Wasatch Documentation

Tony Saad, James C. Sutherland, Amir Biglari, Alex W. Abboud

October 1, 2012

Wasatch is a finite volume computational fluid dynamics (CFD) code that employs the spatial operator (SpatialOps) and expression (ExprLib) libraries at its foundation. It is developed as a software component within the Uintah computational parallel framework. Wasatch provides the interface between the Uintah API and the various classes and tools supplied by the SpatialOps and ExprLib libraries making its development entirely based on the expressions graph abstraction.

Contents

Chapter 1

Capabilities

Wasatch currently supports basic physical models of transport phenomena in fluids and gases. These may be reactive with variable density or non-reactive. The working model is based on writing expressions that reflect physical processes, akin to mathematical terms in a partial differential equation (PDE). These expressions are subsequently used in a variety of transport equations as needed. Transport equations are derived from a base class and each specializes in solving a particular class of problems. Currently, Wasatch currently supports the following transport equations with constant density:

- Generic scalar transport
- Momentum transport
- Population balances via moment transport

A wide range of expressions are currently implemented in Wasatch. These include the standard convective and diffusive fluxes as well as stresses as well as a variety of source terms that are inherent from specific physical models such as precipitation among others.

1.1 Momentum Transport

Wasatch solves the variable density momentum equations

$$\frac{\partial \rho \mathbf{u}}{\partial t} = -\nabla \cdot \rho \mathbf{u}\mathbf{u} - \nabla \cdot \boldsymbol{\tau}_{ij} - \nabla p + \rho \mathbf{g} \tag{1.1}$$

in conjunction with the continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0. \tag{1.2}$$

Here, ρ stands for the density, $\mathbf{u} = (u, v, w)$ is the velocity field, p is the pressure, $\mathbf{g}$ is the gravity vector, and $\boldsymbol{\tau}_{ij}$ is the stress tensor given by

$$\boldsymbol{\tau}_{ij} = -2\mu S_{ij} + \frac{2}{3}\mu \delta_{ij} S_{kk}; \quad S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{1.3}$$

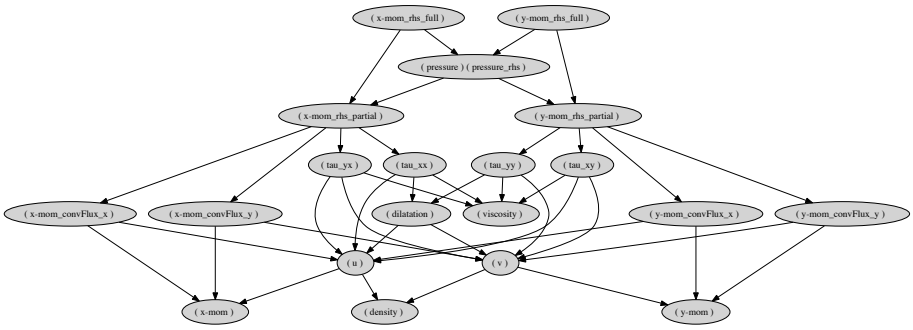


Figure 1.1: Expression graph for a two dimensional fluid flow problem using the Poisson equation for the pressure.

The rate of strain tensor S_{ij} is a fundamental quantity when using gradient diffusion turbulence models and will be used extensively in subsequent sections.

The code supports central, upwind, and over ten flux limiters for the discretization of the convective flux. The pressure is resolved by solving the pressure Poisson equation (PPE). This is constructed by first taking the divergence of the momentum equations. Then the continuity equation is used to substitute for the temporal term in the resulting equation. At the outset, one recovers a Poisson equation for the pressure

$$\nabla^2 p = \frac{\partial^2 \rho}{\partial t^2} + \nabla \cdot \nabla \cdot \rho \mathbf{u}\mathbf{u} + \nabla \cdot \nabla \cdot \boldsymbol{\tau}_{ij} - \nabla \cdot \rho \mathbf{g}. \tag{1.4}$$

The solution of Eq. (1.4) requires the inversion of a linear system of equations. At the time of writing, Wasatch can solve the constant density momentum equations in three dimensions with periodic boundary conditions. The solution is based on the expressions graph showing that a basic CFD algorithm may be achieved using the graph based algorithm approach. A sample graph for a two dimensional fluid flow is shown in Fig. 1.1.

Pressure Projection Wasatch uses the Pressure-Poisson-Equation (PPE) formulation to handle the pressure-velocity couple. The PPE formulation consists of first taking the divergence of the momentum equations and substituting the continuity equation to yield a Poisson equation for the pressure. This version of the pressure projection provides an elegant and straight-forward equation for the pressure. Part of the difficulty of using the PPE formulation resides in the specification of appropriate boundary and initial conditions. While boundary conditions correspond to "natural" conditions imposed on physical quantities such as mass flowrate, pressure gradient, and velocity; specification of the initial conditions requires further attention on the Pressure poisson equation.

Constant Density For a constant density flow, the Pressure-Poisson-Equation (1.4) reduces to

$$\nabla^2 p = \nabla \cdot \nabla \cdot \rho \mathbf{u}\mathbf{u} + \nabla \cdot \nabla \cdot \boldsymbol{\tau}_{ij} - \nabla \cdot \rho \mathbf{g}. \tag{1.5}$$

1.1.1 LES Modeling

Wasatch uses Favre filtering as the basis for its LES approach. Favre filtering is defined as a density-weighted filtering operation to accomodate variable density. It is given by the following operation

$$\overline{\rho\phi} = \overline{\rho}\widetilde{\phi}, \tag{1.6}$$

where the overbar denotes a traditional filtering operation. A traditional filtering operation consists of a convolution of a transported variable (i.e. a variable that one wishes to filter) with a filter. This is written as

$$\overline{\phi} = \int_{\Omega} G(\mathbf{r})\phi(\mathbf{x}-\mathbf{r})\,\mathrm{d}\mathbf{r}. \tag{1.7}$$

Filtered Momentum Equations To make further headway, we apply the traditional filter to the governing fluid transport equations. Starting with continuity, we have

$$\frac{\partial\overline{\rho}}{\partial t} + \nabla\cdot\overline{\rho\mathbf{u}} = 0. \tag{1.8}$$

Then, one makes use of the Favre filtering operation by setting $\overline{\rho\mathbf{u}} = \overline{\rho}\widetilde{\mathbf{u}}$. At the outset, the Favre-filtered continuity equation is given by

$$\frac{\partial\overline{\rho}}{\partial t} + \nabla\cdot\overline{\rho}\widetilde{\mathbf{u}} = 0. \tag{1.9}$$

Moving on to the momentum equation, the filtering operation yields

$$\frac{\partial\overline{\rho\mathbf{u}}}{\partial t} = -\nabla\cdot\overline{\rho\mathbf{u}\mathbf{u}} - \nabla\cdot\overline{\tau_{ij}} - \nabla\overline{p} + \overline{\rho\mathbf{g}}. \tag{1.10}$$

It is best to transform to Favre-filtered terms one at a time. This leads to the following

- $\frac{\partial\overline{\rho\mathbf{u}}}{\partial t} = \frac{\partial\overline{\rho}\widetilde{\mathbf{u}}}{\partial t}$
- $\nabla\cdot\overline{\rho\mathbf{u}\mathbf{u}} = \nabla\cdot\overline{\rho}\widetilde{\mathbf{u}\mathbf{u}} = \nabla\cdot\overline{\rho}\widetilde{\mathbf{u}}\widetilde{\mathbf{u}} + (\nabla\cdot\overline{\rho}\widetilde{\mathbf{u}}\widetilde{\mathbf{u}} - \nabla\cdot\overline{\rho}\widetilde{\mathbf{u}}\widetilde{\mathbf{u}}) \equiv \nabla\cdot\overline{\rho}\widetilde{\mathbf{u}}\widetilde{\mathbf{u}} + \nabla\cdot\boldsymbol{\tau}^R_{ij}; \quad \boldsymbol{\tau}^R_{ij} \equiv \overline{\rho}\widetilde{\mathbf{u}}\widetilde{\mathbf{u}} - \overline{\rho}\widetilde{\mathbf{u}}\widetilde{\mathbf{u}}$
- $\nabla\cdot\overline{\tau_{ij}} = \nabla\cdot\widetilde{\tau_{ij}} + \left(\nabla\cdot\overline{\tau_{ij}} - \nabla\cdot\widetilde{\tau_{ij}}\right) \equiv \nabla\cdot\widetilde{\tau_{ij}} + \nabla\cdot\boldsymbol{\tau}^L_{ij}$
- $\overline{\rho\mathbf{g}} = \overline{\rho}\mathbf{g}$

where $\boldsymbol{\tau}^R_{ij}$ and $\boldsymbol{\tau}^L_{ij}$ are subgrid-scale stress tensors (SGS) that result from applying filtering to the convective and diffusive terms in the momentum equations, respectively. At the outset, the filtered momentum equations may be written as

$$\frac{\partial\overline{\rho}\widetilde{\mathbf{u}}}{\partial t} = -\nabla\cdot\overline{\rho}\widetilde{\mathbf{u}}\widetilde{\mathbf{u}} - \nabla\cdot\widetilde{\tau_{ij}} - \nabla\overline{p} - \nabla\cdot\boldsymbol{\tau}^R_{ij} - \nabla\cdot\underbrace{\boldsymbol{\tau}^L_{ij}}_{\text{neglect}} + \overline{\rho\mathbf{g}}. \tag{1.11}$$

It is evident from Eq. (1.11) that both subgrid tensors $\boldsymbol{\tau}^R_{ij}$ and $\boldsymbol{\tau}^L_{ij}$ require closure. However, it may be argued that the subgrid tensor $\boldsymbol{\tau}^L_{ij}$ resulting from the viscous terms is negligible compared

to $\boldsymbol{\tau}_{ij}^R$ that results from filtering of the convective terms (Geurts et al., 1993; Vreman et al., 1995). Hence, the resulting filtered momentum equations take the form

$$\frac{\partial \overline{\rho}\widetilde{\mathbf{u}}}{\partial t} = -\nabla \cdot \overline{\rho}\widetilde{\mathbf{u}}\widetilde{\mathbf{u}} - \nabla \cdot \widetilde{\boldsymbol{\tau}_{ij}} - \nabla \overline{p} - \nabla \cdot \boldsymbol{\tau}_{ij}^R + \overline{\rho}\mathbf{g}. \tag{1.12}$$

The subgrid stress tensor τ_{ij}^R requires closure as it introduces unknown quantities. The subject of closure is at the heart of the LES method and will be discussed in the next subsection.

1.1.1.1 Subgrid Closure

The most common approach to modeling the subgrid-scale tensors is based on eddy-viscosity models. First propsed by Smagorinsky (1963), these have the general form

$$\boldsymbol{\tau}_{ij}^R = \overbrace{\underbrace{\boldsymbol{\tau}_{ij}^R - \tfrac{1}{3}\delta_{ij}\boldsymbol{\tau}_{kk}^R}_{\text{deviatoric}}}^{\boldsymbol{\tau}_{ij}^{\mathrm{d}}} + \overbrace{\underbrace{\tfrac{1}{3}\delta_{ij}\boldsymbol{\tau}_{kk}^R}_{\text{isotropic}}}^{\boldsymbol{\tau}_{ij}^{\mathrm{s}}}. \tag{1.13}$$

The deviatoric subgrid-stress tensor $\boldsymbol{\tau}_{ij}^{\mathrm{d}}$ is modeled as

$$\boldsymbol{\tau}_{ij}^{\mathrm{d}} = 2\mu_{\mathrm{t}}\left(\widetilde{S}_{ij} - \tfrac{1}{3}\delta_{ij}\widetilde{S}_{kk}\right), \tag{1.14}$$

where ν_{t} is the eddy- or turbulent-viscosity and $\widetilde{S}_{ij}$ is the Favre-filtered rate of strain - a computable quantity.

The isotropic subgrid-stress tensor $\boldsymbol{\tau}_{ij}^{\mathrm{s}}$ may be absorbed into the filtered pressure or entirely neglected (Erlebacher et al., 1992; Garnier et al., 2009). This is based on the observation that small scales are generally incompressible (Garnier et al., 2009). In fact, $\boldsymbol{\tau}_{kk}$ may be written in terms of the subgrid Mach number as

$$\boldsymbol{\tau}_{kk} = \gamma M_{\mathrm{sgs}}^2 \overline{p}. \tag{1.15}$$

where γ is the ratio of specific heats and M_{sgs} is the subgrid Mach number whose values is expected to be small when the turbulent Mach number of the large scales is low.

Upon substitution of Eq. (1.13) into Eq. (1.11) one recovers

$$\frac{\partial \overline{\rho}\widetilde{\mathbf{u}}}{\partial t} = -\nabla \cdot \overline{\rho}\widetilde{\mathbf{u}}\widetilde{\mathbf{u}} - \nabla \cdot \widetilde{\boldsymbol{\tau}_{ij}} - \nabla \overline{p} - \nabla \cdot \boldsymbol{\tau}_{ij}^{\mathrm{d}} + \overline{\rho}\mathbf{g}. \tag{1.16}$$

Given that all quantities – except ν_{t} – are computable, the essence of all eddy-viscosity models is then to model the turbulent viscosity ν_{t}.

Constant Smagorinsky-Lilly Model In the constant Smagorinsky-Lilly model, a simple algebraic relation is proposed for the eddy-viscosity (Smagorinsky, 1963). This form may be written as

$$\mu_{\mathrm{t}} = \overline{\rho}\left(C_{\mathrm{s}}\Delta\right)^2 |\widetilde{S}| = \overline{\rho} L_s^2 |\widetilde{S}|,$$

where C_{s} is a constant, Δ is an average measure of the cell size, usually taken as the average cell volume $V^{1/3}$, and $|\widetilde{S}| = \sqrt{2\widetilde{S}_{ij}\widetilde{S}_{ij}}$. Also, L_s is commonly referred to as the mixing length and is a measure of the characteristic mixing length of turbulent eddies.

In Wasatch, we implement a variant of Smagorinsky's classic model. To allow users to fine-tune the turbulent viscosity based on physical input, we introduce a lengthscale to the calculation of the smagorinsky mixing length. The desired features of our model are

- As the grid spacing Δ approaches the kolmogorov scale, the turbulent viscosity should approach zero since all scales are resolved.
- As the grid spacing drifts from the kolmogorov scale, the effect of mixing due to small scales becomes more pronounced and hence the turbulent viscosity increases.
- Ability to recover constant smagorinsky model when desirable.

These features may be accomplished by setting

$$L_s = C_s \Delta \left(1 - \frac{\Delta}{\eta}\right). \tag{1.17}$$

Here, C_s is a scaling constant that is *not* equal to the Smagorinsky constant and η is a characteristic small scale of the problem. Typically, η may be taken as the Kolmogorov scale. It may be evalulated from the following scaling law

$$\frac{L}{\eta} \sim \mathrm{Re}^{3/4}, \tag{1.18}$$

where L is a characteristic length of the large scale eddies and Re is the Reynolds number of the flow. For example, for the flow in a pipe of diameter D, the characteristic length may be taken as $L = D$ while $\mathrm{Re} = \frac{\rho V D}{\mu}$. The behavior of Eq. (1.17) is shown in Fig. 1.2. Two key regimes may be identified in this figure. The first lies at $0 < \eta < \Delta$ (to the left of $\eta = \Delta$ mark) designates the region where the mesh spacing is larger than the small scales. Hence, as the mesh spacing becomes larger than the small scales (i.e. only very large scales are resolved), more dissipation must be added to mimic the dissipation caused by the small scales. The second regime lies in the region $\Delta < \eta < \infty$ and is only included for mathematical completion. This region indicates that the mesh size is smaller than the smallest scales and is not truly part of an LES model. Its inclusion, however, is necessary to recover the Smagorinsky classic model as $\eta \to \infty$.

Dynamic Smagorinsky Model The most important difference between filtered and non-filtered momentum equation is in the diffusive term, τ_{kl}. According to the Smagorinsky eddy viscosity model it can be modeled as

$$\tau_{kl} - \frac{1}{3} q^2 \delta_{kl} = -2C\bar{\rho}\Delta^2 |\tilde{S}| (\tilde{S}_{kl} - \frac{1}{3}\tilde{S}_{mm}\delta_{kl}), \tag{1.19}$$

where $\tilde{S}_{kl} = \frac{1}{2}(\partial \tilde{u}_k/\partial x_l + \partial \tilde{u}_l/\partial x_k)$, $|\tilde{S}| = (2\tilde{S}_{kl}\tilde{S}_{kl})^{1/2}$ and $q^2 = \tau_{ii}$. q^2 is also parameterized using Yoshizawa's model to obtain

$$q^2 = 2C_I \bar{\rho} \Delta^2 |\tilde{S}|^2, \tag{1.20}$$

where Δ is the grid size.

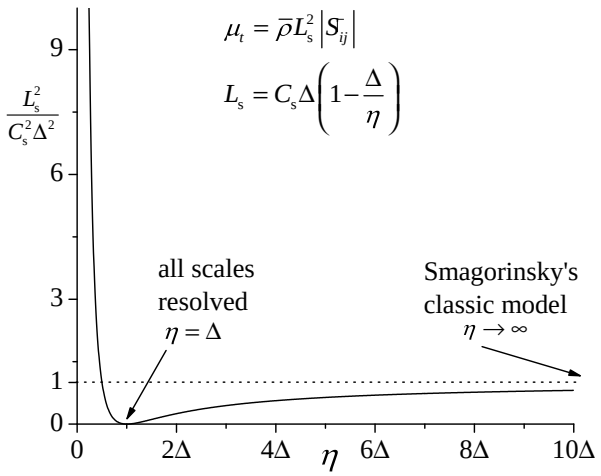


Figure 1.2: Behavior of the mixing length in the modified Smagorinsky model used in Wasatch.

C and C_I are Smagorinsky constants which can be assumed to be a fixed number and also can be calculated using dynamic models. By using dynamic models and applying test-filters these constants are parameterized as

$$C_I = \frac{< \widehat{\bar{\rho}\tilde{v}_k \tilde{v}_k} - (1/\hat{\bar{\rho}})(\widehat{\bar{\rho}\tilde{v}_k})(\widehat{\bar{\rho}\tilde{v}_k}) >}{< 2\hat{\bar{\rho}}\hat{\Delta}^2 |\hat{\tilde{S}}|^2 - 2\Delta^2 \widehat{\bar{\rho}|\tilde{S}|^2} >}, \tag{1.21}$$

$$C = \frac{\left\langle [\widehat{\bar{\rho}\tilde{v}_k \tilde{v}_l} - (1/\hat{\bar{\rho}})(\widehat{\bar{\rho}\tilde{v}_k}\widehat{\bar{\rho}\tilde{v}_l})]\tilde{S}_{kl} - \frac{1}{3}\tilde{S}_{mm}(\mathcal{T}_{jj} - \hat{\tau}_{jj}) \right\rangle}{\left\langle -2\hat{\bar{\rho}}\hat{\Delta}^2 |\hat{\tilde{S}}|(\hat{\tilde{S}}_{kl}\tilde{S}_{kl} - \frac{1}{3}\hat{\tilde{S}}_{mm}\tilde{S}_{jj}) + 2\Delta^2(\widehat{\bar{\rho}|\tilde{S}|\tilde{S}_{kl}}\tilde{S}_{kl} - \frac{1}{3}\widehat{\bar{\rho}|\tilde{S}|\tilde{S}_{mm}}\tilde{S}_{jj}) \right\rangle}, \tag{1.22}$$

$$\tag{1.23}$$

where <> indicates volume averaging and $\mathcal{T}_{jj} - \hat{\tau}_{jj} = \mathcal{L}_{jj} = \widehat{\bar{\rho}\tilde{v}_j\tilde{v}_j} - (1/\hat{\bar{\rho}})(\widehat{\bar{\rho}\tilde{v}_j})(\widehat{\bar{\rho}\tilde{v}_j})$ and over hat means the test filter is applied.

Wall-Adapting Local Eddy-Viscosity (WALE) In the WALE model (Geurts et al., 1993), the eddy-viscosity is modeled as

$$\mu_t = \rho (C_w \Delta)^2 \frac{(S_{ij}^d S_{ij}^d)^{3/2}}{(\widetilde{S}_{ij}\widetilde{S}_{ij})^{5/2} + (S_{ij}^d S_{ij}^d)^{5/4}}. \tag{1.24}$$

Here, C_w is the WALE model constant, Δ is the average cell size, and S_{ij}^d is the traceless symmetric part f the square of the velocity gradient tensor, given by (Nicoud and Ducros, 1999)

$$S_{ij}^d = \frac{1}{2}\left(g_{ij}^2 + g_{ji}^2\right) - \frac{1}{3}\delta_{ij} g_{kk}^2; \quad g_{ij}^2 = g_{ik}g_{kj}; \quad g_{ij} = \frac{\partial u_i}{\partial x_j}. \tag{1.25}$$

Care must be taken in evaluating the components of S_{ij}^d.

Vreman Model The Vreman model Vreman (2004) is supposed to compete with the dynamic model by performing analytical filtering thus avoiding the computationally-intensive dynamic filtering of the dynamic model. The model is relatively simple to implement and is given by

$$\mu_t = \rho C_v \Delta^2 \sqrt{\frac{B_\beta}{\alpha_{ij}\alpha_{ij}}}, \tag{1.26}$$

where

$$C_v \approx 2.5 C_s^2, \tag{1.27}$$

$$\alpha_{ij} = \frac{\partial \widetilde{u_j}}{x_i}, \tag{1.28}$$

and

$$B_\beta = \beta_{11}\beta_{22} - \beta_{12}^2 + \beta_{11}\beta_{33} - \beta_{13}^2 + \beta_{22}\beta_{33} - \beta_{23}^2; \quad \beta_{ij} = \alpha_{mi}\alpha_{mj}. \tag{1.29}$$

1.2 Generic Scalar Transport

Wasatch can solve any scalar transport equation of the general form

$$\frac{\partial \rho\phi}{\partial t} = -\nabla \cdot \rho\phi\mathbf{u} + \nabla \cdot \rho\Gamma_\phi \nabla\phi + S_\phi. \tag{1.30}$$

This general form is basically the strong form of the scalar transport equation when density is variable. Wasatch has the capability of choosing between the strong form or the weak form of this transport equation. Density type can also be specified in the input files of wasatch to determine if it is constant or variable. Regarding these capabilities we can have three more possible forms of transport equations rather than 1.30. Those forms are:

- Strong form with constant density,

$$\frac{\partial \phi}{\partial t} = -\nabla \cdot \phi\mathbf{u} + \nabla \cdot \Gamma_\phi \nabla\phi + \frac{S_\phi}{\rho}. \tag{1.31}$$

- Weak form with variable density,

$$\frac{\partial \phi}{\partial t} = -\mathbf{u} \cdot \nabla\phi + \frac{1}{\rho}\nabla \cdot \rho\Gamma_\phi \nabla\phi + \frac{S_\phi}{\rho}. \tag{1.32}$$

- Weak from with constant density

$$\frac{\partial \phi}{\partial t} = -\mathbf{u} \cdot \nabla\phi + \nabla \cdot \Gamma_\phi \nabla\phi + \frac{S_\phi}{\rho}. \tag{1.33}$$

Wasatch can now solve transport equations in strong form with either constant or variable density. Solving the weak form of the equation has not been implemented yet and is in progress.

The main reason that wasatch keeps both strong and weak form is to compare their result in future to see which one is more beneficial. In strong form with variable density wasatch solves for, $\rho\phi$, and in order to update the for scalar variable we need to solve a nonlinear equation where, $\rho\phi$ and $\rho = F(\phi)$ are given, in order to find the scalar variable in the current time step. On the other hand, weak form has its own problems. For example to solve weak form equations' convective terms in order to apply limiters on this term this equation is used

$$\mathbf{u} \cdot \nabla\phi = \nabla \cdot \phi\mathbf{u} - \phi\nabla \cdot \mathbf{u}. \tag{1.34}$$

And limiters can be used on the terms on the RHS easily. The scalar variable in this case should be assumed to be approximately constant in each cell on the grid.

1.2.1 LES Modeling

The process of filtering the generic scalar transport equation follows from that used for the momentum equations.

Filtered Scalar Transport Equation By applying the traditional filter on the strong form of the scalar transport equation with variable density for capturing the most general form of the equation, the filtered scalar equation will be:

$$\frac{\partial\overline{\rho\phi}}{\partial t} = -\nabla \cdot \overline{\rho\phi\mathbf{u}} + \nabla \cdot \overline{\rho\Gamma_\phi\nabla\phi} + \overline{S_\phi}. \tag{1.35}$$

Application of the Favre filtering to the scalar transport equation term-by-term yields:

- $\frac{\partial\overline{\rho\phi}}{\partial t} = \frac{\partial\overline{\rho}\widetilde{\phi}}{\partial t}$
- $\nabla \cdot \overline{\rho\phi\mathbf{u}} = \nabla \cdot \overline{\rho}\widetilde{\phi}\widetilde{\mathbf{u}} = \nabla \cdot \overline{\rho}\widetilde{\phi}\widetilde{\mathbf{u}} + \left(\nabla \cdot \overline{\rho\phi\mathbf{u}} - \nabla \cdot \overline{\rho}\widetilde{\phi}\widetilde{\mathbf{u}}\right) \equiv \nabla \cdot \overline{\rho}\widetilde{\phi}\widetilde{\mathbf{u}} + \nabla \cdot J_t^R; \quad J_t^R \equiv \overline{\rho\phi\mathbf{u}} - \overline{\rho}\widetilde{\phi}\widetilde{\mathbf{u}}$
- $\nabla.\overline{\rho\Gamma_\phi\nabla\phi} = \nabla.(\overline{\rho\Gamma_\phi\nabla\phi}) = \nabla.(\overline{\rho}\overline{\Gamma_\phi}\nabla\widetilde{\phi}) + \left(\nabla.(\overline{\rho\Gamma_\phi\nabla\phi}) - \nabla.(\overline{\rho}\overline{\Gamma_\phi}\nabla\widetilde{\phi})\right) = \nabla.(\overline{\rho}\overline{\Gamma_\phi}\nabla\widetilde{\phi}) + \nabla.J_t^L; \quad J_t^L = \nabla.(\overline{\rho\Gamma_\phi\nabla\phi}) - \nabla.(\overline{\rho}\overline{\Gamma_\phi}\nabla\widetilde{\phi})$
- $\overline{S_\phi} = \overline{S_\phi}$

where J_t^R and J_t^L are subgrid-scale diffusive fluxes that result from applying filtering to the convective and diffusive terms in the scalar transport equations, respectively. At the outset, the filtered scalar equations may be written as

$$\frac{\partial\overline{\rho}\widetilde{\phi}}{\partial t} = \nabla \cdot \overline{\rho}\widetilde{\phi}\widetilde{\mathbf{u}} + \nabla.(\overline{\rho}\overline{\Gamma_\phi}\nabla\widetilde{\phi}) + \nabla \cdot J_t^R + \underbrace{\nabla.J_t^L}_{neglect} + \overline{S_\phi}. \tag{1.36}$$

It is evident from Eq. (1.36) that both subgrid diffusive fluxes J_t^R and J_t^L require closure. However, it may be argued that the subgrid diffusive flux J_t^L resulting from the diffusive terms is

negligible compared to J_t^R that results from filtering of the convective terms or not. As long as we model these turbulent closure terms with a turbulent diffusive term, one may argue that J_t^L is more like a diffusive term, if you look at what it is representing, rather than J_t^R, which is representing a convective form term, so, J_t^R should be neglected comparing to J_t^L. But we keep neglecting J_t^L just to be consistent with the same approximation that we made in momentum equation in LES mode. Hence, the resulting filtered momentum equations take the form

$$\frac{\partial \overline{\rho}\widetilde{\phi}}{\partial t} = \nabla \cdot \overline{\rho}\widetilde{\phi}\widetilde{\mathbf{u}} + \nabla.(\overline{\rho}\overline{\Gamma_\phi}\nabla\widetilde{\phi}) + \nabla \cdot J_t^R + \overline{S_\phi}. \tag{1.37}$$

Note that we could even get a total turbulent diffusive flux as $J_t = J_t^L + J_t^R$ and say that our closure model is representing both of the subgrid diffusive fluxes. Which is like merging J_t^L into our current subgrid closure model.

As we mentioned, the subgrid diffusive flux J_t^R again requires closure as it introduces unknown quantities. The scalar closure model will be discussed in the next subsection.

1.2.1.1 Subgird Scalar Closure

We model our subgrid diffusive flux term(s) as some diffusive form terms in Wasatch,

$$J_t^R = \Gamma_{\phi_T}\nabla\widetilde{\phi}, \tag{1.38}$$

where Γ_{ϕ_T} is turbulent diffusivity and in order to obtain this constant we utilize the turbulent Schmidt number which is defined as

$$Sc_t = \frac{\mu_t}{\rho\Gamma_{\phi_t}} = \frac{\nu_t}{\Gamma_{\phi_t}}. \tag{1.39}$$

With this definition if we just define the Schmidt number of our problem, by getting the turbulent viscosity, μ_t, from one of our closure models for momentum equation, we can obtain turbulent diffusivity as

$$\Gamma_{\phi_t} = \frac{\rho S c_t}{\mu_t}. \tag{1.40}$$

Finally, our filtered scalar transport equation is

$$\frac{\partial \overline{\rho}\widetilde{\phi}}{\partial t} = \nabla \cdot \overline{\rho}\widetilde{\phi}\widetilde{\mathbf{u}} + \nabla.(\overline{\rho}(\overline{\Gamma_\phi} + \overline{\Gamma_{\phi_t}})\nabla\widetilde{\phi}) + \overline{S_\phi}. \tag{1.41}$$

1.3 Population Balances

Wasatch currently supports a basic population balance equation with a single internal coordinate. The targeted equation is written as

$$\frac{\partial N}{\partial t} = -\nabla \cdot \mathbf{u}N + \frac{\partial gN}{\partial r} + b; \quad N \equiv N(\mathbf{x}, r, t), \tag{1.42}$$

where r is an internal coordinate, $g \equiv \frac{\mathrm{d}r}{\mathrm{d}t}$ denotes the growth rate of r, and b represents the death/birth rates of particles. To make further headway, solution of the population balance equation

is usually achieved by solving for the moments of the number density function, N. The k^{th} moment is defined as

$$m_k \equiv \int_{-\infty}^{+\infty} r^k N(\mathbf{x}, r, t)\, \mathrm{d}r. \tag{1.43}$$

By applying the moment transformation to 1.42, we recover the moment transport equation

$$\frac{\partial m_k}{\partial t} = -\nabla \cdot \mathbf{u} m_k - k \int_{-\infty}^{+\infty} r^{k-1} g N \, \mathrm{d}r + \int_{-\infty}^{+\infty} r^k b \, \mathrm{d}r \equiv F(m_{k+i}); \quad i \in \mathbb{Q}. \tag{1.44}$$

The right-hand-side (RHS) in Eq. (1.44) may be a function of any moment order. As such, we can write additional transport equations for these moments. However, one usually ends up with a several moments that require closure. This may be achieve by the quadrature method of moments.

The quadrature method of moments (QMOM) is a closure technique used to solve the moment transport equations for population balance problems with a single internal coordinate. The technique is based on Gaussian quadrature to achieve closure for the unknown moments. Subsequently, the quadrature nodes and abscissae are obtained using the lower order moments by using the product-difference algorithm. The product-difference algorithm incurs a series of recursion formulas that result in an eigenvalue problem whose solution yields the abscissae (eigenvalues) and weights (eigenvectors).

We recently implemented the moment transport equation in Wasatch with the flexibility to solve for any number of moments. We also added three growth rate models; namely, constant growth, mono-surface nucleation, and bulk diffusion to illustrate the ease with which new growth models can be implemented.

Chapter 2

Order of Accuracy Verification

2.1 Introduction

Order-of-accuracy analysis is based on the concept that the error must be reduced as spatial and temporal resolutions are increased. By comparing the errors at different grid and time resolutions, one can estimate the theoretical order of accuracy for a code. We first review the general concept of analyzing the order of accuracy of a code and follow that with the accuracy verification in Wasatch.

The order of accuracy of a discretization method is defined as *the exponent of the leading order term in the total error* incurred by the discretization process (Oberkampf and Roy, 2010). The total error refers to the accumulation of truncation and round off errors throughout the spatial and temporal domains. It can be generally defined as

$$E = |\phi_{\text{exact}} - \phi_{\text{numerical}}|\,, \tag{2.1}$$

where ϕ_{exact} and $\phi_{\text{numerical}}$ refer to the exact and numerical solutions, respectively.

Round-off errors are those due to finite precision calculation on a computer while the truncation error represents the error induced by neglecting terms in a finite numerical approximation. For most practical purposes, one can neglect round-off errors and focus on how the truncation error is propagated throughout the solution domain. This is the strategy that we will adopt in this document.

To illustrate how the truncation error is estimated, we make use of Taylor series expansions to approximate the various differential terms in an equation. For example, for a transported scalar ϕ on a grid with uniform spacing Δx, the Taylor series at an arbitrary point i is written as

$$\phi_{i+1} = \phi_i + \frac{\mathrm{d}\phi_i}{\mathrm{d}x}\Delta x + \frac{\mathrm{d}^2\phi_i}{\mathrm{d}x^2}\frac{\Delta x^2}{2!} + \frac{\mathrm{d}^3\phi_i}{\mathrm{d}x^3}\frac{\Delta x^3}{3!} + \ldots. \tag{2.2}$$

One can then approximate the first derivative at grid point i via

$$\frac{\mathrm{d}\phi_i}{\mathrm{d}x} = \frac{\phi_{i+1} - \phi_i}{\Delta x} + \underbrace{\frac{\mathrm{d}^2\phi_i}{\mathrm{d}x^2}\frac{\Delta x}{2!} + \frac{\mathrm{d}^3\phi_i}{\mathrm{d}x^3}\frac{\Delta x^2}{3!} + \ldots.}_{\text{truncation error}} \tag{2.3}$$

Then, one can choose to approximate the first derivative by using the first term on the right-hand-side of the previous equation. The remaining terms are then known as the truncation error.

The truncation error has to satisfy certain properties for a discretization scheme to be consistent. For example, all terms in the truncation error must decrease monotonically. Furthermore, the coefficients in the truncation error must be bounded.

The truncation error is usually lumped into a single term and assigned the O terminology

$$T.E. = \underset{\Delta x \to 0}{O}(\Delta x^m), \quad m \in \mathbb{R}^+. \tag{2.4}$$

This definition has a precise meaning: as $\Delta x \to 0$, the truncation error is proportional to Δx^p with a proportionality constant that is independent of Δx. Alternatively, one can write

$$\lim_{\Delta x \to 0} \frac{T.E.}{\Delta x^m} = M, \tag{2.5}$$

where M is a constant. Then, the truncation error can be written as

$$T.E. = M\Delta x^m. \tag{2.6}$$

Subsequently, and by neglecting round-off errors, the total error consists of the accumulated truncation error when discretization is carried out throughout the spatial or temporal domain. In general, this is not the sum of the truncation error incurred by applying a discretization method at a single point. We will discuss how the truncation error accumulates shortly, but for the moment, the total error can be written as

$$E = \alpha \Delta x^p, \tag{2.7}$$

where α is a real constant that is independent of the mesh size while p is a positive real number that designates the order-of-accuracy of a discretization method. The purpose of an order-of-accuracy verification is to determine the exponent p in the total error induced by the discretization process.

2.2 Order-of-Accuracy in a Single Dimension

To solve for p in Eq. (2.7), one needs two equations. With knowledge of an exact solution, a test problem can be setup and solved on two different grid spacings, Δx_1 and $\Delta x_2 = \frac{\Delta x_1}{r_x}$. Then, one calculates the errors associated with these two grids. These are given by

$$\begin{cases} E_1 = & \alpha \Delta x_1^p \\ E_2 = & \alpha \Delta x_2^p \end{cases}. \tag{2.8}$$

These equations can be easily solved by taking the difference of their logarithms. One recovers

$$\ln E_1 - \ln E_2 = p \ln \Delta x_1 - p \ln \Delta x_2 \tag{2.9}$$

or

$$\ln \frac{E_1}{E_2} = p \ln \frac{\Delta x_1}{\Delta x_2} = p \ln r_x \tag{2.10}$$

finally, the observed order of accuracy is at hand

$$p = \frac{\ln \frac{E_1}{E_2}}{\ln r_x}. \tag{2.11}$$

A word of caution must be said about temporal order of accuracy analysis. Many people think that by taking

It is easiest to first determine the temporal order of accuracy of a transient CFD code. As discussed in the previous paragraph

2.3 Combined Spatio-Temporal Order-of-Accuracy Analysis

Without loss of generality, the truncation error may be represented as

$$E = u_{\text{exact}} - u_{\text{numerical}} = \alpha \Delta t^p + \beta \Delta x^q + \gamma \Delta y^r + \delta \Delta z^s. \tag{2.12}$$

This excludes all terms that have mixed orders in them.

2.4 Order-of-Accuracy Verification in Wasatch

Temporal Order of Accuracy Wasatch currently entertains both forward Euler (FE) and 3rd order Rungue-Kutta strong stability preserving (RK3SSP) time integrators. To verify the temporal order of accuracy of these integrators, an ODE test was designed. In the first, we solve an ODE with a time dependent source term. Care must be taken when calculating the order of accuracy at the first time-step. In this scenario, the observed order of accuracy will be one order higher than the theoretical one. This can be shown as follows. Consider a time-stepping discretization of order p. This solves the following ODE

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = L(\phi). \tag{2.13}$$

Written in discretized form, this leads to

Time-Dependent Source Term Our choice falls upon

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = \sin t; \quad \phi(0) = -1. \tag{2.14}$$

The analytical solution for this case is

$$\phi_{\text{exact}} = -\cos t. \tag{2.15}$$

The tests varied over a time interval of 1 second with systematic mesh refinement. The results for both the FE and the RK3SSP are shown in the table below

Implicit Source Term In this verification test, we solve the following ODE

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = \phi; \quad \phi(0) = 1. \tag{2.16}$$

Since this is an ODE, the spatial resolution is set to [2, 2, 2] in Wasatch. The code is run with both FE and RK3 time integrators at several time-steps ranging from

Table 2.1: Observed temporal order of accuracy using $\frac{d\phi}{dt} = \sin t$. Results are taken at $t = 1$s with systematic timestep refinement.

	Forward Euler		RK3SSP	
Δt	$E = \lvert\Phi - \phi\rvert$	Observed Order	$E = \lvert\Phi - \phi\rvert$	Observed Order
0.1	4.245×10^{-1}		1.596×10^{-8}	
0.05	2.113×10^{-1}	**1.006**	9.976×10^{-10}	**4.000**
0.025	1.054×10^{-1}	**1.003**	6.623×10^{-11}	**4.000**
0.0125	5.265×10^{-2}	**1.001**	3.900×10^{-12}	**3.998**
0.00625	2.631×10^{-2}	**1.000**	2.508×10^{-13}	**3.959**
0.003125	1.315×10^{-2}	**1.000**	3.175×10^{-14}	**2.984**

Table 2.2: Observed temporal order of accuracy using $\frac{d\phi}{dt} = \phi$. Results are taken at $t = 2$s with systematic timestep refinement.

	Forward Euler		RK3SSP	
Δt	$E = \lvert\Phi - \phi\rvert$	Observed Order	$E = \lvert\Phi - \phi\rvert$	Observed Order
0.1	6.615×10^{-1}		5.684×10^{-4}	
0.05	3.491×10^{-1}	**0.922**	7.395×10^{-5}	**2.942**
0.025	1.795×10^{-1}	**0.959**	9.431×10^{-6}	**2.971**
0.0125	9.103×10^{-2}	**0.979**	1.191×10^{-6}	**2.985**
0.00625	4.584×10^{-2}	**0.989**	1.496×10^{-7}	**2.992**
0.003125	2.301×10^{-3}	**0.994**	1.874×10^{-8}	**2.996**
0.0015625	1.152×10^{-3}	**0.977**	2.346×10^{-9}	**2.997**

Table 2.3: Observed spatial order of accuracy using Taylor-Vortex. Results are taken at $t = 2$s with systematic time-step refinement.

	u		v	
Δx	$E = norm(U - u)$	Observed Order	$E = norm(U - u)$	Observed Order
0.1257	1.856×10^{-3}		1.856×10^{-3}	
0.0628	4.650×10^{-4}	**1.997**	4.650×10^{-4}	**1.997**
0.0314	1.163×10^{-4}	**1.999**	1.163×10^{-4}	**1.999**
0.0157	2.908×10^{-5}	**2.000**	2.908×10^{-5}	**2.000**

Spatial Order of Accuracy A method of manufactured solution has been done on wasatch using forward Euler method on a Taylor-Vortex problem to check the spatial and temporal order of accuracy. We state the spatial order of accuracy here as long as the temporal order of accuracy has been already shown in this document in previous subsection.

In order to check the spatial order of accuracy we looked at the four different grid resolutions for our domain, and we chose 0.001, 0.0005, 0.00025 and 0.000125 respectively for their time step. However, we just looked at the results after the first time step. This is in order to neglecting the temporal accuracy in the explicit solvers. Because, those terms will be calculated in the previous time step, therefore, in all of our cases they will be calculated at time zero. So, because of using the same time for these terms we don't let the time accuracy to influence our results for grid convergence study

Taylor-Vortex problem is a 2D problem where the exact solutions are:

$$u(x, y, t) = 1 - A\cos(x - t)\sin(y - t)\exp(-2\nu t), \tag{2.17}$$

$$v(x, y, t) = 1 + A\sin(x - t)\cos(y - t)\exp(-2\nu t). \tag{2.18}$$

Where in this study, $A = 4$ and $\nu = 0.1$.

In order to calculate the error the norm over the whole area is taken to obtain

$$E = \|u_{exact} - u_{numerical}\|. \tag{2.19}$$

So we can consider all of the point on the grid together.

Results for Spatial order of accuracy The results for both u and v using the FE are shown in the table below

As it is clear from the table the spatial order of accuracy is 2 as it was expected for the FE method with central 3 point discretization in space.

Chapter 3

Scalability

To test some scalability aspects of Wasatch, we implemented the model discussed in Eq. (**??**). [DEVIN, can you add some results here?]

Chapter 4

Boundary Conditions

4.1 Overview of Defining Boundary Conditions

Wasatch provides the machinery to apply Dirichlet and Neumann boundary conditions. Our approach is based on setting boundary conditions on ghost cells at the boundaries of the computational domain to reproduce the desired boundary at the face. This is illustrated in Fig. 4.1.

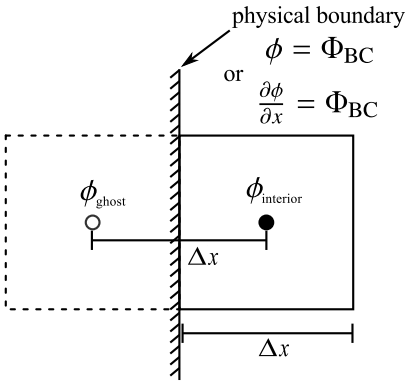


Figure 4.1: Boundary condition specification in Wasatch. Here, Φ_{BC} designates the desired value of the scalar or its flux at the boundary, $\phi_{interior}$ is the interior value of the scalar, and ϕ_{ghost} is the ghost value to be set.

Mathematically, this may be written as follows. Consider a scalar ϕ for which a boundary condition is to be specified. The ghost value of ϕ may then be written as

$$\phi_{ghost} = \alpha \phi_{interior} + \beta \Phi_{BC}, \tag{4.1}$$

where α and β are real coefficients that depend on the type of boundary condition specified. These are summarized in Table 4.1 .

For Staggered fields on uniform structured grids, a simplification follows for setting Dirichlet conditions. In this case, certain boundary faces coincide with staggered control volumes and the desired Dirichlet condition may be set directly on those cells.

	α	β
Dirichlet	0.5	0.5
Neumann	1.0	$-\Delta x$

Table 4.1: Summary of coefficients for setting values at ghost cells to reproduce a desired boundary condition at a physical boundary.

4.2 Scalar Transport Boundary Conditions

The treatment of scalar boundary conditions follows from the previous discussion. Specification of either fixed values or fluxes of transported scalars is easily achieved using the machinery discussed in §4.1.

4.3 Fluid Flow Boundary Conditions

4.3.1 Inlets, Walls, and Moving Walls

Inlets and walls, whether moving or stationary, share the same boundary condition specification. The user only specifies the momentum at those boundaries,

$$\rho \mathbf{u} = \rho(U, V, W). \tag{4.2}$$

Treatment of the pressure is derived from the pressure Poisson equation as follows. For simplicity, we illustrate this in one-dimension at a boundary cell on the x-minus side of the computational domain as shown in 4.2. In this case, the pressure poisson equation is written as

$$\frac{\partial^2 p}{\partial x^2} = \frac{\partial F}{\partial x}. \tag{4.3}$$

Using second order finite differencing, Eq. (4.3) may be written as

$$\frac{p_{-1} - 2p_0 + p_1}{\Delta x^2} = \frac{F_{\frac{1}{2}} - F_{-\frac{1}{2}}}{\Delta x}. \tag{4.4}$$

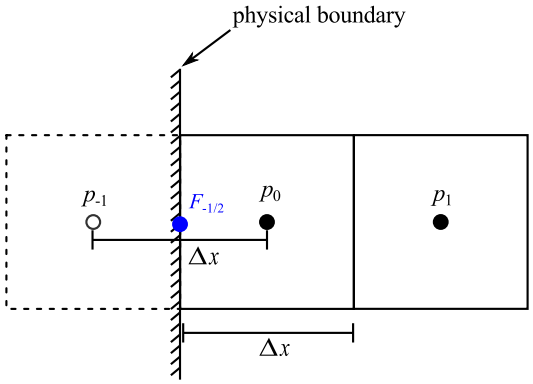


Figure 4.2: Derivation of pressure boundary conditions.

Here, both p_{-1} and $F_{i-\frac{1}{2}}$ are unknown and one may be inclined to specify these quantities at the boundary. However, by using the x-momentum equation, one can relate these quantities by projecting and evaluating the momentum equations in the direction normal to the boundary. In our example, on an x-minus face, we evaluate the x-momentum equation at the x-minus face. This yields

$$\left(\frac{\partial \rho u}{\partial t}\right)_{-\frac{1}{2}} = -\left(\frac{\partial p}{\partial x}\right)_{-\frac{1}{2}} - F_{-\frac{1}{2}}. \tag{4.5}$$

Again, using central finite differencing, one recovers

$$\left(\frac{\partial \rho u}{\partial t}\right)_{-\frac{1}{2}} = -\frac{p_0 - p_{-1}}{\Delta x} - F_{i-\frac{1}{2}} \tag{4.6}$$

or

$$\frac{p_{-1} - p_0}{\Delta x} = \left(\frac{\partial \rho u}{\partial t}\right)_{-\frac{1}{2}} + F_{-\frac{1}{2}}. \tag{4.7}$$

The force balance in 4.7 states that the pressure gradient is balanced by accumulation of momentum and convective and diffusive fluxes - a restatement of Newton's second law. One then substitutes 4.7 into 4.4. This operation returns the following Poisson equation at the boundary

$$\frac{\left(\frac{\partial \rho u}{\partial t}\right)_{-\frac{1}{2}}}{\Delta x} + \frac{F_{-\frac{1}{2}}}{\Delta x} + \frac{-p_0 + p_1}{\Delta x^2} = \frac{F_{\frac{1}{2}} - F_{-\frac{1}{2}}}{\Delta x}. \tag{4.8}$$

Notice how the diffusive and convective fluxes cancel out and one is left with

$$\frac{-p_0 + p_1}{\Delta x^2} = \frac{F_{\frac{1}{2}}}{\Delta x} - \frac{\left(\frac{\partial \rho u}{\partial t}\right)_{-\frac{1}{2}}}{\Delta x}. \tag{4.9}$$

According to 4.9, both the pressure-coefficient-matrix and the pressure right-hand-side must be modified. Because $F_{-\frac{1}{2}}$ never shows up in 4.9, we simply set it to zero in the Wasatch input file while also setting the coefficient for p_{-1} to 0 (the code automatically takes care of that).

4.3.2 Pressure Outlet

The designation of a pressure outlet has a different meaning for different CFD schools of thought. In Wasatch, a pressure outlet is simply an artificial location in space where one does not have information about the velocity or momentum profile. In practice, however, one may have knowledge of the pressure at such boundaries.

The user must specify a value for the outlet pressure. This value may be specified at the extra-cell or on the domain boundary face. Both options give stable results. In Wasatch, we implement the latter option.

Treatment of the pressure follows the same guidelines exposed in the previous section. In this case, however, the pressure p_{-1} may be related to $P_{-\frac{1}{2}}$ on the boundary face. Given that the pressure lives on scalar cell centers, using second order approximation, we have

$$P_{-\frac{1}{2}} = \frac{p_{-1} + p_0}{2} \tag{4.10}$$

or

$$p_{-1} = 2P_{-\frac{1}{2}} - p_0. \tag{4.11}$$

Then, the pressure poisson equation (evaluated at the boundary) returns

$$\frac{-3p_0 + p_1}{\Delta x^2} = -2P_{-\frac{1}{2}} + \frac{F_{\frac{1}{2}} - F_{-\frac{1}{2}}}{\Delta x}. \tag{4.12}$$

The momentum partial right-hand-side remains unknown in the previous equation. One can choose either of the following conditions

$$\begin{cases} F_{-\frac{1}{2}} = 0 \\ \frac{\partial F}{\partial x} = 0 \end{cases}. \tag{4.13}$$

The former may be interpreted to imply a balance between diffusive and convective fluxes at the pressure outlet boundary. It may also be interpreted as a zero flux condition for convective and diffusive fluxes. The latter is self evident where one is assuming a self-similar profile for convective and diffusive fluxes.

Chapter 5

Modeling General Scalar Transport

Chapter 6

Modeling Species Transport

Chapter 7

Modeling Dispersed-Phase Flows

7.1 Precipitation

Wasatch supports precipitation-physics reactions utilizing the quadrature method of moments to solve the population balance equation. The internal coordinate in this case is the particle radius.

7.1.1 Governing Equations

The precipitation model utilized in Wasatch uses the implemented moment transport equations.

$$\frac{\partial m_k}{\partial t} = -\nabla \cdot \mathbf{u} m_k - k \int_{-\infty}^{+\infty} r^{k-1} g N \, \mathrm{d}r + \int_{-\infty}^{+\infty} r^k b \, \mathrm{d}r \equiv F(m_{k+i}); \quad i \in \mathbb{Q}, \tag{7.1}$$

with QMOM used as closure for the moments. The moments that require closure will change based on the growth rate model that is utilized, and the closure is taken care of automatically in the Wasatch algorithm. The relavent terms dictating the reaction rate are tabulated by the extent of reaction and the mixture fraction. The main term required is the supersaturation value of the liquid. The source term for the reaction progress is

$$S_\xi = \frac{1}{\iota} \sum_{\alpha=1}^{n} \nu_\alpha S_{3,\alpha}, \tag{7.2}$$

where $S_{3,\alpha}$ is the source term of the third moment, representing the volume, of species α, ν_α is the molecular volume, and ι is the maximum moles of precipitate.

7.1.2 Birth Models

Nucleation terms in Wasatch can be specifed as one of three distributions, point, uniform, and normal. Each of these distributions requires the specification of a nucleation radius which can either be set as a constant or as the critical radius $r^* = r_0 / \log(S)$. The uniform and normal distributions also require a specified standard deviation for the distribution. The coefficient used for the birth term can either be set as constant or using the classical nucleation definition

$$B_0 = \exp\left[\tfrac{16\pi}{3} (\tfrac{\gamma}{K_B T})^3 \frac{\nu}{N_\mathrm{A} \log(S)^2} \right], \tag{7.3}$$

where γ is the surface energy, K_B is the Boltzman constant, T is the temperature in Kelvin, ν is the molecular volume, and N_A is Avogadro's constant. The Wasatch formulation also includes a "pre-Birth coefficient" which is used as a converison factor from SI units into smaller lengths (μm or nm) used as precipitate measurements.

7.1.3 Growth Models

Currently Wasatch has three growth models implemented; constant growth, bulk diffusion and monosurface nucleation. Each model uses a different form of $g(r)$ with $g(r) = g_0$, $g(r) = g_0\frac{1}{r}$, and $g(r) = g_0 r^2$, respectively, as well as a different requirement for the closure of the moment transport equations as: none, m_{-2} & m_{-1}, and m_{k+1}. The bulk diffusion model uses the appropriate coefficient as

$$g_0 = \nu D C_{eq}(S - \bar{S}) \tag{7.4}$$

with D as the diffusion coefficient, C_{eq} as the equilibirum concentration and where $\bar{S}$ can either be set as one, or set as needed for Ostwald Ripening as

$$\bar{S} = \exp(2\nu\gamma/RTr) \tag{7.5}$$

For Ostwald ripening, the r appearing in the exponential does not allow for this term to be converted into a moment of the problem, therefore the quadrature weights and abscissae are used to evaluate the Ostwald ripening term. For monosurface nucleation the growth coefficient is

$$g_0 = \beta_A D d^3 exp(-\Delta G/K_B T) \tag{7.6}$$

with

$$\Delta G = \frac{\beta_L \gamma^2 d^2}{4\beta_A K_B T \log(S)}, \tag{7.7}$$

where β_L and β_A are shape factors, and d is the molecular diameter. As with the birth models, a "pre-Growth coeffcient" can be used for unit conversions.

7.1.4 Aggregation Models

To account for aggregation in particulate systems, two terms must be considered the death kernel from two particles combining alogn with the birth kernel from the new particle. The general form for each of these is in volume formulation

$$\frac{1}{2}\int_0^v \beta(v-\epsilon,\epsilon)n'(v-\epsilon;t)n'(\epsilon;t)d\epsilon \tag{7.8}$$

$$-n'(v;t)\int_0^\infty \beta'(v,\epsilon)n'(\epsilon;t)d\epsilon \tag{7.9}$$

Converting the population balance into a radius formulation, and applying a moment transform, using quadrature on the integrals yields

$$\frac{1}{2}\sum_i w_i \sum_j w_j (r_i^3 + r_j^3)^{k/3}\beta_{ij} \tag{7.10}$$

$$-\sum_i r_i^k w_i \sum \beta_{ij} w_j \tag{7.11}$$

where β_{ij} is a specified frequency kernel, and r_i & w_i are the abscissae and weights. The Wasatch formulation spilts the frequency term into two, the size dependent frequncy and the fluid property frequency. The fluid property frequency can then be factored outside of the summation. For aggregation due to Brownian motion

$$\beta_{ij} = \frac{2k_B T}{3\rho} \times \frac{(r_i + r_j)^2}{r_i r_j}, \tag{7.12}$$

where k_B is the Boltzmann cosntant, T is the temperature, and ρ is the fluid density. For aggregation due to turbulent shear forces the frequency kernel is

$$\beta_{ij} = \frac{4}{3}\left(\frac{3\pi}{10}\right)^{1/2}\left(\frac{\epsilon}{\nu}\right)^{1/2} \times \left(r_i + r_j^3\right), \tag{7.13}$$

where ϵ is the energy dissipation and ν is the kinematic viscosity of the fluid. In addition a simple constant frequency is included, $B_{ij} = 1$.

7.1.5 Multi Environment Mixing Model

Wasatch has a subgrid scale model for mixing for the precipitation flows. Since the source terms of the moment transport equations are highly dependent on the current value of the moments, which are solved on the slow scale, a tabulation of data for use in a continuous PDF (i.e. such as a flamelet) is not a viable option. Instead a discrete PDF is used consisting of three delta functions, one at mixture fraction $z = 0$, one at mixture fraction $z = 1$, and the third at the average mixture fraction. In precipitation flows, no reaction occurs at either mixture fraction 0 or 1, so only the source terms from the central delta function need to be solved.

Wasatch requires the averaged mixture fraction and the scalar variance to be solved on the grid. Then the weights of the three delta functions are calculated as

$$w_1 = \frac{\left\langle z'^2 \right\rangle}{\langle z \rangle}, \tag{7.14}$$

$$w_2 = \frac{\left\langle z'^2 \right\rangle - \langle z \rangle + \langle z \rangle^2}{\langle z \rangle^2 - \langle z \rangle}, \tag{7.15}$$

$$w_3 = \frac{-\left\langle z'^2 \right\rangle}{\langle z \rangle - 1}, \tag{7.16}$$

where $\langle z \rangle$ is the average mixture fraction, and $\left\langle z'^2 \right\rangle$ is the scalar variance.

When the Multi Environment Model is enabled, the moment equations are only solved for the second environment with the moments from environments 1 and 3 held constant. The averaged moments are then post-processed based on the values of the environment weights. In addition the

transport of the second environment requires a new source term to account for the change in the moments due to mixing

$$S_{\phi_{\alpha,2},\,mix} = -\frac{dw_1/dt}{w_2}\left(\phi_{\alpha,1} - \phi_{\alpha,2}\right) - \frac{dw_3/dt}{w_2}\left(\phi_{\alpha,3} - \phi_{\alpha,2}\right), \tag{7.17}$$

where $\phi_{\alpha,i}$ represents the αth precipitate moment located in the ith mixing environment. With the time derivatives of the weights of the mixing environments calculated as

$$\frac{dw_1}{dt} = \frac{-\left\langle \epsilon_\phi \right\rangle}{\langle z \rangle}, \tag{7.18}$$

$$\frac{dw_3}{dt} = \frac{-\left\langle \epsilon_\phi \right\rangle}{1 - \langle z \rangle}, \tag{7.19}$$

where $\left\langle \epsilon_\phi \right\rangle$ is the scalar dissipation rate.

7.1.6 Viscosity Model

Wasatch has one model to account for the viscosity change of the fluid as particles precipitate. The two parameters dictating the viscosity are the volume fraction of particles and the strain tensor magnitude. The volume fraction is calculated as

$$\phi = \sum_i^n \frac{4\pi}{3}\left(\frac{m_{1,i}}{m_{0,i}}\right)^3 m_{0,i}, \tag{7.20}$$

where $m_{1,i}$ and $m_{0,i}$ are the first and zeroth moments of the ith polymorph phase. In flows with low volume fractions of particles the relative viscosity can be solved for using kinetic theory with ideal assumptions as

$$\mu_r = \frac{\mu}{\mu_0} = 1 + 2.5\lambda\phi, \tag{7.21}$$

with λ as a correction factor for non-idealities. In some case particle flows will have a shear thinning phenomena in which case the viscosity can be expressed as a power law with relation to the strain rate. The model used by Wasatch combines this with the linear fit of the kinetic theory assumptions so that

$$\mu_r = (1 + 2.5\lambda\phi)\,|S|^n, \tag{7.22}$$

where $|S|$ is the strain tensor magnitude. The parameters n and λ should then be fit to experimental data. If no shear thinning is expected, n can be set as 0. This formulation should only be used for low volume fractions of particles.

Chapter 8

Verification and Benchmark Studies

8.1 Constant Density Lid-Driven-Cavity

This case aims at verifying the results obtained by the Wasatch pressure projection method with wall boundary conditions. Although experimental data is scarcely available for this test case, there are several numerical studies on the lid-driven-cavity that one may employ for comparison purposes. Our choice falls on the study by Ghia et al. (1982) which presents a comprehensive review of data in the literature as well as a reliable benchmark for comparison purposes. Here, we reproduce all the cases discussed in their study.

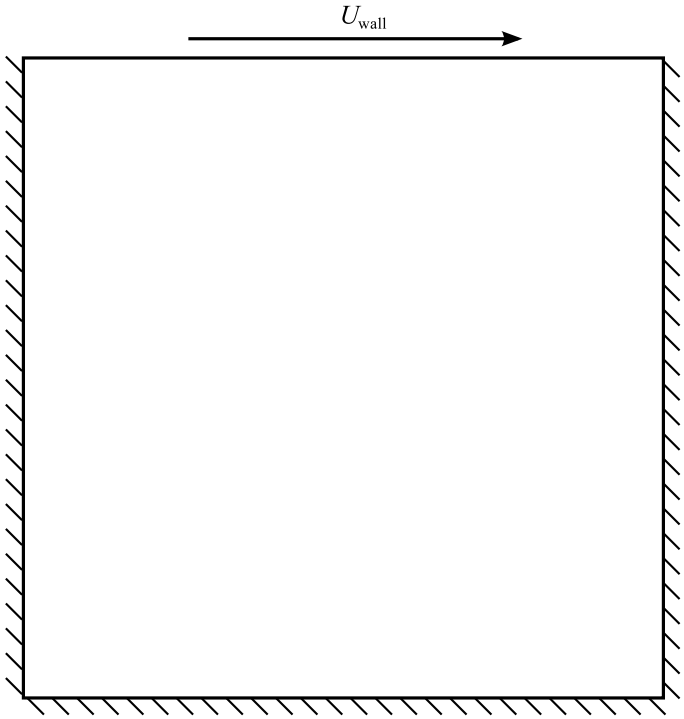


Figure 8.1: Lid-Driven-Cavity geometry.

Table 8.1: Summary of parameters used for the various cases.

Re	ρU_{wall}	Δt	CFL
100	0.1002	0.75	0.0096
400	0.4008	0.75	0.0385
1000	1.002	0.75	0.0963
3200	3.2064	0.75	0.3083
5000	5.01	0.5	0.3212
7500	7.515	0.25	0.2409
10000	10.02	0.125	0.1606

8.1.1 Problem Description

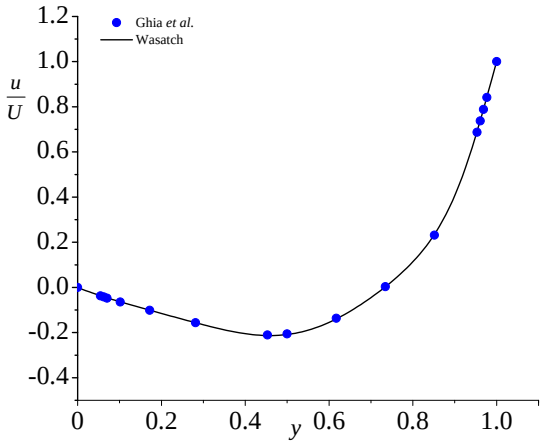
Consider a 1mx1m square cavity that is filled with a fluid with the top lid moving at a constant speed. We choose water at 293 K as our working fluid. The domain is discretized into $N_x = N_y = 128$ equally spaced control volumes (Note that in Uintah, we must specify a nonzero dimension in the orthogonal direction to the cavity's plane. here, we set $L_z = 1$m and $N_z = 2$ with periodic conditions in the z-direction. We also only solve the momentum equations in the plane of the driven cavity).

We test seven cases with different Reynolds number. The Reynolds number for the cavity is defined as $\text{Re} = \frac{\rho U_{\text{wall}} L}{\mu}$ where ρ is the density of the fluid (in this case, water @ 293 K), U_{wall} is the velocity of the lid, L is the width of the lid, and μ is the dynamic viscosity. The Reynolds numbers range is given by Re = {100, 400, 1000, 3200, 5000, 7500, 10000} as dictated by the reference data. The boundary conditions on all walls are set to no-slip and no-permeability conditions ($\rho \mathbf{u} = \mathbf{0}$) except on the moving lid where the tangential momentum is set to reproduce U_{wall} that corresponds to the desired Reynolds number while the normal velocity is set to zero. A summary of these conditions is given in Table

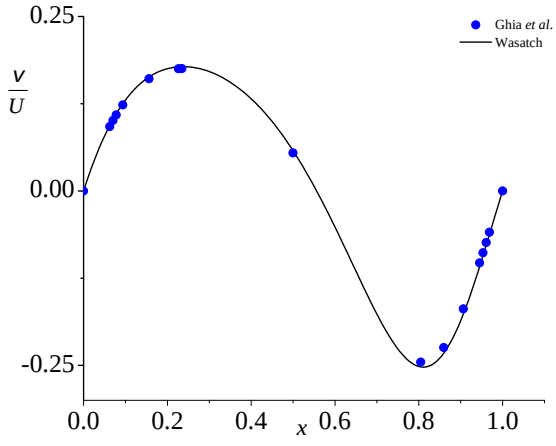
8.2 Taylor Flow in a Channel

The Taylor flow in a channel describes the steady inviscid state of a fluid in a channel with permeable sidewalls. This is a great case for testing inlet and outflow boundary conditions.

8.3 Decay of Isotropic Turbulence

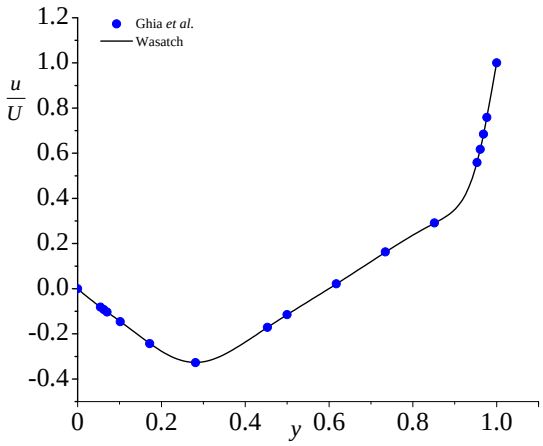


(a) Normalized u-velocity component.

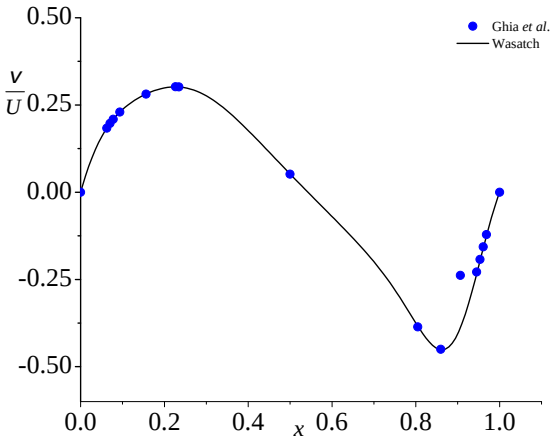


(b) Normalized y-velocity component.

Figure 8.2: Comparison of Wasatch results with those presented in Ghia et al. (1982) for Re = 100.

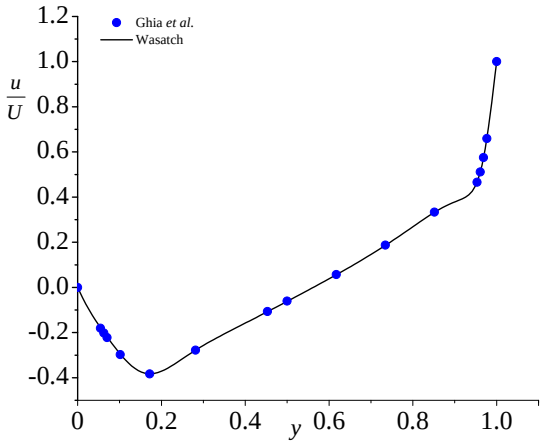


(a) Normalized u-velocity component.

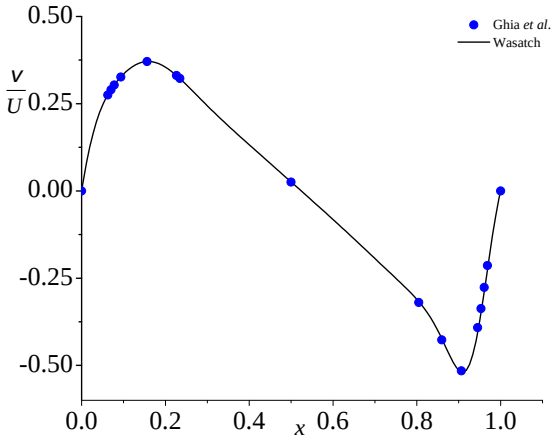


(b) Normalized y-velocity component.

Figure 8.3: Comparison of Wasatch results with those presented in Ghia et al. (1982) for Re = 400.

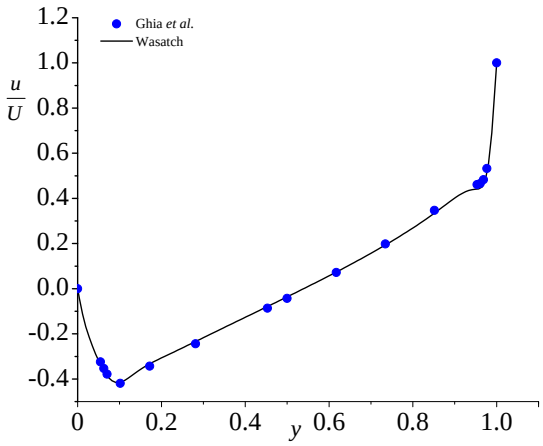


(a) Normalized u-velocity component.

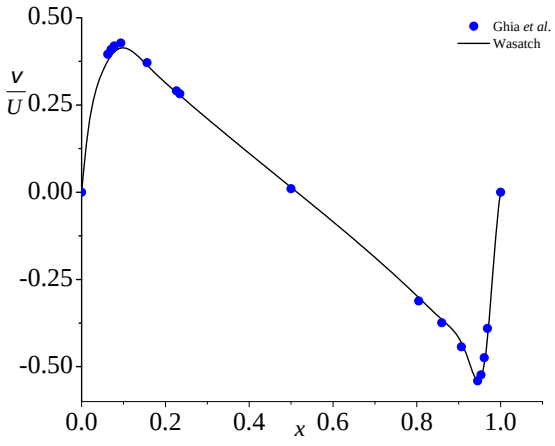


(b) Normalized y-velocity component.

Figure 8.4: Comparison of Wasatch results with those presented in Ghia et al. (1982) for Re = 1000.

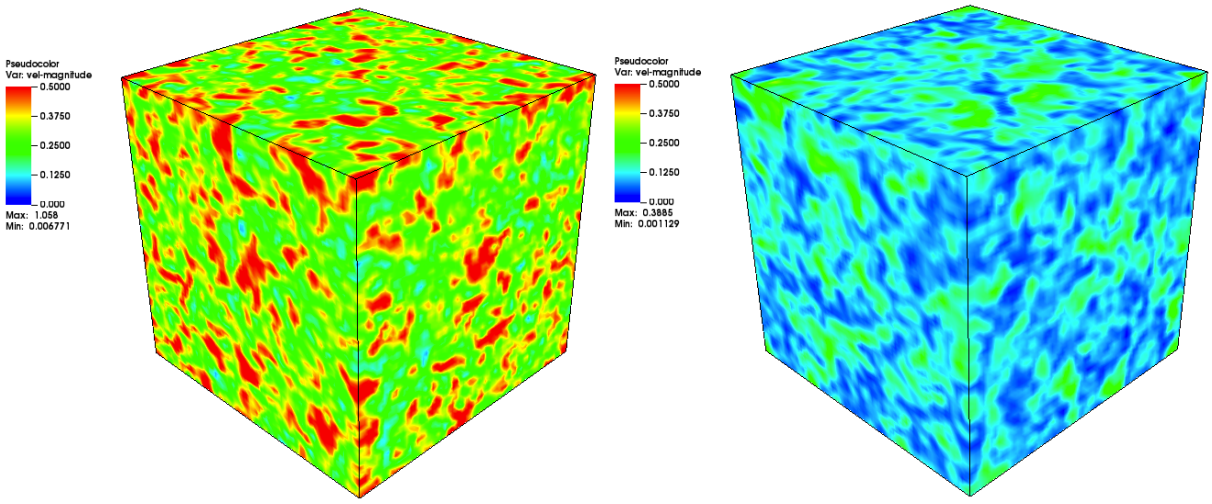


(a) Normalized u-velocity component.



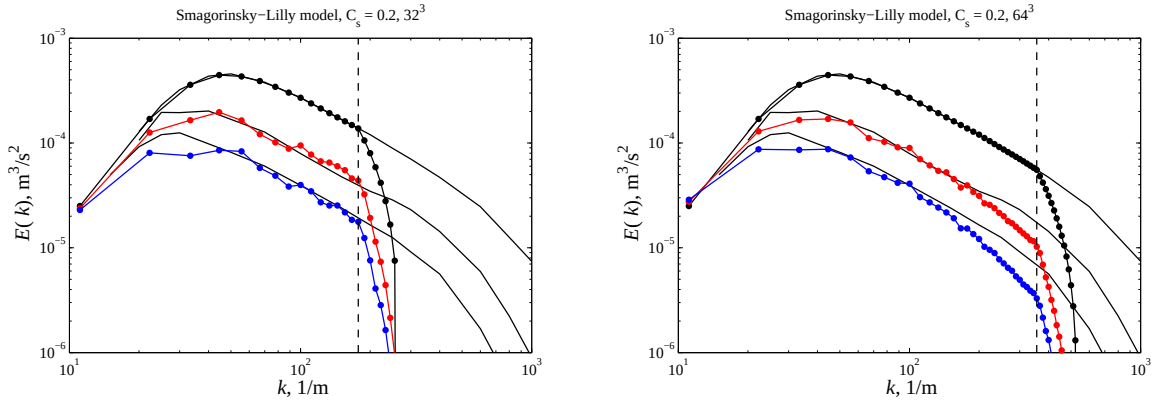
(b) Normalized y-velocity component.

Figure 8.5: Comparison of Wasatch results with those presented in Ghia et al. (1982) for Re = 3200.



(a) $t = 0$ s **(b)** $t = 0.67$ s

Figure 8.6: Contours of velocity magnitude for the decay of isotropic turbulence in a periodic box. Results are shown using the Smagorinsky-Lilly model at (a) $t = 0$ s and (b) $t = 0.67$ s.



(a) **(b)**

Figure 8.7: Turbulent kinetic energy spectrum for the constant Smagorinsky model with $C_s = 0.2$ and for grid sizes (a) 32x32x32, and (b) 64x64x64.

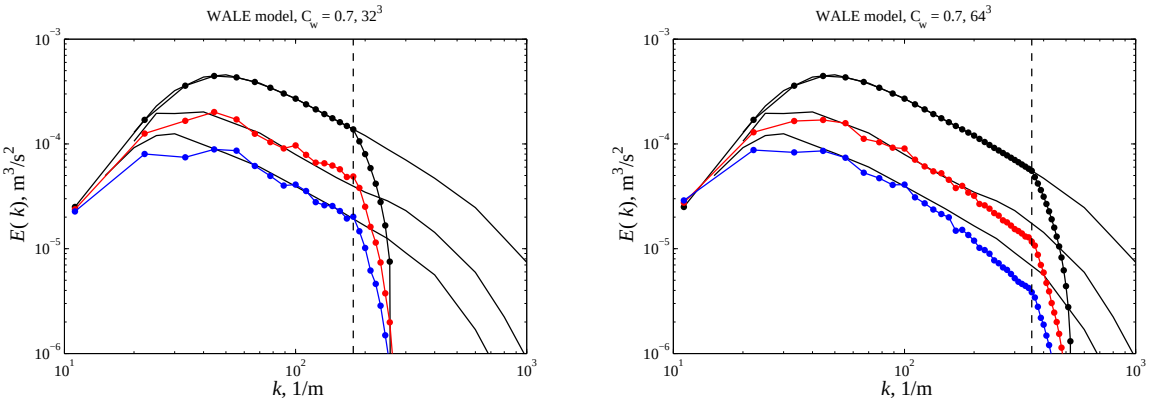


Figure 8.8: Turbulent kinetic energy spectrum for the WALE model with $C_{\mathrm{w}} = 0.7$ and for grid sizes (a) 32x32x32, and (b) 64x64x64.

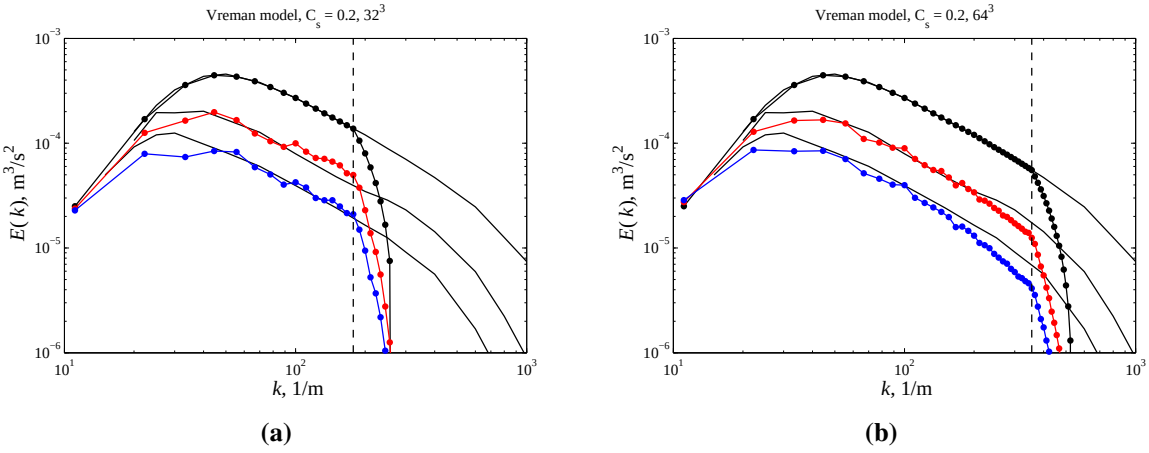


Figure 8.9: Turbulent kinetic energy spectrum for the constant Vreman model with $C_{\mathrm{s}} = 0.2$ (or $C_{\mathrm{v}} = 2.5C_{\mathrm{s}}^2 = 0.1$) and for grid sizes (a) 32x32x32, and (b) 64x64x64.

Chapter 9

Summary of Supported Expressions

Linear Function

The `LinearFunction` expression applies a linear transformation to any field of your choice according to

$$\psi = a\phi + b, \tag{9.1}$$

where ψ is the destination field, ϕ is the source field, a is the slope, and b is the intercept. Below is an example of using the LinearFunction expression to initialize a scalar field ϕ as a linear function of the the coordinate x as $\phi = 0.1x + 1.25$.

```
<BasicExpression type="SVOL">
  <TaskList>initialization</TaskList>
  <NameTag name="phi" state="STATE_N" />
  <LinearFunction slope="0.1" intercept="1.25">
      <!-- specify the tag of the independent variable -->
      <NameTag name="XSVOL" state="STATE_NONE"/>
  </LinearFunction>
</BasicExpression>
```

Listing 9.1: Specification of a LinearFunction as an initialization for a scalar field ϕ.

Parabolic Function

As the name implies, the `ParabolicFunction` expression generates a field of the form

$$\phi = ax^2 + bx + c. \tag{9.2}$$

The following listing shows how to initialize a field with a `ParabolicFunction` as $\phi = 0.1x^2 + 1.25x + 0.3$.

```
<BasicExpression type="SVOL">
  <TaskList>initialization</TaskList>
  <NameTag name="phi" state="STATE_N" />
  <ParabolicFunction a="0.1" b="1.25" c="0.3">
```

```
      <!-- specify the tag of the independent variable -->
      <NameTag name="XSVOL" state="STATE_NONE"/>
  </ParabolicFunction>
</BasicExpression>
```

Listing 9.2: Specification of a LinearFunction as an initialization for a scalar field ϕ.

ExponentialVortex

This expression implements a rotating vortex in a uniform stream. The corresponding streamfunction is given by the superposition of an exponential vortex and a uniform flow and is given by

$$\psi = \Gamma \exp\left(-\frac{r^2}{2R^2}\right) - Vx + Uy, \tag{9.3}$$

where Γ is the strength of the vortex, $r = \sqrt{(x-x_0)^2 + (y-y_0)^2}$ is the radial distance from the center of the vortex, R is the radius of the vortex, and V and U are the the transverse and axial free stream velocities, respectively. The resulting velocity field is then calculated using the Stokes' streamfunction

$$u = \frac{\partial \psi}{\partial y} = U - \frac{\Gamma}{R^2}(y-y_0)\exp\left(-\frac{r^2}{2R^2}\right) \tag{9.4}$$

and

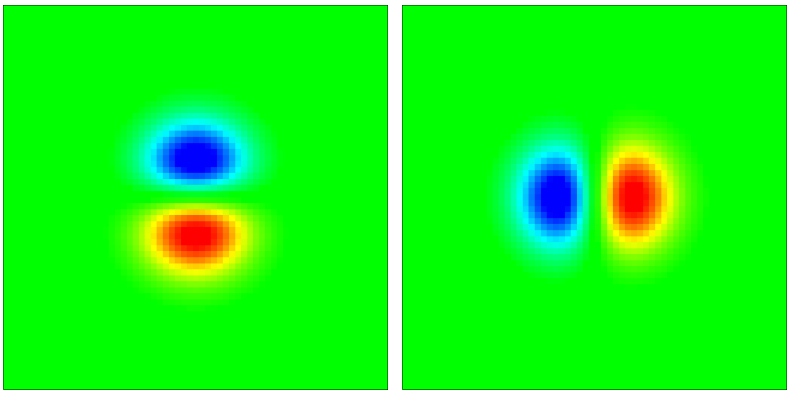
$$v = -\frac{\partial \psi}{\partial x} = V + \frac{\Gamma}{R^2}(x-x_0)\exp\left(-\frac{r^2}{2R^2}\right). \tag{9.5}$$

This expression is best used as an initialization for the momentum or velocity fields.

What follows is an example for initializating the axial and transverse momentum components with an exponential vortex, centered at $[0.5, 0.5]$ with a strength of 10^{-3}, a radius of $R_0 = 0.1$, and a free stream velocity $U_0 = 1$. Note that you must specify both velocity or momentum components to recover physically relevant initial conditions. You can find the "ups" file in the Wasatch inputs directory under the name "exponential-vortex-example.ups".

```
<BasicExpression type="XVOL">
  <TaskList>initialization</TaskList>
  <NameTag name="x-mom" state="STATE_N" />
  <ExponentialVortex x0="0.5" y0="0.5" G="1e-3" R="0.1" U="1" V="0"
     velocityComponent="X1">
    <Coordinate1>
      <NameTag name="XXVOL" state="STATE_NONE"/>
    </Coordinate1>
    <Coordinate2>
      <NameTag name="YXVOL" state="STATE_NONE"/>
    </Coordinate2>
  </ExponentialVortex>
</BasicExpression>
```

Listing 9.3: Specification of an exponential vortex as initial condition for axial momentum.



(a) x-momentum **(b)** y-momentum

Figure 9.1: Momentum field resulting from ExponentialVortex initialization.

```
<BasicExpression type="YVOL">
  <TaskList>initialization</TaskList>
  <NameTag name="y-mom" state="STATE_N" />
  <ExponentialVortex x0="0.5" y0="0.5" G="1e-3" R="0.1" U="1" V="0"
     velocityComponent="X2">
    <Coordinate1>
      <NameTag name="XYVOL" state="STATE_NONE"/>
    </Coordinate1>
    <Coordinate2>
      <NameTag name="YYVOL" state="STATE_NONE"/>
    </Coordinate2>
  </ExponentialVortex>
</BasicExpression>
```

Listing 9.4: Specification of an exponential vortex as initial condition for transverse momentum.

The resulting flow field generated by this initialization is shown below.

LambsDipole

Lamb's dipole is a steady-state solution to the Euler equations using a vorticity-streamfunction approach. In this approach, one specifies an explicit relation between the vorticity and the streamfunction. Lamb's dipole solution assumes that the azimuthal vorticity has a linear dependence on the streamfunction such that

$$\omega_z = k^2 \psi \tag{9.6}$$

The solution to the Euler equations in this case, in an infinite mass of fluid, results in the

following

$$\psi_{\text{dipole}}(r,\theta) = \begin{cases} \frac{2U_0}{kJ_0(kR)} J_1(kr)\sin\theta & r \le R \\ U_0(r - \frac{R^2}{r})\sin\theta & r > R \end{cases} \tag{9.7}$$

where U_0 is the propagation velocity of the dipole, k is the proportionality constant relating the vorticity to the streamfunction and may be determined from $J_1(kR) = 0$ or $kR \approx 3.8317$, R is the radius of the dipole, and J_0 and J_1 are the zeroth- and first-order Bessel functions of the first kind, respectively. The dipole is finall be superimposed on a uniform stream.

To make this solution accessible to code implementation, it must be transformed to cartesian coordinates. We also wish to place the dipole at any location in the flow field. We first calculate the radial and tangenial velocities of ψ_{dipole}. These are given by the polar streamfunction relations

$$u_r = \frac{1}{r}\frac{\partial\psi}{\partial\theta}; \quad u_\theta = -\frac{\partial\psi}{\partial r} \tag{9.8}$$

These calculations return

$$u_{r,\text{dipole}} = \begin{cases} \frac{2U_0}{kJ_0(kR)} J_1(kr)\cos\theta & r \le R \\ U_0(1 - \frac{R^2}{r^2})\cos\theta & r > R \end{cases} \tag{9.9}$$

$$u_{\theta,\text{dipole}} = \begin{cases} -U_0\sin\theta + \frac{U_0}{J_0(kR)} J_2(kr)\sin\theta & r \le R \\ -U_0(1 + \frac{R^2}{r^2})\cos\theta & r > R \end{cases} \tag{9.10}$$

Finally, the axial and transverse velocities are constructed by setting

$$u_{x,\text{dipole}} = u_{r,\text{dipole}}\cos\theta - u_{\theta,\text{dipole}}\sin\theta \tag{9.11}$$

and

$$u_{y,\text{dipole}} = u_{r,\text{dipole}}\sin\theta + u_{\theta,\text{dipole}}\cos\theta \tag{9.12}$$

By subsituting $\theta = \tan^{-1}(\frac{y}{x})$ or using $\cos\theta = \frac{x}{r}$ and $\sin\theta = \frac{y}{r}$, we recover

$$u_{x,\text{dipole}} = \begin{cases} \frac{2U_0}{kJ_0(kR)r^3}[kry^2 J_0(kr) + (x^2 - y^2)J_1(kr)] & r \le R \\ U_0 + \frac{U_0R^2}{r^4}(r^2 - 2x^2) & r > R \end{cases}; \quad r = \sqrt{x^2 + y^2} \tag{9.13}$$

and

$$u_{y,\text{dipole}} = \begin{cases} \frac{2U_0}{J_0(kR)r^2} J_2(kr) & r \le R \\ -\frac{2U_0R^2}{r^4} xy & r > R \end{cases}; \quad r = \sqrt{x^2 + y^2} \tag{9.14}$$

Finally, the total flowfield with a uniform flow given by U in the x-direction is computed as

$$u_x = U + u_{x,\text{dipole}} \tag{9.15}$$

and

$$u_y = u_{y,\text{dipole}} \tag{9.16}$$

To place the dipole at a random point (x_0, y_0), we simply perform the following substitutions in the above equations

$$x \rightarrow x - x_0; \quad y \rightarrow y - y_0; \quad r \rightarrow \sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{9.17}$$

What follows is an example for initializating the axial and transverse momentum components with a Lamb-dipole vortex, centered at $[0.0, 0.0]$ with a propagation velocity of $U_0 = 0.9$, a radius of $R_0 = 0.1$, and a free stream velocity $U = 0.1$. Note that you must specify both velocity or momentum components to recover physically relevant initial conditions. You can find the "ups" file in the Wasatch inputs directory under the name "dipole-vortex-example.ups".

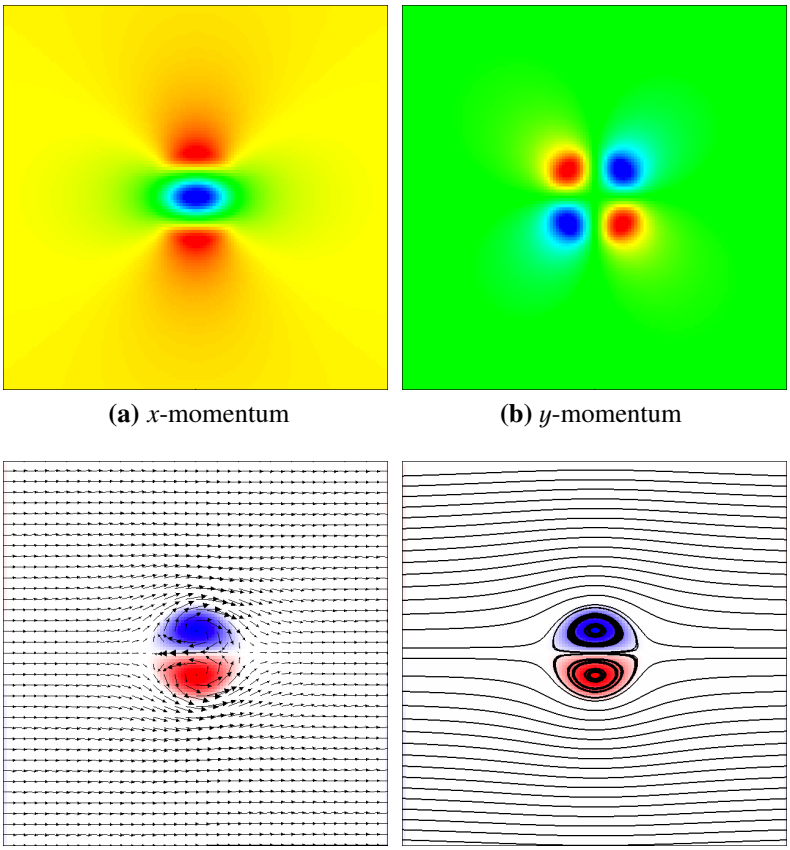
```
<BasicExpression type="XVOL">
  <TaskList>initialization</TaskList>
  <NameTag name="x-mom" state="STATE_N" />
  <LambsDipole x0="0.0" y0="0.0" G="0.9" R="0.25" U="0.1"
     velocityComponent="X1">
    <Coordinate1>
      <NameTag name="XXVOL" state="STATE_NONE"/>
    </Coordinate1>
    <Coordinate2>
      <NameTag name="YXVOL" state="STATE_NONE"/>
    </Coordinate2>
  </LambsDipole>
</BasicExpression>
```

Listing 9.5: Specification of an exponential vortex as initial condition for axial momentum.

```
<BasicExpression type="YVOL">
  <TaskList>initialization</TaskList>
  <NameTag name="y-mom" state="STATE_N" />
  <LambsDipole x0="0.0" y0="0.0" G="0.9" R="0.25" U="0.1"
     velocityComponent="X2">
    <Coordinate1>
      <NameTag name="XYVOL" state="STATE_NONE"/>
    </Coordinate1>
    <Coordinate2>
      <NameTag name="YYVOL" state="STATE_NONE"/>
    </Coordinate2>
  </ExponentialVortex>
</BasicExpression>
```

Listing 9.6: Specification of an exponential vortex as initial condition for transverse momentum.

The resulting flow field generated by this initialization is shown below.



(a) x-momentum

(b) y-momentum

(c) vorticity field wih vectors

(d) vorticity field with streamlines

Figure 9.2: Momentum and vorticity fields resulting from LambsDipole initialization.

Chapter 10

Summary of Current Capabilities and Work in Progress

Table 10.1: Synopsis of Wasatch development.

Task	Status
Advective terms including flux limiters	completed
Diffusive terms for basic diffusive flux expressions	completed
Tabular property evaluation for implementation of common combustion models	completed
Pressure projection	completed
Basic CFD algorithm using expressions graph	completed
Boundary conditions for scalar transport	completed
Quadrature method of moments	completed
Basic population balance equation	completed
Filter operator for dynamic turbulent models	completed
Boundary conditions for momentum equations	in progress
Weak form of transport equations	in progress
Variable density algorithm (including appropriate terms in the pressure projection)	in progress
Higher-order time integrator (Runge-Kutta based)	in progress
LES models for turbulent diffusive fluxes (Smagorinsky and dynamic models)	in progress

Bibliography

Erlebacher, G., Hussaini, M. Y., Speziale, C. G., and Zang, T. A. (1992). Toward the Large-Eddy simulation of compressible turbulent flows. *Journal of Fluid Mechanics*, 238:155–185.

Garnier, E., Sagaut, P., and Adams, N. (2009). *Large eddy simulation for compressible flows*. Springer Verlag.

Geurts, B., Vreman, B., Kuerten, H., and Theofilis, V. (1993). LES modelling errors in free and wall-bounded compressible shear layers. *Engineering Turbulence Modelling and Experiments*, 2:325–334.

Ghia, U., Ghia, K., and Shin, C. (1982). High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *Journal of computational physics*, 48(3):387–411.

Nicoud, F. and Ducros, F. (1999). Subgrid-Scale stress modelling based on the square of the velocity gradient tensor. *Flow, Turbulence and Combustion*, 62(3):183–200.

Oberkampf, W. L. and Roy, C. J. (2010). *Verification and validation in scientific computing*. Cambridge University Press.

Smagorinsky, J. (1963). General circulation experiments with the primitive equations. *Monthly Weather Review*, 91:99–164.

Vreman, A. (2004). An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications. *Physics of fluids*, 16:3670.

Vreman, B., Geurts, B., and Kuerten, H. (1995). Subgrid-modelling in LES of compressible flow. *Applied Scientific Research*, 54(3):191–203.