{\partial}{\partial \theta}\left(f\left\langle \frac{\mathrm{d}\omega}{\mathrm{d}t}|V,\psi,\theta \right\rangle\right) \\ -\frac{\partial}{\partial V}\left(f\left\langle \frac{\mathrm{d}U}{\mathrm{d}t}|V,\psi,\theta \right\rangle\right)\end{aligned} \tag{A.8}$$

A.2. More stable integration scheme

Both the IEM and the SPMM can be written in the form of

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = -\frac{1}{2}C\left\langle \omega \right\rangle(\phi - \left\langle \phi \right\rangle), \tag{A.9}$$

where the coefficient C and the mean are different. One way to integrate this scheme is using Euler forward, which has the following approximation[1],

$$\phi_{n+1} = \phi_n - \frac{1}{2}C\left\langle \omega \right\rangle(\phi_n - \left\langle \phi \right\rangle)\Delta t. \tag{A.10}$$

However, this approximation is only stable if

$$|1 - \frac{1}{2}C\left\langle \omega \right\rangle \Delta t| < 1 \Rightarrow \Delta t = \frac{4}{C\left\langle \omega \right\rangle}. \tag{A.11}$$

By solving the eq. A.9 first, this scheme can become exact. Define $\phi' \equiv \phi - \left\langle \phi \right\rangle$ and note that in both section 3 and 4 $\left\langle \phi \right\rangle = 0$, then

$$\frac{\mathrm{d}\phi'}{\mathrm{d}t} \equiv \frac{\mathrm{d}(\phi - \left\langle \phi \right\rangle)}{\mathrm{d}t} = \frac{\mathrm{d}\phi}{\mathrm{d}t} = -\frac{1}{2}C\left\langle \omega \right\rangle \phi', \tag{A.12}$$

thus solving for ϕ', it reduces to

$$\phi'(t) = A\exp\left(-\frac{1}{2}C\left\langle \omega \right\rangle t\right). \tag{A.13}$$

To remove the constant A, we divide values of ϕ' for different times. Define $t_n = n\Delta t$, where $\Delta t > 0$, then

$$\frac{\phi'_{n+1}}{\phi'_n} = \frac{\phi'(t_{n+1})}{\phi'(t_n)} = \exp\left(-\frac{1}{2}C\left\langle \omega \right\rangle \Delta t\right). \tag{A.14}$$

Using the definition of ϕ', it rewrites to

$$\phi_{n+1} - \left\langle \phi \right\rangle = (\phi_n - \left\langle \phi \right\rangle)\exp\left(-\frac{1}{2}C\left\langle \omega \right\rangle \Delta t\right), \tag{A.15}$$

which rewritten as a general integration scheme becomes

$$\phi_{n+1} = \phi_n + (\phi_n - \left\langle \phi \right\rangle)\left[\exp\left(-\frac{1}{2}C\left\langle \omega \right\rangle \Delta t\right) - 1\right]. \tag{A.16}$$

Define the error of approximation ϕ_n as $\epsilon_n = \phi_n - \tilde{\phi}_n$, where $\tilde{\phi}_n$ is the actual value of ϕ at t_n. This error for eq. A.16 becomes

$$\epsilon_{n+1} = \exp\left(-\frac{1}{2}C\left\langle \omega \right\rangle \Delta t\right)\epsilon_n. \tag{A.17}$$

To become absolute stable , $|\epsilon_{n+1}| \leq |\epsilon_n|$, thus (since $C, \left\langle \omega \right\rangle > 0$)

$$|\exp\left(-\frac{1}{2}C\left\langle \omega \right\rangle \Delta t\right)| < 1 \Rightarrow \Delta t > 0. \tag{A.18}$$

Which is clear since we actually solved eq. A.9 exact in A.13.

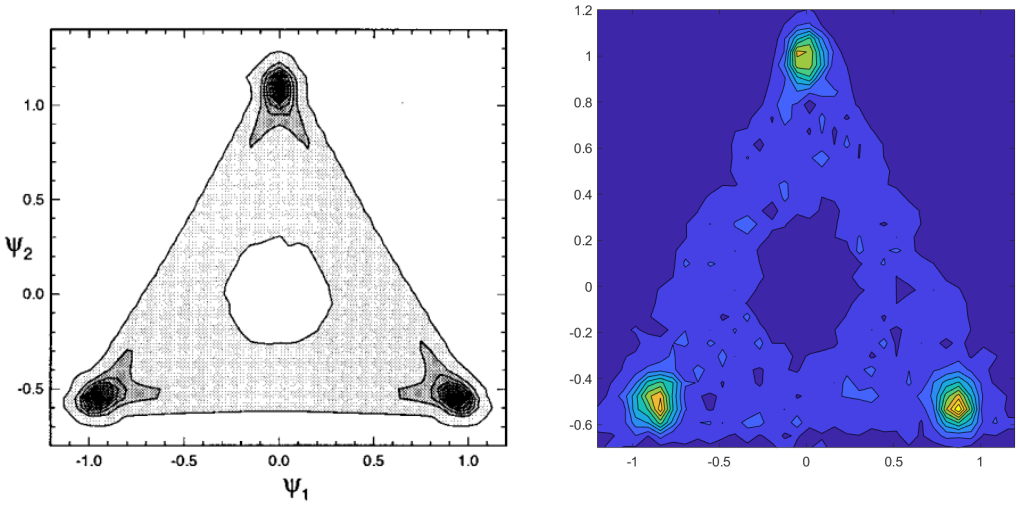
A.3. Near-Neighbor Algorithm

Using Matlab to calculate the development of the jpdf f, a important part of the SPMM model, is the near-neighbor sorting algorithm. Below, the Matlab code is given to sort the particle ensamble.

```
%% A near neigbor sort function to calculate wich particles are phi_{i-1} and
    phi_{i+1}
% input: the position X and the shadow position R of all sample particles
% output: sorted index of all particles
function index = nnsort(X,R)
  N = size(X,1); %Number of particles
  index = (1:N); %Assuming the particles are already sorted correct
  for i = 1:(N-1)
    bestit = 0;
    bestdist = Inf;
%Particles 1 to i are already sorted correctly
%Calculates for particles i+1 to N, which particle is closest to i.
    for j = (i+1):N
      distij = sum((X(index(i),:)-X(index(j),:)).^2)
          +sum((R(index(i),:)-R(index(j),:)).^2);
      if(distij < bestdist)
        bestdist = distij;
        bestit = j;
      end
    end
    %Swap the closest partcicle to i, with parcticle i+1
    if(bestit ~= 0)
      index([bestit (i+1)]) = index([(i+1) bestit]);
    end
  end
end
```

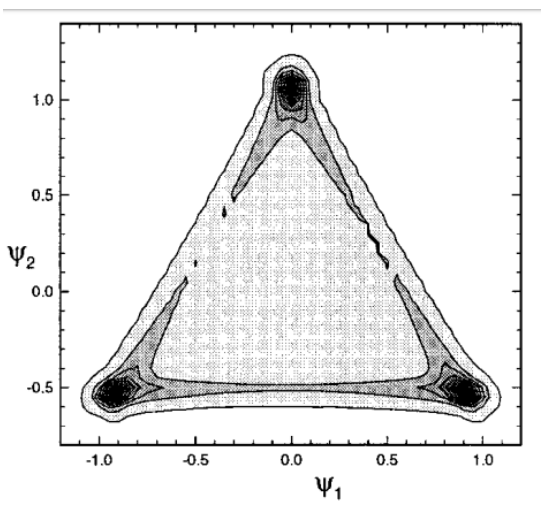
A.4. Three Stream data

Below, the jpdf for the three stream mixing flow, described in section 3.1, is displayed using the DNS data[5] as well as the SPMM calculated data with $a = 0.87, 1, 2, 3$.

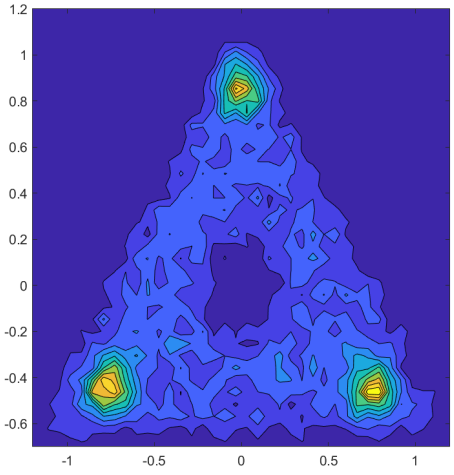


(a) DNS, $\frac{\phi''(t)}{\phi''(0)} = 1.0$ (b) a=6, $\frac{\phi''(t)}{\phi''(0)} = 1.0$

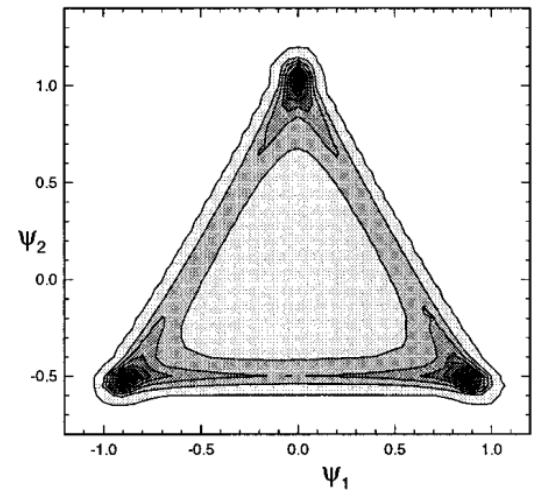
Figure A.1: The jpdf f at $\frac{\phi''(t)}{\phi''(0)} = 1.0$. Left shows the provided DNS figure, and right the calculated SPMM with $a = 6.0$.



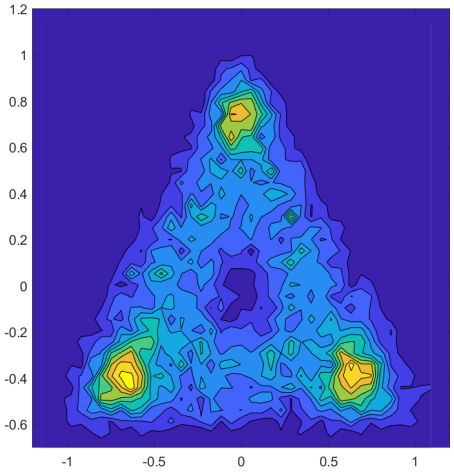
(a) DNS, $\frac{\phi''(t)}{\phi''(0)} = 0.9$



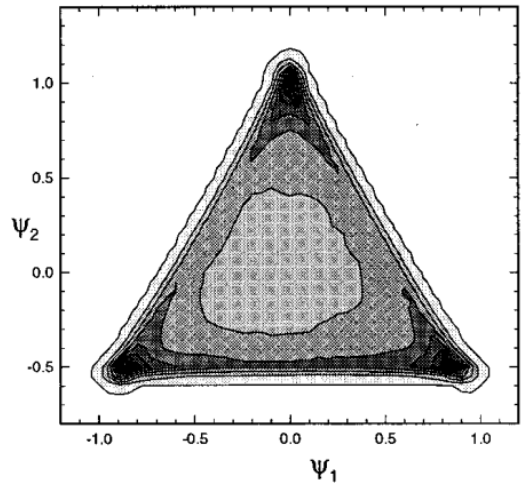
(b) a=6, $\frac{\phi''(t)}{\phi''(0)} = 0.9$



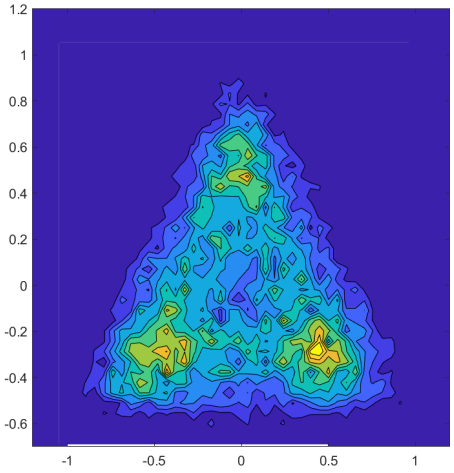
(c) DNS, $\frac{\phi''(t)}{\phi''(0)} = 0.8$



(d) a=6, $\frac{\phi''(t)}{\phi''(0)} = 0.8$

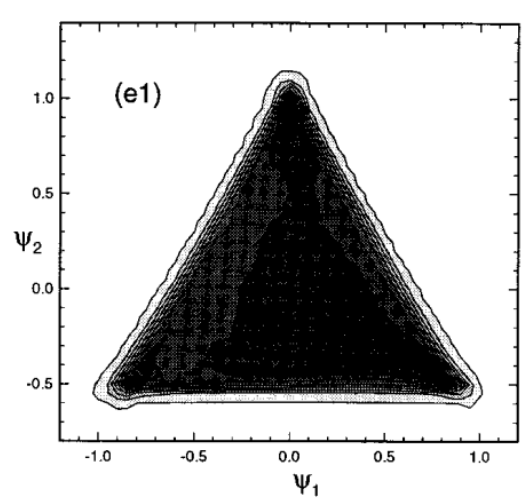


(e) DNS, $\frac{\phi''(t)}{\phi''(0)} = 0.7$

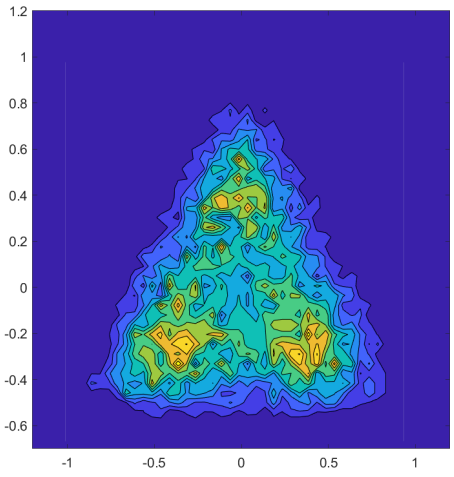


(f) a=6, $\frac{\phi''(t)}{\phi''(0)} = 0.7$

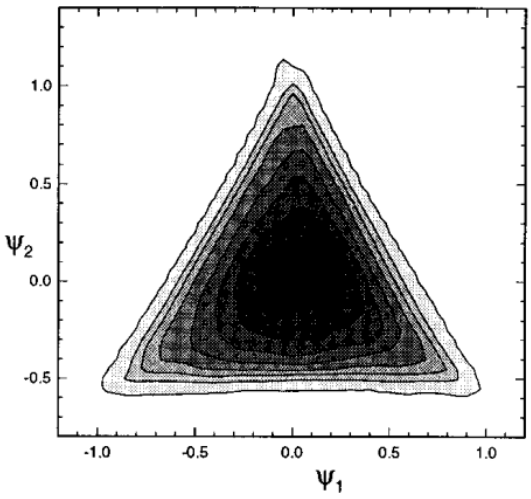
Figure A.2: The jpdf f at $\frac{\phi''(t)}{\phi''(0)} = 0.9, 0.8, 0.7$. Left shows the provided DNS figure, and right the calculated SPMM with $a = 6.0$.



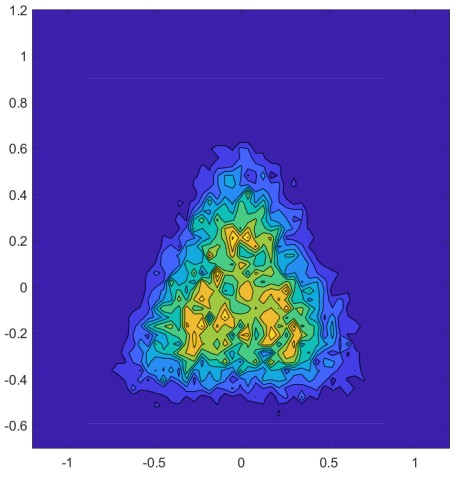
(a) DNS, $\frac{\phi''(t)}{\phi''(0)} = 0.6$



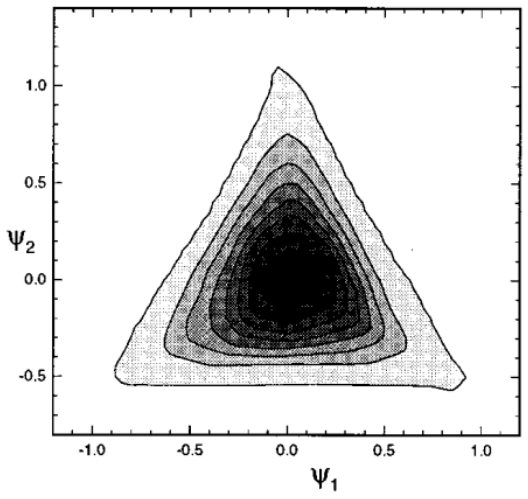
(b) a=6, $\frac{\phi''(t)}{\phi''(0)} = 0.6$



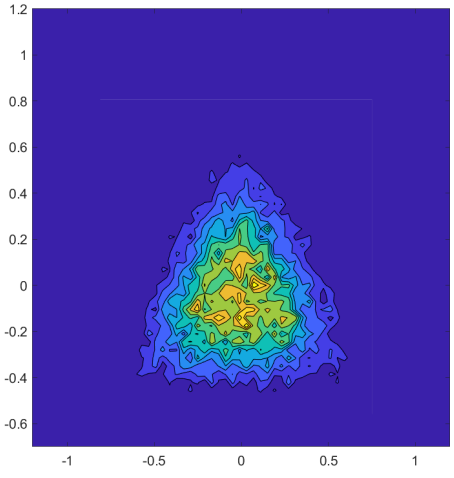
(c) DNS, $\frac{\phi''(t)}{\phi''(0)} = 0.5$



(d) a=6, $\frac{\phi''(t)}{\phi''(0)} = 0.5$

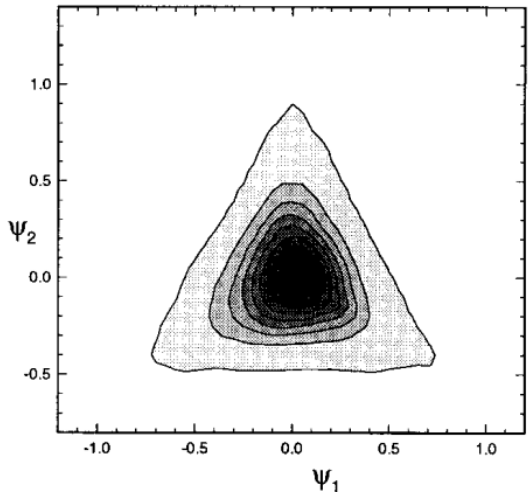


(e) DNS, $\frac{\phi''(t)}{\phi''(0)} = 0.4$

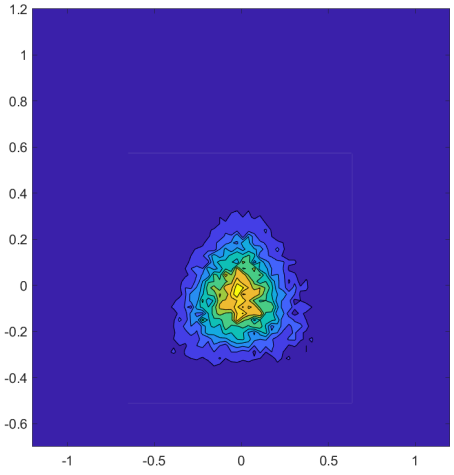


(f) a=6, $\frac{\phi''(t)}{\phi''(0)} = 0.4$

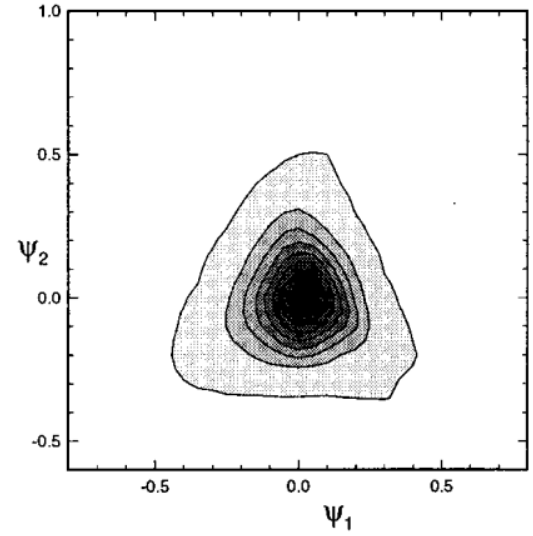
Figure A.3: The jpdf f at $\frac{\phi''(t)}{\phi''(0)} = 0.6, 0.5, 0.4$. Left shows the provided DNS figure, and right the calculated SPMM with $a = 6.0$.



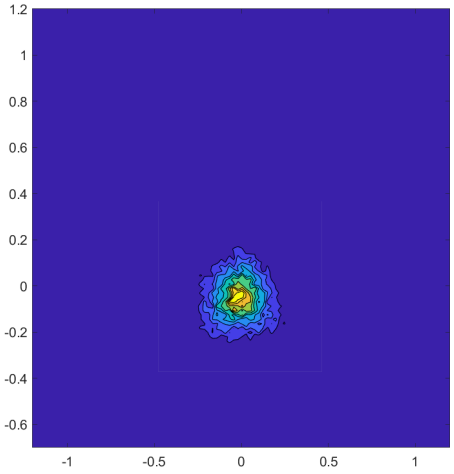
(a) DNS, $\frac{\phi''(t)}{\phi''(0)} = 0.3$



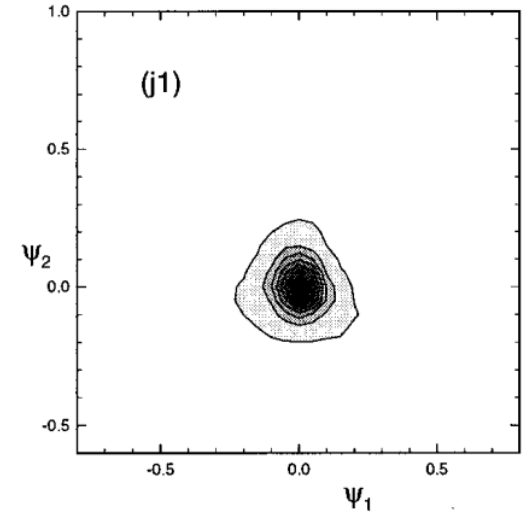
(b) a=6, $\frac{\phi''(t)}{\phi''(0)} = 0.3$



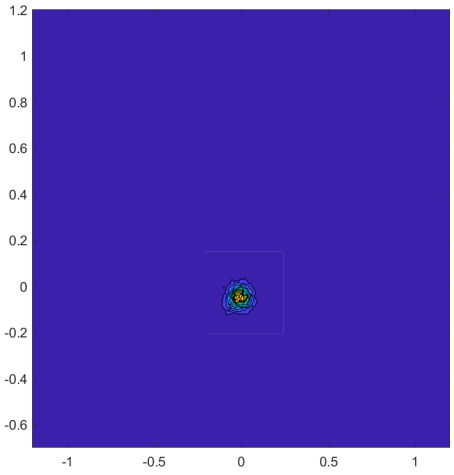
(c) DNS, $\frac{\phi''(t)}{\phi''(0)} = 0.2$



(d) a=6, $\frac{\phi''(t)}{\phi''(0)} = 0.2$



(e) DNS, $\frac{\phi''(t)}{\phi''(0)} = 0.1$



(f) a=6, $\frac{\phi''(t)}{\phi''(0)} = 0.1$

Figure A.4: The jpdf f at $\frac{\phi''(t)}{\phi''(0)} = 0.3, 0.2, 0.1$. Left shows the provided DNS figure, and right the calculated SPMM with $a = 6.0$.

Bibliography

[1] C. Vuik, F.J. Vermolen, M.B. van Gijzen, M.J. Vuik *Numerical methods for ordinary differential equations*, Delft Academic Press, 2015.

[2] S.B. Pope *A model for turbulent mixing based on shadow-position conditioning*, Physics of fluids 25:11, 110803, 2013.

[3] D.W. Meyer, P. Jenny *Micromixing models for turbulent flows*, Journal of computational physics 4, 1275-1293, 2008.

[4] S.B. Pope *Turbulent Flows*, Cambridge university press, 2000.

[5] A. Juneja, S.B. Pope *A DNS study of turbulent mixing of two passive scalars*, Physics of Fluids 8:8, 2161-2184, 1996.

[6] M. R. Overholt, S. B. Pope *Direct numerical simulation of a passive scalar with imposed mean gradient in isotropic turbulence*, Physics of Fluids 8:11, 3128-3148 1996.

[7] D. Roekaerts, T. Peeters *Turbulente reagerende stromingen*, TU Delft, 1997.

[8] MATLAB 2017b *The Mathworks, Inc* Massachussets, United States, 2017.

[9] S. Subramaniam, S.B. Pope *A mixing model for turbulent reactive flows based on euclidean minimum spanning trees* Combustion an flame 115:487-514 1998

[10] D.W. Meyer, P. Jenny *A mixing model for turbulent flows based on parameterized scalar profiles* Physics of fluids 18:3, 2006

[11] D.T. Gillespie *Exact numerical simulation of the Ornstein-Uhlenbeck process and its integral* Naval Air Warfare Center, California 1995

[12] D.J. Higham, X. Mao *Convergence of Monte Carlo simulations involving the mean-reverting square root process*, Journal of Computational Finance 8 (3). pp. 35-61, 2005.

[13] J.A. Rice *Mathematical statistics and data analysis* Belmont, CA, Brooks/Cole 2007

[14] M. Koene *Evaluation of Micromixing Models for Turbulent Mixing in Supercritical Water* Delft, 2018

[15] G. I. Taylor *Diffusion by continuous movements* Proc. London Math. Soc. s2-20, 196–212, 1922

[16] S.B. Pope *Lagrangian pdf methods for turbulent flows* Cornell University, Ithaca, New York, 1994

[17] S. Corrsin, *Heat Transfer in Isotropic Turbulence*, Journal of Applied Physics 23:1, 113-118, 1952