LA-UR-12-21022

Approved for public release; distribution is unlimited.

Title: A Specification for a Godunov-type Eulerian 2-D Hydrocode, Revision 0

Author(s): Nystrom, William D
Robey, Jonathan M

Intended for: Report

A Specification for a Godunov-type Eulerian 2-D Hydrocode, Revision 0

WD Nystrom and JM Robey

April 24, 2012

Contents

1 Overview

The purpose of this code specification is to describe an algorithm for solving the Euler equations of hydrodynamics in a 2D rectangular region in sufficient detail to allow a software developer to produce an implementation on their target platform using their programming language of choice without requiring detailed knowledge and experience in the field of computational fluid dynamics. It should be possible for a software developer who is proficient in the programming language of choice and is knowledgable of the target hardware to produce an efficient implementation of this specification if they also possess a thorough working knowledge of parallel programming and have some experience in scientific programming using fields and meshes. On modern architectures, it will be important to focus on issues related to the exploitation of the fine grain parallelism and data locality present in this algorithm. This specification aims to make that task easier by presenting the essential details of the algorithm in a systematic and language neutral manner while also avoiding the inclusion of implementation details that would likely be specific to a particular type of programming paradigm or platform architecture.

The equations to be solved are the Euler equations of computational fluid dynamics which in conservative form are [3]

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot \rho \boldsymbol{v} = 0 \tag{1.0.1}$$

$$\frac{\partial \rho \boldsymbol{v}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{v}\boldsymbol{v} + P\mathsf{I}) = 0 \tag{1.0.2}$$

$$\frac{\partial E}{\partial t} + \boldsymbol{\nabla} \cdot [(E + P)\boldsymbol{v}] = 0 \tag{1.0.3}$$

where

$\rho \equiv$ mass density
$\boldsymbol{v} \equiv$ fluid velocity
$E \equiv$ total energy density
$P \equiv$ total pressure
$\mathsf{I} \equiv$ unit tensor

Symbols in bold regular type are 2D cartesian vectors while symbols in bold Sans Serif type represent 2D second order cartesian tensors. The term $\boldsymbol{v}\boldsymbol{v}$ represents the dyadic or tensor product of the two vectors, $\boldsymbol{v}$. In two spatial dimensions, the Euler equations represent four equations for five unknown variables. A polytropic gas equation of state is used to close these equations and relate P to ρ, E and $\boldsymbol{v}$. This equation of state is

$$E = \frac{P}{\gamma - 1} + \frac{1}{2}\rho v^2 \tag{1.0.4}$$

where γ is the ratio of specific heats and is 5/3 for an ideal gas. Thus, equations 1.0.1, 1.0.2, 1.0.3 and 1.0.4 are the closed set of equations in five variables that will be solved by the algorithm presented in the next section. This algorithm is a two dimensional finite volume Godunov [2] algorithm using Strang splitting [4], explicit time integration and an inexact Riemann solver.

A compact and alternative way to write the Euler equations in Cartesian coordinates is as follows:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} = 0 \tag{1.0.5}$$

where

$$\boldsymbol{U} \equiv \begin{bmatrix} U^D \\ U^U \\ U^V \\ U^P \end{bmatrix} \equiv \begin{bmatrix} \rho \\ \rho v_x \\ \rho v_y \\ E \end{bmatrix} \tag{1.0.6}$$

$$\boldsymbol{F} \equiv \begin{bmatrix} F^D \\ F^U \\ F^V \\ F^P \end{bmatrix} \equiv \begin{bmatrix} \rho v_x \\ \rho v_x^2 + P \\ \rho v_x v_y \\ v_x \left(E + P\right) \end{bmatrix} \tag{1.0.7}$$

$$\boldsymbol{G} \equiv \begin{bmatrix} G^D \\ G^U \\ G^V \\ G^P \end{bmatrix} \equiv \begin{bmatrix} \rho v_y \\ \rho v_y v_x \\ \rho v_y^2 + P \\ v_y \left(E + P\right) \end{bmatrix} \tag{1.0.8}$$

This algorithm also makes use of an additional set of state variables which are referred to as primitive variables. These variables are the following.

$$\boldsymbol{Q} \equiv \begin{bmatrix} Q^D \\ Q^U \\ Q^V \\ Q^P \end{bmatrix} \equiv \begin{bmatrix} \rho \\ v_x \\ v_y \\ P \end{bmatrix} \tag{1.0.9}$$

These primitive variables are used during most of the calculation performed by this algorithm including the Riemann step.

2 Algorithm Description

In this algorithm, the rectangular solution region is of dimension L_x by L_y and is divided into rectangular cells. The x-coordinate is divided into n_x cells of equal spacing such that the width of a cell in the x-coordinate is L_x/n_x. Similarly, the y-coordinate is divided into n_y cells of equal spacing such that the height of a cell in the y-coordinate is L_y/n_y. In order to allow for the enforcement of boundary conditions and accomodate the numerical differencing of the spatial derivatives, two extra layers of cells are included and referenced for each spatial coordinate in the description below. Thus, the reader can imagine a rectangular array of cells where in the x-coordinate there are $n_x + 4$ cells and in the y-coordinate there are $n_y + 4$ cells. In the x-coordinate, the cells located in the problem domain are numbered from $i = 3, ..., n_x + 2$. In the y-coordinate, the cells located in the problem domain are numbered from $j = 3, ..., n_y + 2$. How the cells are labeled and how or whether the guard cells are stored is an implementation detail left to the discretion of the software developer. However, in order to make the following description explicit, the above labeling scheme will be adopted.

To summarize, the computational domain is a rectangular region of cells with $n_x + 4$ cells in the x-coordinate and $n_y + 4$ cells in the y-coordinate. In the x-coordinate, the cells are numbered from $i = 1$ to $n_x + 4$. In the y-coordinate, the cells are numbered from $j = 1$ to $n_y + 4$. Each of the cells has two faces in the x-direction and two faces in the y-direction. Thus in the x-direction, there are $n_x + 5$ faces and in the y-direction there are $n_y + 5$ faces. Faces are labeled with a half-integer index. In the x-coordinate, the right face of a cell is labeled as $i + \frac{1}{2}$ and the left face is labeled as $i - \frac{1}{2}$ where i is the x-coordinate index of the cell. In the y-coordinate, the top face of a cell is labeled $j + \frac{1}{2}$ and the bottom face is labeled $j - \frac{1}{2}$ where j is the y-coordinate index of the cell. Thus, the x-coordinate faces are labeled from $\frac{1}{2}$ to $n_x + 4 + \frac{1}{2}$ and the y-coordinate faces are labeled from $\frac{1}{2}$ to $n_y + 4 + \frac{1}{2}$.

One way to discretize the Euler equations described in equation 1.0.5 is as follows.

$$\boldsymbol{U}^{n+1}_{i,j} = \boldsymbol{U}^{n}_{i,j} - \frac{\Delta t}{\Delta x}\left(\boldsymbol{F}_{i+\frac{1}{2},j} - \boldsymbol{F}_{i-\frac{1}{2},j}\right) - \frac{\Delta t}{\Delta y}\left(\boldsymbol{G}_{i,j+\frac{1}{2}} - \boldsymbol{G}_{i,j-\frac{1}{2}}\right) \tag{2.0.10}$$

This approach is what is referred to as an unsplit method or algorithm. However, we will choose to use a split method or algorithm. In a split method, on odd timesteps we will solve the following two equations.

$$\boldsymbol{U}^{\dagger}_{i,j} = \boldsymbol{U}^{n}_{i,j} - \frac{\Delta t}{\Delta x}\left(\boldsymbol{F}_{i+\frac{1}{2},j} - \boldsymbol{F}_{i-\frac{1}{2},j}\right) \tag{2.0.11}$$

$$\boldsymbol{U}^{n+1}_{i,j} = \boldsymbol{U}^{\dagger}_{i,j} - \frac{\Delta t}{\Delta y}\left(\boldsymbol{G}_{i,j+\frac{1}{2}} - \boldsymbol{G}_{i,j-\frac{1}{2}}\right) \tag{2.0.12}$$

On even timesteps, we will solve the following two equations.

$$\boldsymbol{U}^{\dagger}_{i,j} = \boldsymbol{U}^{n}_{i,j} - \frac{\Delta t}{\Delta y}\left(\boldsymbol{G}_{i,j+\frac{1}{2}} - \boldsymbol{G}_{i,j-\frac{1}{2}}\right) \tag{2.0.13}$$

$$\boldsymbol{U}^{n+1}_{i,j} = \boldsymbol{U}^{\dagger}_{i,j} - \frac{\Delta t}{\Delta x}\left(\boldsymbol{F}_{i+\frac{1}{2},j} - \boldsymbol{F}_{i-\frac{1}{2},j}\right) \tag{2.0.14}$$

This approach to a split algorithm where the order of solution in the two spatial coordinates is reversed every timestep is referred to as Strang splitting [4] and results in a higher formal order of accuracy for the solution. Note that in equation 2.0.11 $\boldsymbol{F}$ depends on $\boldsymbol{U}^n$ while in equation 2.0.12 $\boldsymbol{G}$ depends on $\boldsymbol{U}^{\dagger}$. In contrast, in equation 2.0.13 $\boldsymbol{G}$ depends on $\boldsymbol{U}^n$ while in equation 2.0.14 $\boldsymbol{F}$ depends on $\boldsymbol{U}^{\dagger}$.

3 CodeHydro Calculation Flow

This section describes the actual flow of the calculation in sufficient detail to serve as a guide for implementing the CodeHydro algorithm. The flow of the calculation is described from a top down perspective. Because of the data dependencies, the algorithm would be implemented in a bottom up fashion. More will be said on this point later.

1. This is the beginning of the computational sweep over cells in the x-coordinate. In this sweep, we will be solving equation 2.0.11.

For $i = 3, ..., n_x + 2$ and $j = 3, ..., n_y + 2$

$$\boldsymbol{U}^{\dagger}_{i,j} = \boldsymbol{U}^n_{i,j} - \frac{\Delta t}{\Delta x}\left(\boldsymbol{F}_{i+\frac{1}{2},j} - \boldsymbol{F}_{i-\frac{1}{2},j}\right) \tag{3.0.15}$$

where $\boldsymbol{U}$ and $\boldsymbol{F}$ were defined above.

2. For the first step, we need $\boldsymbol{U}^0_{i,j}$, Δt, Δx and $\boldsymbol{F}$. $\boldsymbol{U}^0_{i,j}$ is an initial condition and will be discussed later. The value of Δx is determined by user input. The value of Δt varies according to numerical stability and accuracy constraints. Its calculation will also be described later. Note that in the next step or item, the index, i, starts at 2 instead of 3. By allowing the i index to start at 2, values are specified for both $\boldsymbol{F}_{i+\frac{1}{2},j}$ and $\boldsymbol{F}_{i-\frac{1}{2},j}$. This is because $\boldsymbol{F}_{i+\frac{1}{2},j}$ for $i = 2$ is also equal to $\boldsymbol{F}_{i-\frac{1}{2},j}$ for $i = 3$. So there is no need to add extra complexity to the description by explicitly providing a formula for $\boldsymbol{F}_{i-\frac{1}{2},j}$. We compute these needed quantities as follows.

For $i = 2, ..., n_x + 2$ and $j = 3, ..., n_y + 2$

$$F^D_{i+\frac{1}{2},j} = Q^D_{i+\frac{1}{2},j} Q^U_{i+\frac{1}{2},j} \tag{3.0.16}$$

$$F^U_{i+\frac{1}{2},j} = F^D_{i+\frac{1}{2},j} Q^U_{i+\frac{1}{2},j} + Q^P_{i+\frac{1}{2},j} \tag{3.0.17}$$

$$F^V_{i+\frac{1}{2},j} = F^D_{i+\frac{1}{2},j} Q^V_{i+\frac{1}{2},j} \tag{3.0.18}$$

$$F^P_{i+\frac{1}{2},j} = Q^U_{i+\frac{1}{2},j}\left\{\frac{1}{\gamma - 1} Q^P_{i+\frac{1}{2},j} + \frac{1}{2} Q^D_{i+\frac{1}{2},j}\left[\left(Q^U_{i+\frac{1}{2},j}\right)^2 + \left(Q^V_{i+\frac{1}{2},j}\right)^2\right] + Q^P_{i+\frac{1}{2},j}\right\} \tag{3.0.19}$$

3. For the previous step, we need values for γ and $\boldsymbol{Q}$ where $\boldsymbol{Q} \equiv \left(Q^D, Q^U, Q^V, Q^P\right)$. The value of γ is determined by user input. The value of $\boldsymbol{Q}$ is computed using a Riemann solver as follows.

For $i = 2, ..., n_x + 2$ and $j = 3, ..., n_y + 2$

$$Q^D_{i+\frac{1}{2},j} = f_{i+\frac{1}{2},j} r^*_{i+\frac{1}{2},j} + \left(1 - f_{i+\frac{1}{2},j}\right) r^o_{i+\frac{1}{2},j} \tag{3.0.20}$$

$$Q^U_{i+\frac{1}{2},j} = f_{i+\frac{1}{2},j} u^*_{i+\frac{1}{2},j} + \left(1 - f_{i+\frac{1}{2},j}\right) u^o_{i+\frac{1}{2},j} \tag{3.0.21}$$

$$Q^P_{i+\frac{1}{2},j} = f_{i+\frac{1}{2},j} p^*_{i+\frac{1}{2},j} + \left(1 - f_{i+\frac{1}{2},j}\right) p^o_{i+\frac{1}{2},j} \tag{3.0.22}$$

`if` $\left(u^*_{i+\frac{1}{2},j} \geq 0\right)$

$$Q^V_{i+\frac{1}{2},j} = Q^{LV}_{i+\frac{1}{2},j} \tag{3.0.23}$$

`else`

$$Q^V_{i+\frac{1}{2},j} = Q^{RV}_{i+\frac{1}{2},j} \tag{3.0.24}$$

4. For step 3, we need values for $f, r^*, r^o, u^*, u^o, p^*, p^o, Q^{LV}$ and Q^{RV}. We compute these needed quantities as follows.

For $i = 2, ..., n_x + 2$ and $j = 3, ..., n_y + 2$

$$w^L_{i+\frac{1}{2},j} = \sqrt{c^L_{i+\frac{1}{2},j}} \tag{3.0.25}$$

$$w^R_{i+\frac{1}{2},j} = \sqrt{c^R_{i+\frac{1}{2},j}} \tag{3.0.26}$$

$$p^o_{i+\frac{1}{2},j} = \mathtt{max}\left(\frac{w^R_{i+\frac{1}{2},j}p^L_{i+\frac{1}{2},j} + w^L_{i+\frac{1}{2},j}p^R_{i+\frac{1}{2},j} + w^L_{i+\frac{1}{2},j}w^R_{i+\frac{1}{2},j}\left(u^L_{i+\frac{1}{2},j} - u^R_{i+\frac{1}{2},j}\right)}{w^L_{i+\frac{1}{2},j} + w^R_{i+\frac{1}{2},j}}, 0\right) \tag{3.0.27}$$

`do` $i = 1, \mathtt{nr\_it\_lim}$

$$z^L_{i+\frac{1}{2},j} = \left[c^L_{i+\frac{1}{2},j}\left(1 + \frac{\gamma+1}{2\gamma}\frac{p^o_{i+\frac{1}{2},j} - p^L_{i+\frac{1}{2},j}}{p^L_{i+\frac{1}{2},j}}\right)\right]^{0.5} \tag{3.0.28}$$

$$z^R_{i+\frac{1}{2},j} = \left[c^R_{i+\frac{1}{2},j}\left(1 + \frac{\gamma+1}{2\gamma}\frac{p^o_{i+\frac{1}{2},j} - p^R_{i+\frac{1}{2},j}}{p^R_{i+\frac{1}{2},j}}\right)\right]^{0.5} \tag{3.0.29}$$

$$q^L_{i+\frac{1}{2},j} = \frac{2\left(z^L_{i+\frac{1}{2},j}\right)^3}{\left(z^L_{i+\frac{1}{2},j}\right)^2 + c^L_{i+\frac{1}{2},j}} \tag{3.0.30}$$

$$q^R_{i+\frac{1}{2},j} = \frac{2\left(z^R_{i+\frac{1}{2},j}\right)^3}{\left(z^R_{i+\frac{1}{2},j}\right)^2 + c^R_{i+\frac{1}{2},j}} \tag{3.0.31}$$

$$v^L_{i+\frac{1}{2},j} = u^L_{i+\frac{1}{2},j} - \frac{p^o_{i+\frac{1}{2},j} - p^L_{i+\frac{1}{2},j}}{z^L_{i+\frac{1}{2},j}} \tag{3.0.32}$$

$$v^R_{i+\frac{1}{2},j} = u^R_{i+\frac{1}{2},j} + \frac{p^o_{i+\frac{1}{2},j} - p^R_{i+\frac{1}{2},j}}{z^R_{i+\frac{1}{2},j}} \tag{3.0.33}$$

$$\delta p_{i+\frac{1}{2},j} = \mathtt{max}\left[\frac{q^L_{i+\frac{1}{2},j}q^R_{i+\frac{1}{2},j}}{q^L_{i+\frac{1}{2},j} + q^R_{i+\frac{1}{2},j}}\left(v^L_{i+\frac{1}{2},j} - v^R_{i+\frac{1}{2},j}\right), -p^o_{i+\frac{1}{2},j}\right] \tag{3.0.34}$$

$$p^o_{i+\frac{1}{2},j} = p^o_{i+\frac{1}{2},j} + \delta p_{i+\frac{1}{2},j} \tag{3.0.35}$$

$$\texttt{if}\left(\left|\frac{\delta p_{i+\frac{1}{2},j}}{p^o_{i+\frac{1}{2},j}+r_s p_s}\right| \leq \texttt{nr\_tol}\right) p^*_{i+\frac{1}{2},j} = p^o_{i+\frac{1}{2},j} \tag{3.0.36}$$

`end do`

$$w^L_{i+\frac{1}{2},j} = \left[c^L_{i+\frac{1}{2},j}\left(1+\frac{\gamma+1}{2\gamma}\frac{p^*_{i+\frac{1}{2},j}-p^L_{i+\frac{1}{2},j}}{p^L_{i+\frac{1}{2},j}}\right)\right]^{0.5} \tag{3.0.37}$$

$$w^R_{i+\frac{1}{2},j} = \left[c^R_{i+\frac{1}{2},j}\left(1+\frac{\gamma+1}{2\gamma}\frac{p^*_{i+\frac{1}{2},j}-p^R_{i+\frac{1}{2},j}}{p^R_{i+\frac{1}{2},j}}\right)\right]^{0.5} \tag{3.0.38}$$

$$u^*_{i+\frac{1}{2},j} = \frac{1}{2}\left[u^L_{i+\frac{1}{2},j}+\frac{p^L_{i+\frac{1}{2},j}-p^*_{i+\frac{1}{2},j}}{w^L_{i+\frac{1}{2},j}}+u^R_{i+\frac{1}{2},j}-\frac{p^R_{i+\frac{1}{2},j}-p^*_{i+\frac{1}{2},j}}{w^R_{i+\frac{1}{2},j}}\right] \tag{3.0.39}$$

$\texttt{if}\left(u^*_{i+\frac{1}{2},j}<0\right)$

$$r^o_{i+\frac{1}{2},j} = r^R_{i+\frac{1}{2},j} \tag{3.0.40}$$

$$u^o_{i+\frac{1}{2},j} = u^R_{i+\frac{1}{2},j} \tag{3.0.41}$$

$$p^o_{i+\frac{1}{2},j} = p^R_{i+\frac{1}{2},j} \tag{3.0.42}$$

$$w^o_{i+\frac{1}{2},j} = w^R_{i+\frac{1}{2},j} \tag{3.0.43}$$

`else`

$$r^o_{i+\frac{1}{2},j} = r^L_{i+\frac{1}{2},j} \tag{3.0.44}$$

$$u^o_{i+\frac{1}{2},j} = u^L_{i+\frac{1}{2},j} \tag{3.0.45}$$

$$p^o_{i+\frac{1}{2},j} = p^L_{i+\frac{1}{2},j} \tag{3.0.46}$$

$$w^o_{i+\frac{1}{2},j} = w^L_{i+\frac{1}{2},j} \tag{3.0.47}$$

$$c^o_{i+\frac{1}{2},j} = \texttt{max}\left\{c_s,\left[\texttt{abs}\left(\frac{\gamma p^o_{i+\frac{1}{2},j}}{r^o_{i+\frac{1}{2},j}}\right)\right]^{0.5}\right\} \tag{3.0.48}$$

$$r^*_{i+\frac{1}{2},j} = \texttt{max}\left[r_s,\frac{r^o_{i+\frac{1}{2},j}}{1+\frac{r^o_{i+\frac{1}{2},j}\left(p^o_{i+\frac{1}{2},j}-p^*_{i+\frac{1}{2},j}\right)}{\left(w^o_{i+\frac{1}{2},j}\right)^2}}\right] \tag{3.0.49}$$

$$c^*_{i+\frac{1}{2},j} = \texttt{max}\left\{c_s,\left[\texttt{abs}\left(\frac{\gamma p^*_{i+\frac{1}{2},j}}{r^*_{i+\frac{1}{2},j}}\right)\right]^{0.5}\right\} \tag{3.0.50}$$

`if` $\left(u^{*}_{i+\frac{1}{2},j} < 0\right)$

$$S^{O}_{i+\frac{1}{2},j} = c^{o}_{i+\frac{1}{2},j} + u^{o}_{i+\frac{1}{2},j} \tag{3.0.51}$$

$$S^{I}_{i+\frac{1}{2},j} = c^{*}_{i+\frac{1}{2},j} + u^{*}_{i+\frac{1}{2},j} \tag{3.0.52}$$

$$u^{s}_{i+\frac{1}{2},j} = \frac{w^{o}_{i+\frac{1}{2},j}}{r^{o}_{i+\frac{1}{2},j}} + u^{o}_{i+\frac{1}{2},j} \tag{3.0.53}$$

`else`

$$S^{O}_{i+\frac{1}{2},j} = c^{o}_{i+\frac{1}{2},j} - u^{o}_{i+\frac{1}{2},j} \tag{3.0.54}$$

$$S^{I}_{i+\frac{1}{2},j} = c^{*}_{i+\frac{1}{2},j} - u^{*}_{i+\frac{1}{2},j} \tag{3.0.55}$$

$$u^{s}_{i+\frac{1}{2},j} = \frac{w^{o}_{i+\frac{1}{2},j}}{r^{o}_{i+\frac{1}{2},j}} - u^{o}_{i+\frac{1}{2},j} \tag{3.0.56}$$

`if` $\left(p^{*}_{i+\frac{1}{2},j} \geq p^{o}_{i+\frac{1}{2},j}\right)$

$$S^{O}_{i+\frac{1}{2},j} = u^{s}_{i+\frac{1}{2},j} \tag{3.0.57}$$

$$S^{I}_{i+\frac{1}{2},j} = u^{s}_{i+\frac{1}{2},j} \tag{3.0.58}$$

$$S^{cr}_{i+\frac{1}{2},j} = \texttt{max}\left[S^{O}_{i+\frac{1}{2},j} - S^{I}_{i+\frac{1}{2},j}, c_s + \texttt{abs}\left(S^{O}_{i+\frac{1}{2},j} + S^{I}_{i+\frac{1}{2},j}\right)\right] \tag{3.0.59}$$

$$f_{i+\frac{1}{2},j} = \texttt{max}\left\{0, \texttt{min}\left[1, \frac{1}{2}\left(1 + \frac{S^{O}_{i+\frac{1}{2},j} + S^{I}_{i+\frac{1}{2},j}}{S^{cr}_{i+\frac{1}{2},j}}\right)\right]\right\} \tag{3.0.60}$$

$$\texttt{if}\left(S^{O}_{i+\frac{1}{2},j} < 0\right) f_{i+\frac{1}{2},j} = 0.0 \tag{3.0.61}$$

$$\texttt{if}\left(S^{I}_{i+\frac{1}{2},j} > 0\right) f_{i+\frac{1}{2},j} = 1.0 \tag{3.0.62}$$

5. For step 4, we need values for $c^L, c^R, p^L, p^R, u^L, u^R, r^L, r^R, r_s, p_s, c_s,$ `nr_it_lim` and `nr_tol`. We also still need values for Q^{LV} and Q^{RV} for step 3. The values of $r_s, c_s,$ `nr_it_lim` and `nr_tol` are determined from user input. We compute the remaining needed quantities as follows.

For $i = 2, ..., n_x + 2$ and $j = 3, ..., n_y + 2$

$$p_s = \frac{c_s^2}{\gamma} \tag{3.0.63}$$

$$r^L_{i+\frac{1}{2},j} = \texttt{max}\left(r_s, Q^{LD}_{i+\frac{1}{2},j}\right) \tag{3.0.64}$$
$$u^L_{i+\frac{1}{2},j} = Q^{LU}_{i+\frac{1}{2},j} \tag{3.0.65}$$
$$p^L_{i+\frac{1}{2},j} = \texttt{max}\left(p_s r^L_{i+\frac{1}{2},j}, Q^{LP}_{i+\frac{1}{2},j}\right) \tag{3.0.66}$$
$$c^L_{i+\frac{1}{2},j} = \gamma p^L_{i+\frac{1}{2},j} r^L_{i+\frac{1}{2},j} \tag{3.0.67}$$

$$r^R_{i+\frac{1}{2},j} = \texttt{max}\left(r_s, Q^{RD}_{i+\frac{1}{2},j}\right) \tag{3.0.68}$$
$$u^R_{i+\frac{1}{2},j} = Q^{RU}_{i+\frac{1}{2},j} \tag{3.0.69}$$
$$p^R_{i+\frac{1}{2},j} = \texttt{max}\left(p_s r^R_{i+\frac{1}{2},j}, Q^{RP}_{i+\frac{1}{2},j}\right) \tag{3.0.70}$$
$$c^R_{i+\frac{1}{2},j} = \gamma p^R_{i+\frac{1}{2},j} r^R_{i+\frac{1}{2},j} \tag{3.0.71}$$

6. For step 5, we need values for $Q^{LD}, Q^{LU}, Q^{LP}, Q^{RD}, Q^{RU}$ and Q^{RP}. We also still need values for Q^{LV} and Q^{RV} for step 3. We compute these needed quantities as follows.

For $i = 2, ..., n_x + 3$ and $j = 3, ..., n_y + 2$

$$Q^{LD}_{i+\frac{1}{2},j} = Q^{mD}_{i,j} \tag{3.0.72}$$
$$Q^{LU}_{i+\frac{1}{2},j} = Q^{mU}_{i,j} \tag{3.0.73}$$
$$Q^{LV}_{i+\frac{1}{2},j} = Q^{mV}_{i,j} \tag{3.0.74}$$
$$Q^{LP}_{i+\frac{1}{2},j} = Q^{mP}_{i,j} \tag{3.0.75}$$

$$Q^{RD}_{i+\frac{1}{2},j} = Q^{pD}_{i+1,j} \tag{3.0.76}$$
$$Q^{RU}_{i+\frac{1}{2},j} = Q^{pU}_{i+1,j} \tag{3.0.77}$$
$$Q^{RV}_{i+\frac{1}{2},j} = Q^{pV}_{i+1,j} \tag{3.0.78}$$
$$Q^{RP}_{i+\frac{1}{2},j} = Q^{pP}_{i+1,j} \tag{3.0.79}$$

7. For step 6, we need values for $Q^{mD}, Q^{mU}, Q^{mV}, Q^{mP}, Q^{pD}, Q^{pU}, Q^{pV}$ and Q^{pP}. These are cell centered variables. We compute these needed quantities as follows.

For $i = 2, ..., n_x + 3$ and $j = 3, ..., n_y + 2$

$$Q^{mD}_{i,j} = q^D_{i,j} - 0.5\left(S^{Lp}_{i,j}\alpha^p_{i,j} + S^{Lm}_{i,j}\alpha^m_{i,j} + S^{L0}_{i,j}\alpha^{0r}_{i,j}\right) \tag{3.0.80}$$
$$Q^{mU}_{i,j} = q^U_{i,j} - 0.5\left(S^{Lp}_{i,j}\alpha^p_{i,j} - S^{Lm}_{i,j}\alpha^m_{i,j}\right)\frac{c_{i,j}}{q^D_{i,j}} \tag{3.0.81}$$
$$Q^{mV}_{i,j} = q^V_{i,j} - 0.5 S^{L0}_{i,j}\alpha^{0v}_{i,j} \tag{3.0.82}$$
$$Q^{mP}_{i,j} = q^P_{i,j} - 0.5\left(S^{Lp}_{i,j}\alpha^p_{i,j} + S^{Lm}_{i,j}\alpha^m_{i,j}\right)c^2_{i,j} \tag{3.0.83}$$

$$Q^{pD}_{i,j} = q^{D}_{i,j} - 0.5\left(S^{Rp}_{i,j}\alpha^{p}_{i,j} + S^{Rm}_{i,j}\alpha^{m}_{i,j} + S^{R0}_{i,j}\alpha^{0r}_{i,j}\right) \tag{3.0.84}$$

$$Q^{pU}_{i,j} = q^{U}_{i,j} - 0.5\left(S^{Rp}_{i,j}\alpha^{p}_{i,j} - S^{Rm}_{i,j}\alpha^{m}_{i,j}\right)\frac{c_{i,j}}{q^{D}_{i,j}} \tag{3.0.85}$$

$$Q^{pV}_{i,j} = q^{V}_{i,j} - 0.5 S^{R0}_{i,j}\alpha^{0v}_{i,j} \tag{3.0.86}$$

$$Q^{pP}_{i,j} = q^{P}_{i,j} - 0.5\left(S^{Rp}_{i,j}\alpha^{p}_{i,j} + S^{Rm}_{i,j}\alpha^{m}_{i,j}\right)c^{2}_{i,j} \tag{3.0.87}$$

`if` $\left(q^{U}_{i,j} - c_{i,j} \le Z_L\right)$

$$S^{Lm}_{i,j} = -\beta \tag{3.0.88}$$

`else`

$$S^{Lm}_{i,j} = \left(q^{U}_{i,j} - c_{i,j}\right)\frac{\Delta t}{\Delta x} - 1.0 \tag{3.0.89}$$

`if` $\left(q^{U}_{i,j} + c_{i,j} \le Z_L\right)$

$$S^{Lp}_{i,j} = -\beta \tag{3.0.90}$$

`else`

$$S^{Lp}_{i,j} = \left(q^{U}_{i,j} + c_{i,j}\right)\frac{\Delta t}{\Delta x} - 1.0 \tag{3.0.91}$$

`if` $\left(q^{U}_{i,j} \le Z_L\right)$

$$S^{L0}_{i,j} = -\beta \tag{3.0.92}$$

`else`

$$S^{L0}_{i,j} = q^{U}_{i,j}\frac{\Delta t}{\Delta x} - 1.0 \tag{3.0.93}$$

`if` $\left(q^{U}_{i,j} - c_{i,j} \ge Z_R\right)$

$$S^{Rm}_{i,j} = \beta \tag{3.0.94}$$

`else`

$$S^{Rm}_{i,j} = \left(q^{U}_{i,j} - c_{i,j}\right)\frac{\Delta t}{\Delta x} + 1.0 \tag{3.0.95}$$

`if` $\left(q^{U}_{i,j} + c_{i,j} \ge Z_R\right)$

$$S^{Rp}_{i,j} = \beta \tag{3.0.96}$$

`else`

$$S^{Rp}_{i,j} = \left(q^{U}_{i,j} + c_{i,j}\right)\frac{\Delta t}{\Delta x} + 1.0 \tag{3.0.97}$$

`if` $\left(q^{U}_{i,j} \le Z_R\right)$

$$S^{R0}_{i,j} = \beta \tag{3.0.98}$$

`else`

$$S^{R0}_{i,j} = q^{U}_{i,j}\frac{\Delta t}{\Delta x} + 1.0 \tag{3.0.99}$$

$$\alpha^m_{i,j} = 0.5\left(\frac{\delta q^P_{i,j}}{q^D_{i,j}c_{i,j}} - \delta q^U_{i,j}\right)\frac{q^D_{i,j}}{c_{i,j}} \tag{3.0.100}$$

$$\alpha^p_{i,j} = 0.5\left(\frac{\delta q^P_{i,j}}{q^D_{i,j}c_{i,j}} + \delta q^U_{i,j}\right)\frac{q^D_{i,j}}{c_{i,j}} \tag{3.0.101}$$

$$\alpha^{0r}_{i,j} = \delta q^D_{i,j} - \frac{\delta q^P_{i,j}}{c^2_{i,j}} \tag{3.0.102}$$

$$\alpha^{0v}_{i,j} = \delta q^V_{i,j} \tag{3.0.103}$$

$\texttt{if}\ (\texttt{scheme} = \texttt{muscl})$

$$z_L = -100\frac{\Delta x}{\Delta t} \tag{3.0.104}$$

$$z_R = 100\frac{\Delta x}{\Delta t} \tag{3.0.105}$$

$$\beta = 1.0 \tag{3.0.106}$$

$\texttt{else if}\ (\texttt{scheme} = \texttt{plmde})$

$$z_L = 0.0 \tag{3.0.107}$$

$$z_R = 0.0 \tag{3.0.108}$$

$$\beta = 1.0 \tag{3.0.109}$$

$\texttt{else if}\ (\texttt{scheme} = \texttt{collela})$

$$z_L = 0.0 \tag{3.0.110}$$

$$z_R = 0.0 \tag{3.0.111}$$

$$\beta = 0.0 \tag{3.0.112}$$

$\texttt{if}\ (\texttt{order} = 1)$

$$\delta q^D_{ij} = 0 \tag{3.0.113}$$

$$\delta q^U_{ij} = 0 \tag{3.0.114}$$

$$\delta q^V_{ij} = 0 \tag{3.0.115}$$

$$\delta q^P_{ij} = 0 \tag{3.0.116}$$

$\texttt{else}$

$\texttt{if}\ \left(d^{LD}_{ij} d^{RD}_{ij} \le 0\right)$

$$\delta q^D_{ij} = 0 \tag{3.0.117}$$

$\texttt{else}$

$$\delta q^D_{ij} = \epsilon_D \texttt{min}\left\{\texttt{min}\left[|d^{LD}_{ij}|, |d^{RD}_{ij}|\right], |d^{CD}_{ij}|\right\} \tag{3.0.118}$$

$\texttt{if}\ \left(d^{LU}_{ij} d^{RU}_{ij} \le 0\right)$

$$\delta q^U_{ij} = 0 \tag{3.0.119}$$

$\texttt{else}$

$$\delta q^U_{ij} = \epsilon_U \texttt{min}\left\{\texttt{min}\left[|d^{LU}_{ij}|, |d^{RU}_{ij}|\right], |d^{CU}_{ij}|\right\} \tag{3.0.120}$$

$$\texttt{if}\ \left(d_{ij}^{LV} d_{ij}^{RV} \leq 0\right)$$
$$\delta q_{ij}^{V} = 0 \tag{3.0.121}$$
$$\texttt{else}$$
$$\delta q_{ij}^{V} = \epsilon_V \texttt{min}\left\{\texttt{min}\left[|d_{ij}^{LV}|, |d_{ij}^{RV}|\right], |d_{ij}^{CV}|\right\} \tag{3.0.122}$$

$$\texttt{if}\ \left(d_{ij}^{LP} d_{ij}^{RP} \leq 0\right)$$
$$\delta q_{ij}^{P} = 0 \tag{3.0.123}$$
$$\texttt{else}$$
$$\delta q_{ij}^{P} = \epsilon_P \texttt{min}\left\{\texttt{min}\left[|d_{ij}^{LP}|, |d_{ij}^{RP}|\right], |d_{ij}^{CP}|\right\} \tag{3.0.124}$$

$$\texttt{if}\ \left(d_{ij}^{CD} \geq 0\right)$$
$$\epsilon_D = 1.0 \tag{3.0.125}$$
$$\texttt{else}$$
$$\epsilon_D = -1.0 \tag{3.0.126}$$

$$\texttt{if}\ \left(d_{ij}^{CU} \geq 0\right)$$
$$\epsilon_U = 1.0 \tag{3.0.127}$$
$$\texttt{else}$$
$$\epsilon_U = -1.0 \tag{3.0.128}$$

$$\texttt{if}\ \left(d_{ij}^{CV} \geq 0\right)$$
$$\epsilon_V = 1.0 \tag{3.0.129}$$
$$\texttt{else}$$
$$\epsilon_V = -1.0 \tag{3.0.130}$$

$$\texttt{if}\ \left(d_{ij}^{CP} \geq 0\right)$$
$$\epsilon_P = 1.0 \tag{3.0.131}$$
$$\texttt{else}$$
$$\epsilon_P = -1.0 \tag{3.0.132}$$

$$d_{ij}^{CD} = 0.5\left(q_{i+1,j}^{D} - q_{i-1,j}^{D}\right) \tag{3.0.133}$$
$$d_{ij}^{CU} = 0.5\left(q_{i+1,j}^{U} - q_{i-1,j}^{U}\right) \tag{3.0.134}$$
$$d_{ij}^{CV} = 0.5\left(q_{i+1,j}^{V} - q_{i-1,j}^{V}\right) \tag{3.0.135}$$
$$d_{ij}^{CP} = 0.5\left(q_{i+1,j}^{P} - q_{i-1,j}^{P}\right) \tag{3.0.136}$$

$$d_{ij}^{LD} = T_s\left(q_{ij}^{D} - q_{i-1,j}^{D}\right) \tag{3.0.137}$$
$$d_{ij}^{LU} = T_s\left(q_{ij}^{U} - q_{i-1,j}^{U}\right) \tag{3.0.138}$$
$$d_{ij}^{LV} = T_s\left(q_{ij}^{V} - q_{i-1,j}^{V}\right) \tag{3.0.139}$$
$$d_{ij}^{LP} = T_s\left(q_{ij}^{P} - q_{i-1,j}^{P}\right) \tag{3.0.140}$$

$$d_{ij}^{RD} = T_s \left(q_{i+1,j}^{D} - q_{ij}^{D}\right) \tag{3.0.141}$$
$$d_{ij}^{RU} = T_s \left(q_{i+1,j}^{U} - q_{ij}^{U}\right) \tag{3.0.142}$$
$$d_{ij}^{RV} = T_s \left(q_{i+1,j}^{V} - q_{ij}^{V}\right) \tag{3.0.143}$$
$$d_{ij}^{RP} = T_s \left(q_{i+1,j}^{P} - q_{ij}^{P}\right) \tag{3.0.144}$$

8. For step 7, we need values for $q^D, q^U, q^V, q^P, c, T_s,$ `scheme` and `order`. $T_s,$ `scheme` and `order` are input parameters. The remaining values above are cell centered values. Also note that we need values for q^D, q^U, q^V and q^V over a larger range of values of the i index. Note that the values for $\boldsymbol{q}$ are the primitive variables described in Section 1. The conversion from conservative variables to primitive variables is accomplished as follows.

For $i = 1, ..., n_x + 4$ and $j = 3, ..., n_y + 2$

$$q_{ij}^{D} = \mathtt{max}\left(r_s, U_{ij}^{nD}\right) \tag{3.0.145}$$
$$q_{ij}^{U} = \frac{U_{ij}^{nU}}{q_{ij}^{D}} \tag{3.0.146}$$
$$q_{ij}^{V} = \frac{U_{ij}^{nV}}{q_{ij}^{D}} \tag{3.0.147}$$
$$q_{ij}^{P} = \mathtt{max}\left[P_s q_{ij}^{D}, (\gamma - 1)\, q_{ij}^{D} e_{ij}\right] \tag{3.0.148}$$

$$c_{ij} = \left(\frac{\gamma q_{ij}^{P}}{q_{ij}^{D}}\right)^{0.5} \tag{3.0.149}$$

$$e_{ij} = \frac{U_{ij}^{nP}}{q_{ij}^{D}} - 0.5\left[\left(q_{ij}^{U}\right)^2 + \left(q_{ij}^{U}\right)^2\right] \tag{3.0.150}$$

$$P_s = \frac{c_s^2}{\gamma} \tag{3.0.151}$$

9. At this point everything has been specified to allow the calculation to proceed except for the following three considerations. First, the values of $U_{ij}^{0D}, U_{ij}^{0U}, U_{ij}^{0V}$ and U_{ij}^{0P} need to be specified for $i = 3, ..., n_x + 2$ and $j = 3, ..., n_y + 2$. These are the initial conditions. Second, we need values of $U_{ij}^{nD}, U_{ij}^{nU}, U_{ij}^{nV}$ and U_{ij}^{nP} for the values of $i = 1, 2, n_x + 3, n_x + 4$ and $j = 3, ..., n_y + 2$. These are the boundary conditions in the x-coordinate. Finally, we need to know how to compute the value of the time step. First, we will specify how to compute the value of the time step.

10. For the calculation specified so far, it remains to specify how to compute the time step, Δt. The timestep is computed from a Courant limit that is based on the sound speed and fluid velocity in each coordinate direction. It is computed as follows at the beginning of a timestep.

$$\texttt{if } n = 0 \tag{3.0.152}$$

$$\Delta t = 0.5 \frac{C_f \texttt{min}\,(\Delta x, \Delta y)}{\texttt{max}\,(C_x, C_y, c_s)} \tag{3.0.153}$$

$$\texttt{else} \tag{3.0.154}$$

$$\Delta t = \frac{C_f \texttt{min}\,(\Delta x, \Delta y)}{\texttt{max}\,(C_x, C_y, c_s)} \tag{3.0.155}$$

where C_f, c_s and Δx are input parameters and n labels the last timestep. When $n = 0$, we are at the beginning of the simulation and the state of the previous timestep is represented by the initial conditions. Note that there is a single value of the timestep, Δt, which is used for every cell location and for the mesh sweep in each of the two coordinates. The timestep is only computed after completion of the pass in each of the two coordinates.

For $i = 3, ..., n_x + 2$ and $j = 3, ..., n_y + 2$

$$C_x = \texttt{maxval}\left(C_{ij} + |q^U_{ij}|\right) \tag{3.0.156}$$

$$C_y = \texttt{maxval}\left(C_{ij} + |q^V_{ij}|\right) \tag{3.0.157}$$

$$C_{ij} = \left(\frac{\gamma q^P_{ij}}{q^D_{ij}}\right)^{0.5} \tag{3.0.158}$$

$$q^D_{ij} = \texttt{max}\left(r_s, U^{nD}_{ij}\right) \tag{3.0.159}$$

$$q^U_{ij} = \frac{U^{nU}_{ij}}{q^D_{ij}} \tag{3.0.160}$$

$$q^V_{ij} = \frac{U^{nV}_{ij}}{q^D_{ij}} \tag{3.0.161}$$

$$q^P_{ij} = \texttt{max}\left[P_s q^D_{ij}, (\gamma - 1)\, q^D_{ij} e_{ij}\right] \tag{3.0.162}$$

$$e_{ij} = \frac{U^{nP}_{ij}}{q^D_{ij}} - 0.5\left[\left(q^U_{ij}\right)^2 + \left(q^V_{ij}\right)^2\right] \tag{3.0.163}$$

$$P_s = \frac{c_s^2}{\gamma} \tag{3.0.164}$$

11. A variety of boundary conditions can be supported but three will be provided in this specification. Using the indexing scheme adopted for this specification, these three types of boundary conditions are specified as follows where `bc_left` is an input variable that selects the boundary condition for the left boundary and `bc_right` is an input variable that selects the boundary condition for the right boundary.

For $i = 1, 2$ and $j = 3, ..., n_y + 2$

`if` $(\texttt{bc\_left} = 1)$

$$U_{ij}^{nD} = U_{5-i,j}^{nD} \quad (3.0.165)$$

$$U_{ij}^{nU} = -U_{5-i,j}^{nU} \quad (3.0.166)$$

$$U_{ij}^{nV} = U_{5-i,j}^{nV} \quad (3.0.167)$$

$$U_{ij}^{nP} = U_{5-i,j}^{nP} \quad (3.0.168)$$

`else if` $(\texttt{bc\_left} = 2)$

$$U_{ij}^{nD} = U_{3,j}^{nD} \quad (3.0.169)$$

$$U_{ij}^{nU} = U_{3,j}^{nU} \quad (3.0.170)$$

$$U_{ij}^{nV} = U_{3,j}^{nV} \quad (3.0.171)$$

$$U_{ij}^{nP} = U_{3,j}^{nP} \quad (3.0.172)$$

`else if` $(\texttt{bc\_left} = 3)$

$$U_{ij}^{nD} = U_{n_x+i,j}^{nD} \quad (3.0.173)$$

$$U_{ij}^{nU} = -U_{n_x+i,j}^{nU} \quad (3.0.174)$$

$$U_{ij}^{nV} = U_{n_x+i,j}^{nV} \quad (3.0.175)$$

$$U_{ij}^{nP} = U_{n_x+i,j}^{nP} \quad (3.0.176)$$

For $i = n_x + 3, n_x + 4$ and $j = 3, ..., n_y + 2$

`if` $(\texttt{bc\_right} = 1)$

$$U_{ij}^{nD} = U_{2n_x+5-i,j}^{nD} \quad (3.0.177)$$

$$U_{ij}^{nU} = -U_{2n_x+5-i,j}^{nU} \quad (3.0.178)$$

$$U_{ij}^{nV} = U_{2n_x+5-i,j}^{nV} \quad (3.0.179)$$

$$U_{ij}^{nP} = U_{2n_x+5-i,j}^{nP} \quad (3.0.180)$$

`else if` $(\texttt{bc\_right} = 2)$

$$U_{ij}^{nD} = U_{n_x+2,j}^{nD} \quad (3.0.181)$$

$$U_{ij}^{nU} = U_{n_x+2,j}^{nU} \quad (3.0.182)$$

$$U_{ij}^{nV} = U_{n_x+2,j}^{nV} \quad (3.0.183)$$

$$U_{ij}^{nP} = U_{n_x+2,j}^{nP} \quad (3.0.184)$$

`else if` $(\texttt{bc\_right} = 3)$

$$U_{ij}^{nD} = U_{i-n_x,j}^{nD} \quad (3.0.185)$$

$$U_{ij}^{nU} = -U_{i-n_x,j}^{nU} \quad (3.0.186)$$

$$U_{ij}^{nV} = U_{i-n_x,j}^{nV} \quad (3.0.187)$$

$$U_{ij}^{nP} = U_{i-n_x,j}^{nP} \quad (3.0.188)$$

12. Finally, in order to begin a calculation, initial values must be specified for $U_{ij}^{0D}, U_{ij}^{0U}, U_{ij}^{0V}$ and U_{ij}^{0P} for $i = 3, ..., n_x + 2$ and $j = 3, ..., n + y + 2$. The initial conditions and boundary conditions determine which problem is being modeled. Note that the initial conditions are only applied once at the beginning of a simulation in contrast to the boundary conditions which are applied at the beginning of each pass or sweep of the mesh in a coordinate direction. Three different sets of initial conditions are specified.

For $i = 3, ..., n_x + 2$ and $j = 3, ..., n_y + 2$

if (problem = jet)

$$U_{ij}^{0D} = 1.0 \quad (3.0.189)$$

$$U_{ij}^{0U} = 0.0 \quad (3.0.190)$$

$$U_{ij}^{0V} = 0.0 \quad (3.0.191)$$

$$U_{ij}^{0P} = \frac{1}{\gamma - 1} \quad (3.0.192)$$

else if (problem = point_explosion)

$$U_{ij}^{0D} = 1.0 \quad (3.0.193)$$

$$U_{ij}^{0U} = 0.0 \quad (3.0.194)$$

$$U_{ij}^{0V} = 0.0 \quad (3.0.195)$$

$$U_{ij}^{0P} = 1.0e - 5 \quad (3.0.196)$$

$$U_{i_e, j_e}^{0P} = \frac{1}{\Delta x \Delta y} \quad (3.0.197)$$

else if (problem = sod)

if $(i < n_x/2 + 3)$

$$U_{ij}^{0D} = 1.0 \quad (3.0.198)$$

$$U_{ij}^{0U} = 0.0 \quad (3.0.199)$$

$$U_{ij}^{0V} = 0.0 \quad (3.0.200)$$

$$U_{ij}^{0P} = \frac{1}{\gamma - 1} \quad (3.0.201)$$

else

$$U_{ij}^{0D} = 0.125 \quad (3.0.202)$$

$$U_{ij}^{0U} = 0.0 \quad (3.0.203)$$

$$U_{ij}^{0V} = 0.0 \quad (3.0.204)$$

$$U_{ij}^{0P} = \frac{0.1}{\gamma - 1} \quad (3.0.205)$$

13. This is the beginning of the computational sweep over cells in the y-coordinate. In this sweep, we will be solving equation 2.0.12. Note as mentioned earlier that $\boldsymbol{G}$ depends on $\boldsymbol{U}^{\dagger}$ and that boundary conditions will get applied to $\boldsymbol{U}^{\dagger}$.

For $i = 3, ..., n_x + 2$ and $j = 3, ..., n_y + 2$

$$\boldsymbol{U}_{i,j}^{n+1} = \boldsymbol{U}_{i,j}^{\dagger} - \frac{\Delta t}{\Delta y} \left(\boldsymbol{G}_{i,j+\frac{1}{2}} - \boldsymbol{G}_{i,j-\frac{1}{2}} \right) \quad (3.0.206)$$

where $\boldsymbol{U}$ and $\boldsymbol{G}$ were defined above.

14. For this first step, we need $\boldsymbol{U}^{\dagger}_{i,j}$, Δt, Δy and $\boldsymbol{G}$. $\boldsymbol{U}^{\dagger}_{i,j}$ is the value of $\boldsymbol{U}$ at the end of the sweep in the previous direction. The value of Δy is determined by user input. The value of Δt varies according to numerical stability and accuracy constraints. It was calculated in step 10 above. Note that we are now referencing face centered quantities in the j index since we are doing a sweep of the mesh in the y-coordinate. We compute these needed quantities as follows.

For $i = 3, ..., n_x + 2$ and $j = 2, ..., n_y + 2$

$$G^D_{i,j+\frac{1}{2}} = Q^D_{i,j+\frac{1}{2}} Q^U_{i,j+\frac{1}{2}} \tag{3.0.207}$$
$$G^U_{i,j+\frac{1}{2}} = G^D_{i,j+\frac{1}{2}} Q^U_{i,j+\frac{1}{2}} + Q^P_{i,j+\frac{1}{2}} \tag{3.0.208}$$
$$G^V_{i,j+\frac{1}{2}} = G^D_{i,j+\frac{1}{2}} Q^V_{i,j+\frac{1}{2}} \tag{3.0.209}$$
$$G^P_{i,j+\frac{1}{2}} = Q^U_{i,j+\frac{1}{2}} \left\{ \frac{1}{\gamma - 1} Q^P_{i,j+\frac{1}{2}} + \frac{1}{2} Q^D_{i,j+\frac{1}{2}} \left[\left(Q^U_{i,j+\frac{1}{2}} \right)^2 + \left(Q^V_{i,j+\frac{1}{2}} \right)^2 \right] + Q^P_{i,j+\frac{1}{2}} \right\} \tag{3.0.210}$$

15. For the previous step, we need values for γ and $\boldsymbol{Q}$ where $\boldsymbol{Q} \equiv \left(Q^D, Q^U, Q^V, Q^P \right)$. The value of γ is determined by user input. The value of $\boldsymbol{Q}$ is computed using a Riemann solver as follows. Note that it is now j, the index for the y-coordinate, that begins at 2 rather than i, the index for the x-coordinate.

For $i = 3, ..., n_x + 2$ and $j = 2, ..., n_y + 2$

$$Q^D_{i,j+\frac{1}{2}} = f_{i,j+\frac{1}{2}} r^*_{i,j+\frac{1}{2}} + \left(1 - f_{i,j+\frac{1}{2}} \right) r^o_{i,j+\frac{1}{2}} \tag{3.0.211}$$
$$Q^U_{i,j+\frac{1}{2}} = f_{i,j+\frac{1}{2}} u^*_{i,j+\frac{1}{2}} + \left(1 - f_{i,j+\frac{1}{2}} \right) u^o_{i,j+\frac{1}{2}} \tag{3.0.212}$$
$$Q^P_{i,j+\frac{1}{2}} = f_{i,j+\frac{1}{2}} p^*_{i,j+\frac{1}{2}} + \left(1 - f_{i,j+\frac{1}{2}} \right) p^o_{i,j+\frac{1}{2}} \tag{3.0.213}$$

`if` $\left(u^*_{i,j+\frac{1}{2}} \geq 0 \right)$

$$Q^V_{i,j+\frac{1}{2}} = Q^{LV}_{i,j+\frac{1}{2}} \tag{3.0.214}$$

`else`

$$Q^V_{i,j+\frac{1}{2}} = Q^{RV}_{i,j+\frac{1}{2}} \tag{3.0.215}$$

16. For step 15, we need values for $f, r^*, r^o, u^*, u^o, p^*, p^o, Q^{LV}$ and Q^{RV}. We compute these needed quantities as follows.

For $i = 3, ..., n_x + 2$ and $j = 2, ..., n_y + 2$

$$w^L_{i,j+\frac{1}{2}} = \sqrt{c^L_{i,j+\frac{1}{2}}} \tag{3.0.216}$$

$$w^R_{i,j+\frac{1}{2}} = \sqrt{c^R_{i,j+\frac{1}{2}}} \tag{3.0.217}$$

$$p^o_{i,j+\frac{1}{2}} = \texttt{max}\left(\frac{w^R_{i,j+\frac{1}{2}}p^L_{i,j+\frac{1}{2}} + w^L_{i,j+\frac{1}{2}}p^R_{i,j+\frac{1}{2}} + w^L_{i,j+\frac{1}{2}}w^R_{i,j+\frac{1}{2}}\left(u^L_{i,j+\frac{1}{2}} - u^R_{i,j+\frac{1}{2}}\right)}{w^L_{i,j+\frac{1}{2}} + w^R_{i,j+\frac{1}{2}}}, 0\right) \tag{3.0.218}$$

`do` $i = 1,$ `nr_it_lim`

$$z^L_{i,j+\frac{1}{2}} = \left[c^L_{i,j+\frac{1}{2}}\left(1 + \frac{\gamma+1}{2\gamma}\frac{p^o_{i,j+\frac{1}{2}} - p^L_{i,j+\frac{1}{2}}}{p^L_{i,j+\frac{1}{2}}}\right)\right]^{0.5} \tag{3.0.219}$$

$$z^R_{i,j+\frac{1}{2}} = \left[c^R_{i,j+\frac{1}{2}}\left(1 + \frac{\gamma+1}{2\gamma}\frac{p^o_{i,j+\frac{1}{2}} - p^R_{i,j+\frac{1}{2}}}{p^R_{i,j+\frac{1}{2}}}\right)\right]^{0.5} \tag{3.0.220}$$

$$q^L_{i,j+\frac{1}{2}} = \frac{2\left(z^L_{i,j+\frac{1}{2}}\right)^3}{\left(z^L_{i,j+\frac{1}{2}}\right)^2 + c^L_{i,j+\frac{1}{2}}} \tag{3.0.221}$$

$$q^R_{i,j+\frac{1}{2}} = \frac{2\left(z^R_{i,j+\frac{1}{2}}\right)^3}{\left(z^R_{i,j+\frac{1}{2}}\right)^2 + c^R_{i,j+\frac{1}{2}}} \tag{3.0.222}$$

$$v^L_{i,j+\frac{1}{2}} = u^L_{i,j+\frac{1}{2}} - \frac{p^o_{i,j+\frac{1}{2}} - p^L_{i,j+\frac{1}{2}}}{z^L_{i,j+\frac{1}{2}}} \tag{3.0.223}$$

$$v^R_{i,j+\frac{1}{2}} = u^R_{i,j+\frac{1}{2}} + \frac{p^o_{i,j+\frac{1}{2}} - p^R_{i,j+\frac{1}{2}}}{z^R_{i,j+\frac{1}{2}}} \tag{3.0.224}$$

$$\delta p_{i,j+\frac{1}{2}} = \texttt{max}\left[\frac{q^L_{i,j+\frac{1}{2}}q^R_{i,j+\frac{1}{2}}}{q^L_{i,j+\frac{1}{2}} + q^R_{i,j+\frac{1}{2}}}\left(v^L_{i,j+\frac{1}{2}} - v^R_{i,j+\frac{1}{2}}\right), -p^o_{i,j+\frac{1}{2}}\right] \tag{3.0.225}$$

$$p^o_{i,j+\frac{1}{2}} = p^o_{i,j+\frac{1}{2}} + \delta p_{i,j+\frac{1}{2}} \tag{3.0.226}$$

$$\texttt{if}\left(\left|\frac{\delta p_{i,j+\frac{1}{2}}}{p^o_{i,j+\frac{1}{2}} + r_s p_s}\right| \leq \texttt{nr\_tol}\right) p^*_{i,j+\frac{1}{2}} = p^o_{i,j+\frac{1}{2}} \tag{3.0.227}$$

`end do`

$$w^L_{i,j+\frac{1}{2}} = \left[c^L_{i,j+\frac{1}{2}}\left(1 + \frac{\gamma+1}{2\gamma}\frac{p^*_{i,j+\frac{1}{2}} - p^L_{i,j+\frac{1}{2}}}{p^L_{i,j+\frac{1}{2}}}\right)\right]^{0.5} \tag{3.0.228}$$

$$w^R_{i,j+\frac{1}{2}} = \left[c^R_{i,j+\frac{1}{2}}\left(1 + \frac{\gamma+1}{2\gamma}\frac{p^*_{i,j+\frac{1}{2}} - p^R_{i,j+\frac{1}{2}}}{p^R_{i,j+\frac{1}{2}}}\right)\right]^{0.5} \tag{3.0.229}$$

$$u^*_{i,j+\frac{1}{2}} = \frac{1}{2}\left[u^L_{i,j+\frac{1}{2}} + \frac{p^L_{i,j+\frac{1}{2}} - p^*_{i,j+\frac{1}{2}}}{w^L_{i,j+\frac{1}{2}}} + u^R_{i,j+\frac{1}{2}} - \frac{p^R_{i,j+\frac{1}{2}} - p^*_{i,j+\frac{1}{2}}}{w^R_{i,j+\frac{1}{2}}}\right] \tag{3.0.230}$$

if $\left(u^*_{i,j+\frac{1}{2}} < 0\right)$

$$r^o_{i,j+\frac{1}{2}} = r^R_{i,j+\frac{1}{2}} \tag{3.0.231}$$

$$u^o_{i,j+\frac{1}{2}} = u^R_{i,j+\frac{1}{2}} \tag{3.0.232}$$

$$p^o_{i,j+\frac{1}{2}} = p^R_{i,j+\frac{1}{2}} \tag{3.0.233}$$

$$w^o_{i,j+\frac{1}{2}} = w^R_{i,j+\frac{1}{2}} \tag{3.0.234}$$

else

$$r^o_{i,j+\frac{1}{2}} = r^L_{i,j+\frac{1}{2}} \tag{3.0.235}$$

$$u^o_{i,j+\frac{1}{2}} = u^L_{i,j+\frac{1}{2}} \tag{3.0.236}$$

$$p^o_{i,j+\frac{1}{2}} = p^L_{i,j+\frac{1}{2}} \tag{3.0.237}$$

$$w^o_{i,j+\frac{1}{2}} = w^L_{i,j+\frac{1}{2}} \tag{3.0.238}$$

$$c^o_{i,j+\frac{1}{2}} = \mathtt{max}\left\{c_s, \left[\mathtt{abs}\left(\frac{\gamma p^o_{i,j+\frac{1}{2}}}{r^o_{i,j+\frac{1}{2}}}\right)\right]^{0.5}\right\} \tag{3.0.239}$$

$$r^*_{i,j+\frac{1}{2}} = \mathtt{max}\left[r_s, \frac{r^o_{i,j+\frac{1}{2}}}{1 + \frac{r^o_{i,j+\frac{1}{2}}\left(p^o_{i,j+\frac{1}{2}} - p^*_{i,j+\frac{1}{2}}\right)}{\left(w^o_{i,j+\frac{1}{2}}\right)^2}}\right] \tag{3.0.240}$$

$$c^*_{i,j+\frac{1}{2}} = \mathtt{max}\left\{c_s, \left[\mathtt{abs}\left(\frac{\gamma p^*_{i,j+\frac{1}{2}}}{r^*_{i,j+\frac{1}{2}}}\right)\right]^{0.5}\right\} \tag{3.0.241}$$

if $\left(u^*_{i,j+\frac{1}{2}} < 0\right)$

$$S^O_{i,j+\frac{1}{2}} = c^o_{i,j+\frac{1}{2}} + u^o_{i,j+\frac{1}{2}} \tag{3.0.242}$$

$$S^I_{i,j+\frac{1}{2}} = c^*_{i,j+\frac{1}{2}} + u^*_{i,j+\frac{1}{2}} \tag{3.0.243}$$

$$u^s_{i,j+\frac{1}{2}} = \frac{w^o_{i,j+\frac{1}{2}}}{r^o_{i,j+\frac{1}{2}}} + u^o_{i,j+\frac{1}{2}} \tag{3.0.244}$$

else

$$S^O_{i,j+\frac{1}{2}} = c^o_{i,j+\frac{1}{2}} - u^o_{i,j+\frac{1}{2}} \tag{3.0.245}$$

$$S^I_{i,j+\frac{1}{2}} = c^*_{i,j+\frac{1}{2}} - u^*_{i,j+\frac{1}{2}} \tag{3.0.246}$$

$$u^s_{i,j+\frac{1}{2}} = \frac{w^o_{i,j+\frac{1}{2}}}{r^o_{i,j+\frac{1}{2}}} - u^o_{i,j+\frac{1}{2}} \tag{3.0.247}$$

$$\texttt{if}\ \left(p^*_{i,j+\frac{1}{2}} \geq p^o_{i,j+\frac{1}{2}}\right)$$

$$S^O_{i,j+\frac{1}{2}} = u^s_{i,j+\frac{1}{2}} \tag{3.0.248}$$

$$S^I_{i,j+\frac{1}{2}} = u^s_{i,j+\frac{1}{2}} \tag{3.0.249}$$

$$S^{cr}_{i,j+\frac{1}{2}} = \texttt{max}\left[S^O_{i,j+\frac{1}{2}} - S^I_{i,j+\frac{1}{2}}, c_s + \texttt{abs}\left(S^O_{i,j+\frac{1}{2}} + S^I_{i,j+\frac{1}{2}}\right)\right] \tag{3.0.250}$$

$$f_{i,j+\frac{1}{2}} = \texttt{max}\left\{0, \texttt{min}\left[1, \frac{1}{2}\left(1 + \frac{S^O_{i,j+\frac{1}{2}} + S^I_{i,j+\frac{1}{2}}}{S^{cr}_{i,j+\frac{1}{2}}}\right)\right]\right\} \tag{3.0.251}$$

$$\texttt{if}\ \left(S^O_{i,j+\frac{1}{2}} < 0\right) f_{i,j+\frac{1}{2}} = 0.0 \tag{3.0.252}$$

$$\texttt{if}\ \left(S^I_{i,j+\frac{1}{2}} > 0\right) f_{i,j+\frac{1}{2}} = 1.0 \tag{3.0.253}$$

17. For step 16, we need values for $c^L, c^R, p^L, p^R, u^L, u^R, r^L, r^R$ and p_s. We also still need values for Q^{LV} and Q^{RV} for step 15. We compute these needed quantities as follows.

For $i = 3, ..., n_x + 2$ and $j = 2, ..., n_y + 2$

$$p_s = \frac{c_s^2}{\gamma} \tag{3.0.254}$$

$$r^L_{i,j+\frac{1}{2}} = \texttt{max}\left(r_s, Q^{LD}_{i,j+\frac{1}{2}}\right) \tag{3.0.255}$$

$$u^L_{i,j+\frac{1}{2}} = Q^{LU}_{i,j+\frac{1}{2}} \tag{3.0.256}$$

$$p^L_{i,j+\frac{1}{2}} = \texttt{max}\left(p_s r^L_{i,j+\frac{1}{2}}, Q^{LP}_{i,j+\frac{1}{2}}\right) \tag{3.0.257}$$

$$c^L_{i,j+\frac{1}{2}} = \gamma p^L_{i,j+\frac{1}{2}} r^L_{i,j+\frac{1}{2}} \tag{3.0.258}$$

$$r^R_{i,j+\frac{1}{2}} = \texttt{max}\left(r_s, Q^{RD}_{i,j+\frac{1}{2}}\right) \tag{3.0.259}$$

$$u^R_{i,j+\frac{1}{2}} = Q^{RU}_{i,j+\frac{1}{2}} \tag{3.0.260}$$

$$p^R_{i,j+\frac{1}{2}} = \texttt{max}\left(p_s r^R_{i,j+\frac{1}{2}}, Q^{RP}_{i,j+\frac{1}{2}}\right) \tag{3.0.261}$$

$$c^R_{i,j+\frac{1}{2}} = \gamma p^R_{i,j+\frac{1}{2}} r^R_{i,j+\frac{1}{2}} \tag{3.0.262}$$

18. For step 17, we need values for $Q^{LD}, Q^{LU}, Q^{LP}, Q^{RD}, Q^{RU}$ and Q^{RP}. We also still need values for Q^{LV} and Q^{RV} for step 15. We compute these needed quantities as follows.

For $i = 3, ..., n_x + 3$ and $j = 2, ..., n_y + 2$

$$Q^{LD}_{i,j+\frac{1}{2}} = Q^{mD}_{i,j} \tag{3.0.263}$$
$$Q^{LU}_{i,j+\frac{1}{2}} = Q^{mU}_{i,j} \tag{3.0.264}$$
$$Q^{LV}_{i,j+\frac{1}{2}} = Q^{mV}_{i,j} \tag{3.0.265}$$
$$Q^{LP}_{i,j+\frac{1}{2}} = Q^{mP}_{i,j} \tag{3.0.266}$$

$$Q^{RD}_{i,j+\frac{1}{2}} = Q^{pD}_{i+1,j} \tag{3.0.267}$$
$$Q^{RU}_{i,j+\frac{1}{2}} = Q^{pU}_{i+1,j} \tag{3.0.268}$$
$$Q^{RV}_{i,j+\frac{1}{2}} = Q^{pV}_{i+1,j} \tag{3.0.269}$$
$$Q^{RP}_{i,j+\frac{1}{2}} = Q^{pP}_{i+1,j} \tag{3.0.270}$$

19. For step 18, we need values for $Q^{mD}, Q^{mU}, Q^{mV}, Q^{mP}, Q^{pD}, Q^{pU}, Q^{pV}$ and Q^{pP}. These are cell centered variables. We compute these needed quantities as follows.

For $i = 3, ..., n_x + 3$ and $j = 2, ..., n_y + 2$

$$Q^{mD}_{i,j} = q^{D}_{i,j} - 0.5\left(S^{Lp}_{i,j}\alpha^{p}_{i,j} + S^{Lm}_{i,j}\alpha^{m}_{i,j} + S^{L0}_{i,j}\alpha^{0r}_{i,j}\right) \tag{3.0.271}$$
$$Q^{mU}_{i,j} = q^{U}_{i,j} - 0.5\left(S^{Lp}_{i,j}\alpha^{p}_{i,j} - S^{Lm}_{i,j}\alpha^{m}_{i,j}\right)\frac{c_{i,j}}{q^{D}_{i,j}} \tag{3.0.272}$$
$$Q^{mV}_{i,j} = q^{V}_{i,j} - 0.5S^{L0}_{i,j}\alpha^{0v}_{i,j} \tag{3.0.273}$$
$$Q^{mP}_{i,j} = q^{P}_{i,j} - 0.5\left(S^{Lp}_{i,j}\alpha^{p}_{i,j} + S^{Lm}_{i,j}\alpha^{m}_{i,j}\right)c^{2}_{i,j} \tag{3.0.274}$$

$$Q^{pD}_{i,j} = q^{D}_{i,j} - 0.5\left(S^{Rp}_{i,j}\alpha^{p}_{i,j} + S^{Rm}_{i,j}\alpha^{m}_{i,j} + S^{R0}_{i,j}\alpha^{0r}_{i,j}\right) \tag{3.0.275}$$
$$Q^{pU}_{i,j} = q^{U}_{i,j} - 0.5\left(S^{Rp}_{i,j}\alpha^{p}_{i,j} - S^{Rm}_{i,j}\alpha^{m}_{i,j}\right)\frac{c_{i,j}}{q^{D}_{i,j}} \tag{3.0.276}$$
$$Q^{pV}_{i,j} = q^{V}_{i,j} - 0.5S^{R0}_{i,j}\alpha^{0v}_{i,j} \tag{3.0.277}$$
$$Q^{pP}_{i,j} = q^{P}_{i,j} - 0.5\left(S^{Rp}_{i,j}\alpha^{p}_{i,j} + S^{Rm}_{i,j}\alpha^{m}_{i,j}\right)c^{2}_{i,j} \tag{3.0.278}$$

`if` $\left(q^{U}_{i,j} - c_{i,j} \le Z_L\right)$

$$S^{Lm}_{i,j} = -\beta \tag{3.0.279}$$

`else`

$$S^{Lm}_{i,j} = \left(q^{U}_{i,j} - c_{i,j}\right)\frac{\Delta t}{\Delta y} - 1.0 \tag{3.0.280}$$

`if` $\left(q^U_{i,j} + c_{i,j} \leq Z_L\right)$

$$S^{Lp}_{i,j} = -\beta \tag{3.0.281}$$

`else`

$$S^{Lp}_{i,j} = \left(q^U_{i,j} + c_{i,j}\right) \frac{\Delta t}{\Delta y} - 1.0 \tag{3.0.282}$$

`if` $\left(q^U_{i,j} \leq Z_L\right)$

$$S^{L0}_{i,j} = -\beta \tag{3.0.283}$$

`else`

$$S^{L0}_{i,j} = q^U_{i,j} \frac{\Delta t}{\Delta y} - 1.0 \tag{3.0.284}$$

`if` $\left(q^U_{i,j} - c_{i,j} \geq Z_R\right)$

$$S^{Rm}_{i,j} = \beta \tag{3.0.285}$$

`else`

$$S^{Rm}_{i,j} = \left(q^U_{i,j} - c_{i,j}\right) \frac{\Delta t}{\Delta y} + 1.0 \tag{3.0.286}$$

`if` $\left(q^U_{i,j} + c_{i,j} \geq Z_R\right)$

$$S^{Rp}_{i,j} = \beta \tag{3.0.287}$$

`else`

$$S^{Rp}_{i,j} = \left(q^U_{i,j} + c_{i,j}\right) \frac{\Delta t}{\Delta y} + 1.0 \tag{3.0.288}$$

`if` $\left(q^U_{i,j} \leq Z_R\right)$

$$S^{R0}_{i,j} = \beta \tag{3.0.289}$$

`else`

$$S^{R0}_{i,j} = q^U_{i,j} \frac{\Delta t}{\Delta y} + 1.0 \tag{3.0.290}$$

$$\alpha^m_{i,j} = 0.5 \left(\frac{\delta q^P_{i,j}}{q^D_{i,j} c_{i,j}} - \delta q^U_{i,j} \right) \frac{q^D_{i,j}}{c_{i,j}} \tag{3.0.291}$$

$$\alpha^p_{i,j} = 0.5 \left(\frac{\delta q^P_{i,j}}{q^D_{i,j} c_{i,j}} + \delta q^U_{i,j} \right) \frac{q^D_{i,j}}{c_{i,j}} \tag{3.0.292}$$

$$\alpha^{0r}_{i,j} = \delta q^D_{i,j} - \frac{\delta q^P_{i,j}}{c^2_{i,j}} \tag{3.0.293}$$

$$\alpha^{0v}_{i,j} = \delta q^V_{i,j} \tag{3.0.294}$$

`if` $(\texttt{scheme} = \texttt{muscl})$

$$z_L = -100\frac{\Delta y}{\Delta t} \tag{3.0.295}$$

$$z_R = 100\frac{\Delta y}{\Delta t} \tag{3.0.296}$$

$$\beta = 1.0 \tag{3.0.297}$$

`else if` $(\texttt{scheme} = \texttt{plmde})$

$$z_L = 0.0 \tag{3.0.298}$$

$$z_R = 0.0 \tag{3.0.299}$$

$$\beta = 1.0 \tag{3.0.300}$$

`else if` $(\texttt{scheme} = \texttt{collela})$

$$z_L = 0.0 \tag{3.0.301}$$

$$z_R = 0.0 \tag{3.0.302}$$

$$\beta = 0.0 \tag{3.0.303}$$

`if` $(\texttt{order} = 1)$

$$\delta q_{ij}^D = 0 \tag{3.0.304}$$

$$\delta q_{ij}^U = 0 \tag{3.0.305}$$

$$\delta q_{ij}^V = 0 \tag{3.0.306}$$

$$\delta q_{ij}^P = 0 \tag{3.0.307}$$

`else`

`if` $\left(d_{ij}^{LD} d_{ij}^{RD} \leq 0\right)$

$$\delta q_{ij}^D = 0 \tag{3.0.308}$$

`else`

$$\delta q_{ij}^D = \epsilon_D \texttt{min}\left\{\texttt{min}\left[|d_{ij}^{LD}|, |d_{ij}^{RD}|\right], |d_{ij}^{CD}|\right\} \tag{3.0.309}$$

`if` $\left(d_{ij}^{LU} d_{ij}^{RU} \leq 0\right)$

$$\delta q_{ij}^U = 0 \tag{3.0.310}$$

`else`

$$\delta q_{ij}^U = \epsilon_U \texttt{min}\left\{\texttt{min}\left[|d_{ij}^{LU}|, |d_{ij}^{RU}|\right], |d_{ij}^{CU}|\right\} \tag{3.0.311}$$

`if` $\left(d_{ij}^{LV} d_{ij}^{RV} \leq 0\right)$

$$\delta q_{ij}^V = 0 \tag{3.0.312}$$

`else`

$$\delta q_{ij}^V = \epsilon_V \texttt{min}\left\{\texttt{min}\left[|d_{ij}^{LV}|, |d_{ij}^{RV}|\right], |d_{ij}^{CV}|\right\} \tag{3.0.313}$$

$$\texttt{if}\ \left(d_{ij}^{LP} d_{ij}^{RP} \leq 0\right)$$
$$\delta q_{ij}^{P} = 0 \tag{3.0.314}$$
$$\texttt{else}$$
$$\delta q_{ij}^{P} = \epsilon_P \texttt{min}\left\{\texttt{min}\left[|d_{ij}^{LP}|, |d_{ij}^{RP}|\right], |d_{ij}^{CP}|\right\} \tag{3.0.315}$$

$$\texttt{if}\ \left(d_{ij}^{CD} \geq 0\right)$$
$$\epsilon_D = 1.0 \tag{3.0.316}$$
$$\texttt{else}$$
$$\epsilon_D = -1.0 \tag{3.0.317}$$

$$\texttt{if}\ \left(d_{ij}^{CU} \geq 0\right)$$
$$\epsilon_U = 1.0 \tag{3.0.318}$$
$$\texttt{else}$$
$$\epsilon_U = -1.0 \tag{3.0.319}$$

$$\texttt{if}\ \left(d_{ij}^{CV} \geq 0\right)$$
$$\epsilon_V = 1.0 \tag{3.0.320}$$
$$\texttt{else}$$
$$\epsilon_V = -1.0 \tag{3.0.321}$$

$$\texttt{if}\ \left(d_{ij}^{CP} \geq 0\right)$$
$$\epsilon_P = 1.0 \tag{3.0.322}$$
$$\texttt{else}$$
$$\epsilon_P = -1.0 \tag{3.0.323}$$

$$d_{ij}^{CD} = 0.5\left(q_{i+1,j}^{D} - q_{i-1,j}^{D}\right) \tag{3.0.324}$$
$$d_{ij}^{CU} = 0.5\left(q_{i+1,j}^{U} - q_{i-1,j}^{U}\right) \tag{3.0.325}$$
$$d_{ij}^{CV} = 0.5\left(q_{i+1,j}^{V} - q_{i-1,j}^{V}\right) \tag{3.0.326}$$
$$d_{ij}^{CP} = 0.5\left(q_{i+1,j}^{P} - q_{i-1,j}^{P}\right) \tag{3.0.327}$$

$$d_{ij}^{LD} = T_s\left(q_{ij}^{D} - q_{i-1,j}^{D}\right) \tag{3.0.328}$$
$$d_{ij}^{LU} = T_s\left(q_{ij}^{U} - q_{i-1,j}^{U}\right) \tag{3.0.329}$$
$$d_{ij}^{LV} = T_s\left(q_{ij}^{V} - q_{i-1,j}^{V}\right) \tag{3.0.330}$$
$$d_{ij}^{LP} = T_s\left(q_{ij}^{P} - q_{i-1,j}^{P}\right) \tag{3.0.331}$$

$$d_{ij}^{RD} = T_s\left(q_{i+1,j}^{D} - q_{ij}^{D}\right) \tag{3.0.332}$$
$$d_{ij}^{RU} = T_s\left(q_{i+1,j}^{U} - q_{ij}^{U}\right) \tag{3.0.333}$$
$$d_{ij}^{RV} = T_s\left(q_{i+1,j}^{V} - q_{ij}^{V}\right) \tag{3.0.334}$$
$$d_{ij}^{RP} = T_s\left(q_{i+1,j}^{P} - q_{ij}^{P}\right) \tag{3.0.335}$$

20. For step 19, we need values for q^D, q^U, q^V, q^P, c and T_s. T_s is an input parameter. The remaining values above are cell centered values. Also note that we need values for q^D, q^U, q^V and q^V over a larger range of values of the j index. Note that the values for $\boldsymbol{q}$ are the primitive variables described in Section 1. The conversion from conservative variables to primitive variables is accomplished as follows.

For $i = 3, ..., n_x + 2$ and $j = 1, ..., n_y + 4$

$$q_{ij}^D = \texttt{max}\left(r_s, U_{ij}^{nD}\right) \tag{3.0.336}$$
$$q_{ij}^U = \frac{U_{ij}^{nU}}{q_{ij}^D} \tag{3.0.337}$$
$$q_{ij}^V = \frac{U_{ij}^{nV}}{q_{ij}^D} \tag{3.0.338}$$
$$q_{ij}^P = \texttt{max}\left[P_s q_{ij}^D, (\gamma - 1)\, q_{ij}^D e_{ij}\right] \tag{3.0.339}$$

$$c_{ij} = \left(\frac{\gamma q_{ij}^P}{q_{ij}^D}\right)^{0.5} \tag{3.0.340}$$
$$e_{ij} = \frac{U_{ij}^{nP}}{q_{ij}^D} - 0.5\left[\left(q_{ij}^U\right)^2 + \left(q_{ij}^U\right)^2\right] \tag{3.0.341}$$

$$P_s = \frac{c_s^2}{\gamma} \tag{3.0.342}$$

21. At this point everything has been specified to allow the calculation to proceed except for the boundary conditions. We need values of $U_{ij}^\dagger, U_{ij}^\dagger, U_{ij}^\dagger$ and $U_{ij}^\dagger$ for the values of $j = 1, 2, n_y+3, n_y+4$ and $i = 3, ..., n_x+2$. These are the boundary conditions in the y-coordinate. A variety of boundary conditions can be supported but the same three will be provided in this specification as for the x-coordinate. Using the indexing scheme adopted for this specification, these three types of boundary conditions are specified as follows.

For $j = 1, 2$ and $i = 3, ..., n_x + 2$

$\texttt{if (bc\_bottom} = 1)$

$$U_{ij}^{nD} = U_{i,5-j}^{nD} \tag{3.0.343}$$
$$U_{ij}^{nU} = -U_{i,5-j}^{nU} \tag{3.0.344}$$
$$U_{ij}^{nV} = U_{i,5-j}^{nV} \tag{3.0.345}$$
$$U_{ij}^{nP} = U_{i,5-j}^{nP} \tag{3.0.346}$$

$\texttt{else if (bc\_bottom} = 2)$

$$U_{ij}^{nD} = U_{i,3}^{nD} \tag{3.0.347}$$
$$U_{ij}^{nU} = U_{i,3}^{nU} \tag{3.0.348}$$
$$U_{ij}^{nV} = U_{i,3}^{nV} \tag{3.0.349}$$
$$U_{ij}^{nP} = U_{i,3}^{nP} \tag{3.0.350}$$
$$\tag{3.0.351}$$

`else if` $(\texttt{bc\_bottom} = 3)$

$$U^{nD}_{ij} = U^{nD}_{i,n_y+j} \quad (3.0.352)$$

$$U^{nU}_{ij} = -U^{nU}_{i,n_y+j} \quad (3.0.353)$$

$$U^{nV}_{ij} = U^{nV}_{i,n_y+j} \quad (3.0.354)$$

$$U^{nP}_{ij} = U^{nP}_{i,n_y+j} \quad (3.0.355)$$

For $j = n_y + 3, n_y + 4$ and $i = 3, ..., n_x + 2$

`if` $(\texttt{bc\_top} = 1)$

$$U^{nD}_{ij} = U^{nD}_{i,2n_y+5-j} \quad (3.0.356)$$

$$U^{nU}_{ij} = -U^{nU}_{i,2n_y+5-j} \quad (3.0.357)$$

$$U^{nV}_{ij} = U^{nV}_{i,2n_y+5-j} \quad (3.0.358)$$

$$U^{nP}_{ij} = U^{nP}_{i,2n_y+5-j} \quad (3.0.359)$$

`else if` $(\texttt{bc\_top} = 2)$

$$U^{nD}_{ij} = U^{nD}_{i,n_y+j} \quad (3.0.360)$$

$$U^{nU}_{ij} = U^{nU}_{i,n_y+j} \quad (3.0.361)$$

$$U^{nV}_{ij} = U^{nV}_{i,n_y+j} \quad (3.0.362)$$

$$U^{nP}_{ij} = U^{nP}_{i,n_y+j} \quad (3.0.363)$$

`else if` $(\texttt{bc\_top} = 3)$

$$U^{nD}_{ij} = U^{nD}_{i,j-n_y} \quad (3.0.364)$$

$$U^{nU}_{ij} = -U^{nU}_{i,j-n_y} \quad (3.0.365)$$

$$U^{nV}_{ij} = U^{nV}_{i,j-n_y} \quad (3.0.366)$$

$$U^{nP}_{ij} = U^{nP}_{i,j-n_y} \quad (3.0.367)$$

This completes the specification of the algorithm used in CodeHydro. Now we will spend some time discussing how to use the details of the specification above.

4 Discussion of Algorithm

As indicated, the previous section provided details on how to implement the algorithm from a top down perspective. However, the algorithm must be executed in a bottom up fashion such that equations 2.0.11, 2.0.12, 2.0.13 and 2.0.11 would be computed at the end of their respective calculations. Thus, an actual calculation would proceed by executing the sweep over the x-coordinate first in the following fashion. Execute the following steps in this order: 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2 and 1. Then execute the sweep over the y-coordinate by executing the following steps in this order: 21, 20, 19, 18, 17, 16, 15, 14 and 13. Then execute another sweep over the y-coordinate by executing the following steps in this order: 10, 21, 20, 19, 18, 17, 16, 15, 14 and 13. Note that in this sweep over the y-coordinate we included step 10 to compute the value of Δt. Then executa a sweep over the x-coordinate by executing the following steps in this order: 11, 9, 8, 7, 6, 5, 4, 3, 2 and 1. Note that in this sweep over the x-coordinate, steps 12 and 10 are omitted. Step 12 is only done once at the very beginning of a simulation. Step 10 is done at the beginning of a pair of sweeps over either x and then y or over y and then x. Step 10 computes the value of the timestep, Δt, for a pair of sweeps over the two spatial coordinates.

There are also several other items which should be noted and discussed. One is that in the steps of the algorithm above, there is freedom to algebraically combine steps as desired to eliminate variables. One might choose a fewer number of variables in order to reduce the memory footprint of the implementation. The variables and steps above were chosen to describe an existing reference implementation. However, if variables are eliminated and steps combined, the result should be algebraically equivalent to the steps described above.

It should also be noted that in the algorithm described above, the calculations associated with a particular cell in a sweep over either of the two coordinates is independent of all the other cell values being calculated at the new time step. More specifically, the values of the state variables described in equation 1.0.9 depend only on the old values of those state variables at the previous time step or previous sweep. This means that this algorithm has a very large amount of fine grain parallelism and this may be exploitable by some hardware architectures. Or, the fine grain parallelism can be organized into larger tasks which may be more appropriate for other hardware architectures.

It should also be noted that when performing a sweep over one of the two coordinates, there is no spatial coupling in the coordinate that is not being swept over. The only spatial coupling is in the coordinate being swept over and only the two nearest neighbors on either side of the cell being updated participate in this coupling. Thus there is significant data locality that can be exploited.

5 Verification of Implementation

In this section we discuss how one verifies that an implementation of this specification is accurate and conformant. There are multiple approaches that one could take to verify an implementation of the algorithm described in this specification. Perhaps the simplest is for Los Alamos National Laboratory to make available for download the output of one of our implementations for several different test problems at several different problem sizes. This is the approach we have currently decided upon. We will provide an input file for a test problem and the output file that goes with it. Both files will be simple text files and the output file will contain the values of the state variable, $\boldsymbol{U}$, printed out to machine precision at the end of the simulation. There will be several input/output file pairs available to represent several test problems run at different problem sizes. These test results will be maintained in a source control repository. Access to this repository may be requested by anyone who chooses to implement this specification and desires to verify it. It should be noted that there is an implicit responsibility assumed by Los Alamos National Laboratory to make sure that we possess and maintain a verified implementation of this specification.

6 User Defined Input Parameters

During the detailed description of the calculation flow presented in Section 3, several parameters were identified as user specified input parameters for the algorithm. These will now be defined and discussed.

1. n_x is the number of cells in the x-coordinate. Together with Δx, it defines the size of the simulation domain in the x-coordinate such that $L_x = n_x \Delta x$. This parameter is referenced in each of the steps in Section 3.

2. n_y is the number of cells in the y-coordinate. Together with Δy, it defines the size of the simulation domain in the y-coordinate such that $L_y = n_y \Delta y$. This parameter is referenced in each of the steps in Section 3.

3. Δx is the size of a cell in the x-coordinate. Together with n_x, it defines the size of the simulation domain in the x-coordinate such that $L_x = n_x \Delta x$. Note that Δx is the same for all cells and this leads to a mesh spacing that is uniform in the x-coordinate. This parameter is referenced in Steps 1, 7, 10 and 12 of Section 3.

4. Δy is the size of a cell in the y-coordinate. Together with n_y, it defines the size of the simulation domain in the y-coordinate such that $L_y = n_y \Delta y$. Note that Δy is the same for all cells and this leads to a mesh spacing that is uniform in the y-coordinate. This parameter is referenced in Steps 10, 12, 13 and 19 of Section 3.

5. γ is the ratio of specific heats and is 5/3 for an ideal gas. This parameter is referenced in Steps 2, 4, 5, 8, 10, 12, 14, 16, 17 and 20 of Section 3.

6. C_f is the Courant time step limit factor which is used in the calculation of the next time step value. For numerical stability, this value should be less than or equal to one. This parameter is referenced in Step 10 of Section 3.

7. c_s is a smallness parameter used in some equations to provide good numerical properties including avoidance of divide by zero errors. A good value for it is 10^{-10}. This parameter is referenced in Steps 4, 5, 8, 10, 16, 17 and 20 of Section 3.

8. r_s is a smallness parameter used in some equations to provide good numerical properties including avoidance of divide by zero errors. A good value for it is 10^{-10}. This parameter is referenced in Steps 4, 5, 8, 10, 16, 17 and 20 of Section 3.

9. `nr_itlim` is the maximum number of iterations to take in the Newton- Raphson calculation used in the Riemann solver. A reasonable value for this is 10. The larger this value, the closer the Riemann solver is to an exact Riemann solver. This parameter is referenced in Steps 4 and 16 of Section 3.

10. `nr_tol` is the convergence tolerance on the Newton-Raphson solver. A reasonable value for this parameter is 10^{-6}. The smaller this value, the more accurate the Riemann solver is and the closer it is to an exact Riemann solver. This parameter is referenced in Steps 4 and 16 of Section 3.

11. T_s is the slope type input parameter. The value of this input variable should be 1.0. It is used in Steps 7 and 19 of Section 3.

12. `order` selects the order of accuracy for the calculation and should be an integer equal to 1 or 2. A good value is 2 as this provides for a more accurate algorithm at the expense of more computation. It is used in Steps 7 and 19 of Section 3.

13. `scheme` is a hydro scheme input parameter. It can take values of `muscl`, `plmde` and `collela`. When `scheme` = `muscl`, the MUSCL-Hancock [5] version of the Godunov algorithm is used. When `scheme` = `plmde`, a Piecewise Linear MUSCL Direct Eulerian version of the Godunov algorithm is used. When

`scheme` = `collela`, a Piecewise Parabolic Method [1] version of the Godunov algorithm is used which is also known as Colella's method. It is used in Steps 7 and 19 of Section 3.

14. `bc_left` specifies the boundary condition for the left boundary in the x-coordinate of the problem domain. This parameter is referenced in Step 11 of Section 3.

15. `bc_right` specifies the boundary condition for the right boundary in the x-coordinate of the problem domain. This parameter is referenced in Step 11 of Section 3.

16. `bc_top` specifies the boundary condition for the top boundary in the y-coordinate of the problem domain. This parameter is referenced in Step 21 of Section 3.

17. `bc_bottom` specifies the boundary condition for the bottom boundary in the y-coordinate of the problem domain. This parameter is referenced in Step 21 of Section 3.

18. `problem` specifies the test problem to be used. This parameter is referenced in Step 12 of Section 3.

References

[1] Phillip Colella and Paul R. Woodward. The piecewise parabolic method (ppm) for gas-dynamical simulations. *Journal of Computational Physics*, 54(1):174 – 201, 1984.

[2] S. K. Godunov, A. V. Zabrodin, and G. P. Prokopov. A computational scheme for two-dimensional non stationary problems of gas dynamics and calculation of the flow from a shock wave approaching a stationary state. *USSR Computational Mathematics and Mathematical Physics*, 1(4):1187 – 1219, 1962.

[3] Randall J. Leveque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 40 West 20th Street, New York, NY 10011-4211, USA, 2002.

[4] Gilbert Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968.

[5] Bram van Leer. Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method. *Journal of Computational Physics*, 32(1):101 – 136, 1979.