Project 3(b): 2D Ultrarelativistic Fluid.

July 30, 2003

1 Introduction

In this project we generalize the code produced in Project 2 to two dimensions (plus time). We consider the equations of a fluid with an ultrarelativistic equation of state in Minkowski spacetime. In general all fluid quantities only depend on 2 spatial coordinates, x and y, and time. We again choose a coordinate system in which v^z vanishes. As in Project 2, good general references for this one are Martí, *et al.* [1] and Font [2].

2 Equation of Motion

As we have seen in the previous project the equations of motion can be calculated from:

$$\left(T^{ab}\right)_{;a} = 0 \tag{1}$$

where remember that the stress energy tensor is given by the following expression:

$$T^{ab} = (\rho + P)\, u^a u^b + P g^{ab}, \tag{2}$$

and we have ultrarelativistic equation of state $P = (\Gamma - 1)\,\rho$. We consider the Minkowski spacetime with metric, $g^{\mu\nu} = \eta^{\mu\nu} = \mathrm{diag}\{-1, 1, 1, 1\}$, therefore the equations of motion can be written as:

$$\left(T^{\mu\nu}\right)_{,\mu} = 0. \tag{3}$$

Now we introduce conservative variables analogous to the ones defined in Project 2, see Hawke [3]:

$$\begin{aligned} \tau &= (\rho + P)\, W^2 - P, && (4)\\ S_x &= (\rho + P)\, W^2 v^x, && (5)\\ S_y &= (\rho + P)\, W^2 v^y, && (6) \end{aligned}$$

where we have used $W^2 = \left(1 - v^2\right)^{-1}$ with $v^2 = v^{x2} + v^{y2}$. Note that $v^i = u^i/u^t$ where i take values $\{x, y\}$ and that $W = u^t$. We left as an exercise to check that we can write equations (3) in conservation law form:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}^x(\mathbf{q})}{\partial x} + \frac{\partial \mathbf{f}^y(\mathbf{q})}{\partial y} = \mathbf{0}, \tag{7}$$

using the following vectors:

$$\mathbf{q} = \begin{bmatrix} \tau \\ S_x \\ S_y \end{bmatrix}, \tag{8}$$

$$\mathbf{f}^x = \begin{bmatrix} (\tau + P)\, v^x \\ S_x v^x + P \\ S_y v^x \end{bmatrix}, \qquad \mathbf{f}^y = \begin{bmatrix} (\tau + P)\, v^y \\ S_x v^y \\ S_y v^y + P \end{bmatrix}. \tag{9}$$

These are the equations that are suited for discretization. Note that in this case we have 2 physical flux vectors.

3 2D Discretization

The discretization in two dimensions is a generalization of the discretization in one dimension. Starting from the continuous equations in differential form, (7), and taking the average over

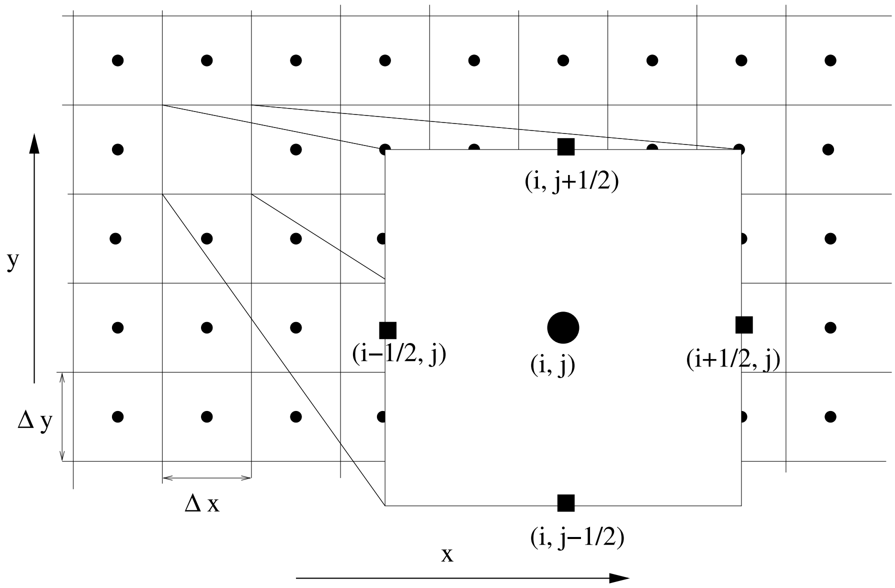

Figure 1: In this Figure we can see the spatial cell structure. The dots represent the points where the spatial averages of the conservation variables, $\bar{\mathbf{q}}^n_{i\ j}$, lie. Note that this quantities are placed on full time steps, i.e. t^n, t^{n+1}, etc, ... The squares represent the positions of the fluxes $\mathbf{F^x}^{n+1/2}_{i+1/2\ j}$ and $\mathbf{F^y}^{n+1/2}_{i\ j+1/2}$ which are positioned on the half step, i.e. $t^{n+1/2}$

a space time cell $C^{n+1/2}_{i\ j} \equiv (t^n, t^{n+1}) \times (x_{i-1/2}, x_{i_{i+1/2}}) \times (y_{j-1/2}, y_{j+1/2})$, we get the following expression:

$$\frac{\bar{\mathbf{q}}^{n+1}_{i\ j} - \bar{\mathbf{q}}^{n}_{i\ j}}{\Delta t} + \frac{\mathbf{F^x}^{n+1/2}_{i+1/2\ j} - \mathbf{F^x}^{n+1/2}_{i-1/2\ j}}{\Delta x} + \frac{\mathbf{F^y}^{n+1/2}_{i\ j+1/2} - \mathbf{F^y}^{n+1/2}_{i\ j-1/2}}{\Delta y} = \mathbf{0} \tag{10}$$

where $\Delta t = t^{n+1} - t^n$, $\Delta x = x_{i+1/2} - x_{i-1/2}$ and $\Delta y = y_{j+1/2} - y_{j-1/2}$ and we have defined the following average quantities:

$$\bar{\mathbf{q}}^n_{i\ j} \equiv \frac{1}{\Delta x \Delta y} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{q}^n(t^n, x, y) dx dy, \tag{11}$$

$$\mathbf{F^x}^{n+1/2}_{i+1/2\ j} \equiv \frac{1}{\Delta t \Delta y} \int_{t^n}^{t^{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{f}^x \left(t, x_{i+1/2}, y\right) dt dy, \tag{12}$$

$$\mathbf{F^y}^{n+1/2}_{i\ j+1/2} \equiv \frac{1}{\Delta t \Delta x} \int_{t^n}^{t^{n+1}} \int_{x_{j-1/2}}^{x_{j+1/2}} \mathbf{f}^y \left(t, x, y_{j+1/2}\right) dt dx. \tag{13}$$

In Figure 2 you can see the position on the spacetime of the quantities previously defined.

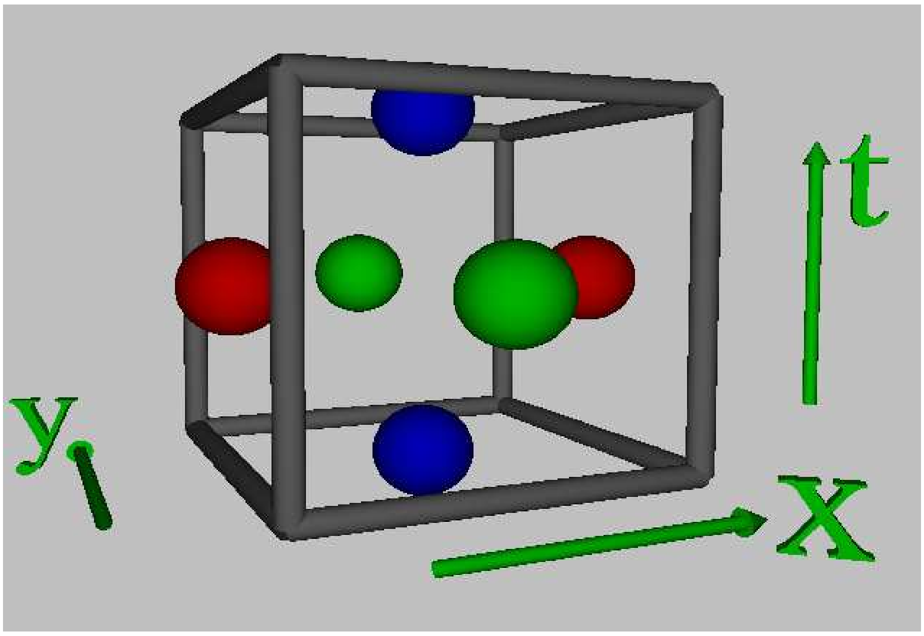

Figure 2: Structure of one cell $C^{n+1/2}_{i\ j}$. The bottom plane represents the spatial surface t^n while the top one is t^{n+1}. The blue spheres on these planes are the spatial averages $\bar{\mathbf{q}}^n_{i\ j}$ and $\bar{\mathbf{q}}^{n+1}_{i\ j}$. The red spheres, on the left and far right, are located at the positions of spacetime where the numerical fluxes in the x direction, $\mathbf{F^x}^{n+1/2}_{i+1/2\ j}$ and $\mathbf{F^x}^{n+1/2}_{i-1/2\ j}$, are located. Similarly the green spheres, on front and behind faces, are located at the positions where the numerical fluxes in the y direction, $\mathbf{F^y}^{n+1/2}_{i\ j+1/2}$ and $\mathbf{F^y}^{n+1/2}_{i\ j-1/2}$, are placed. Note that the each sphere corresponds to an average of a certain quantity over the face of the cube where it is placed.

4 Roe Solver

The procedure to calculate the Roe fluxes is identical as in the one dimensional case. We will consider two different linearizations depending on which flux we want to calculate. For the x numerical flux we solve the following problem at the cell interfaces located at $x_{i+1/2}$:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}^{\mathbf{x}}}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial x} = 0 \tag{14}$$

and for the y, at the $y_{j+1/2}$ interfaces:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}^{\mathbf{y}}}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial y} = 0 \tag{15}$$

The result for the Roe flux in the x direction is:

$$\mathbf{F^x}^{Roe}_{i+1/2\ j} = \frac{1}{2}\left[\mathbf{f}^x\left(\tilde{\mathbf{p}}^R_{i+1/2\ j}\right) + \mathbf{f}^x\left(\tilde{\mathbf{p}}^L_{i+1/2\ j}\right) - \sum_\alpha |\lambda^x_\alpha| \omega^x_\alpha \boldsymbol{r}^x_\alpha\right], \tag{16}$$

and the one in the y direction:

$$\mathbf{F^y}^{Roe}_{i\ j+1/2} = \frac{1}{2}\left[\mathbf{f}^y\left(\tilde{\mathbf{p}}^R_{i\ j+1/2}\right) + \mathbf{f}^y\left(\tilde{\mathbf{p}}^L_{i\ j+1/2}\right) - \sum_\alpha |\lambda^y_\alpha| \omega^y_\alpha \boldsymbol{r}^y_\alpha\right]. \tag{17}$$

Now we have four groups of reconstructed variables. Reconstructed variables along the x direction, $(\tilde{\mathbf{p}}^R_{i+1/2\ j}, \tilde{\mathbf{p}}^L_{i+1/2\ j})$ and the reconstructed variables along the y direction $(\tilde{\mathbf{p}}^R_{i\ j+1/2}, \tilde{\mathbf{p}}^L_{i\ j+1/2})$. In the x direction we calculate the reconstruction of the primitive variables in a similar way we did in the one dimensional case:

$$\tilde{\mathbf{p}}^L_{i+1/2\ j} = \bar{\mathbf{p}}_{i\ j} + \boldsymbol{\sigma}_{i\ j}\left(x_{i+1/2} - x_i\right), \tag{18}$$

$$\tilde{\mathbf{p}}^R_{i+1/2\ j} = \bar{\mathbf{p}}_{i+1\ j} + \boldsymbol{\sigma}_{i+1\ j}\left(x_{i+1/2} - x_{i+1}\right), \tag{19}$$

where $\boldsymbol{\sigma}_{i\ j}$ is:

$$\boldsymbol{\sigma}_{i\ j} = \text{minmod}\left(\mathbf{s}_{i-1/2\ j}, \mathbf{s}_{i+1/2\ j}\right). \tag{20}$$

Here:

$$\mathbf{s}_{i+1/2\ j} = \frac{\bar{\mathbf{p}}_{i+1\ j} - \bar{\mathbf{p}}_{i\ j}}{x_{i+1} - x_i}, \tag{21}$$

with minmod defined exactly the same way as in Project 2. Analogously you can calculate the reconstructed variables along the y direction interchanging the x by the y indexes and coordinates.

In formulas (16) and (17) we also have the characteristic structure of the two Jacobian matrices:

$$\mathbf{A^x} = \left.\frac{\partial \mathbf{f}^x}{\partial \mathbf{q}}\right|_{1/2\ (\tilde{\mathbf{q}}^R_{i+1/2\ j} + \tilde{\mathbf{q}}^L_{i+1/2\ j})}, \quad \mathbf{A^y} = \left.\frac{\partial \mathbf{f}^y}{\partial \mathbf{q}}\right|_{1/2\ (\tilde{\mathbf{q}}^R_{i\ j+1/2} + \tilde{\mathbf{q}}^L_{i\ j+1/2})}. \tag{22}$$

As before $(\lambda^a_\alpha, \mathbf{r}^a_\alpha, \omega^a_\alpha)$ are the eigenvalues, right eigenvectors and the jumps in the characteristic variables of the matrix $\mathbf{A}^i$ with i taking values $\{x, y\}$, note that now α takes values $(1, 2, 3)$[1] because we have 3 equations and therefore 3×3 matrices. The jumps in this case are still defined by:

$$\tilde{\mathbf{q}}^R_{i+1/2\ j} - \tilde{\mathbf{q}}^L_{i+1/2\ j} = \sum_\alpha \omega^x_\alpha \boldsymbol{r}^x{}_\alpha. \tag{23}$$

[1]As in Project 2 here we are not using the convention in Project 1, although α is a Greek index it labels the equation number, it is not a spacetime index.

And analogously for the jumps along y. In the one dimensional case we could solve this equation easily to get the values of ω_α is terms of the differences $\tilde{\mathbf{q}}^R - \tilde{\mathbf{q}}^L$ and the right eigenvectors $\mathbf{r}_\alpha$. In this case it requires a little bit more algebra. It is useful to introduce the left eigenvectors defined by:

$$\mathbf{l}^x_\alpha \mathbf{A}^x = \lambda_\alpha \mathbf{l}^x_\alpha. \tag{24}$$

Using the appropriate normalization the matrix produced by setting the left eigenvectors on matrix form row by row is the inverse of the matrix produced setting the right eigenvectors column by column. This implies that:

$$\mathbf{l}^x_\alpha \cdot \left(\tilde{\mathbf{q}}^R_{i+1/2\ j} - \tilde{\mathbf{q}}^L_{i+1/2\ j}\right)^{\mathrm{T}} = \omega^x_\alpha. \tag{25}$$

An alternative to this procedure is to numerically solve the system of equations (23) for ω^x_α. As before the Roe fluxes (16) and (17) when evaluated using the fluid quantities defined on t^n are only first order approximations to the real numerical fluxes defined by (12) and (13) because of the non-linearity of the equations. In order to make the time step second order in the absence of shocks, we set the whole procedure in a second order Runge-Kutta integration step as in the one dimensional case.

Characteristic Structure

In the following expressions $c_s = \sqrt{\Gamma - 1}$ stands for the speed of sound in the fluid $\sqrt{\partial P/\partial \rho}$. For $\mathbf{A}^x$, the eigenvalues are:

$$\lambda^x_o = v^x, \tag{26}$$

$$\lambda^x_\pm = \frac{1}{1 - v^2 c_s^2}\left\{v^x\left(1 - c_s^2\right) \pm c_s\sqrt{\left(v^2 - 1\right)\left(c_s^2 v^2 - 1 + v^{x2}\left(1 - c_s^2\right)\right)}\right\}. \tag{27}$$

Right eigenvectors:

$$\mathbf{r}^x_o = \begin{bmatrix} 2v^y / \left(1 + v^2 - 2v^{x2}\right) \\ 2v^y v^x / \left(1 + v^2 - 2v^{x2}\right) \\ 1 \end{bmatrix}, \tag{28}$$

$$\mathbf{r}^x_\pm = \begin{bmatrix} \left(1 - v^{x2}\right)\left(v^x - \lambda^x_\pm\right)\left(1 - c_s^2 v^2\right) \\ \left(1 - v^{x2}\right)\left[\left(\lambda^x_\pm v^x - 1\right) c_s^2\left(1 - v^2\right) + v^x\left(1 - c_s^2\right)\left(v^x - \lambda^x_\pm\right)\right] \\ v^y\left[c_s^2\left(\lambda^x_\pm(1 + v^{x2})v^2 - v^{x2}\lambda^x_\pm - 2v^x v^2\right) + \left(1 - c_s^2\right) v^{x2}\left(\lambda^x_\pm - v^x\right) + \left(c_s^2 + 1\right) v^x - \lambda^x_\pm\right] \end{bmatrix}. \tag{29}$$

Left eigenvectors:

$$\mathbf{l}^x_o = \left[-\frac{v^y\left(1 + v^2 - 2v^{x2}\right)}{\left(1 - v^{x2}\right)\left(1 - v^2\right)}, \frac{v^y v^x\left(1 + v^2 - 2v^{x2}\right)}{\left(1 - v^{x2}\right)\left(1 - v^2\right)}, \frac{1 - v^2 + 2v^{y2}}{1 - v^2}\right], \tag{30}$$

$$\mathbf{l}^x_\pm = \frac{\pm}{\Delta_x}\begin{bmatrix} -c_s^2\lambda^x_\mp v^{x3}v^2 + 3c_s^2 v^x\left(v^{y2}\right)\left(v^x - \lambda^x_\mp\right) + \\ \left(c_s^2 + 1\right)\left(1 - v^{x2}\right) + \left(c_s^2 + 1\right)\left(v^{y2}\right) + \left(c_s^2 + 1\right)\lambda^x_\mp v^x\left(3v^{x2} - 2\right) + \\ \lambda^x_\mp v^x\left(3 - 4v^{x2}\right) + v^{x4} - \left(v^{y2}\right) - 1 \\ \left[1 + \left(2c_s^2 - 1 + 2v^{x2}c_s^2\right)v^{x2}\right]v^x + \left(-3c_s^2 - v^{x2}c_s^2\right)v^x v^2 + \\ \left[-1 + \left(1 - 4v^{x2}c_s^2\right)v^{x2} + \left(3v^{x2}c_s^2 + c_s^2\right)v^2\right]\lambda^x_\mp \\ -2c_s^2 v^y\left(1 - v^{x2}\right)\left(1 - 2\lambda^x_\mp v^x + v^{x2}\right) \end{bmatrix}, \tag{31}$$

$$\Delta_x = \left(1 - v^{x2}\right)^3 c_s^2 \left(\lambda_-^x - \lambda_+^x\right) \left(1 - v^2\right) \left(1 - c_s^2 v^2\right). \tag{32}$$

For $\mathbf{A}^y$: The eigenvalues are:

$$\lambda_o^y = v^y, \tag{33}$$

$$\lambda_\pm^y = \frac{1}{1 - v^2 c_s^2} \left\{ v^y \left(1 - c_s^2\right) \pm c_s \sqrt{\left(v^2 - 1\right)\left(c_s^2 v^2 - 1 + v^{y2}\left(1 - c_s^2\right)\right)} \right\}. \tag{34}$$

The right eigenvectors:

$$\mathbf{r}_o^y = \begin{bmatrix} 2v^x / \left(1 + v^2 - 2v^{y2}\right) \\ 1 \\ 2v^y v^x / \left(1 + v^2 - 2v^{y2}\right) \end{bmatrix}, \tag{35}$$

$$\mathbf{r}_\pm^y = \begin{bmatrix} \left(1 - v^{y2}\right)\left(v^y - \lambda_\pm^y\right)\left(1 - c_s^2 v^2\right) \\ v^x \left[c_s^2 \left(\lambda_\pm^y (1 + v^{y2}) v^2 - v^{y2} \lambda_\pm^y - 2 v^y v^2\right) + \left(1 - c_s^2\right) v^{y2} \left(\lambda_\pm^y - v^y\right) + \left(c_s^2 + 1\right) v^y - \lambda_\pm^y\right] \\ \left(1 - v^{y2}\right)\left[\left(\lambda_\pm^y v^y - 1\right) c_s^2 \left(1 - v^2\right) + v^y \left(1 - c_s^2\right)\left(v^y - \lambda_\pm^y\right)\right] \end{bmatrix}. \tag{36}$$

Left Eigenvectors:

$$\mathbf{l}_o^y = \left[-\frac{v^x \left(1 + v^2 - 2v^{y2}\right)}{\left(1 - v^{y2}\right)\left(1 - v^2\right)}, \frac{1 - v^2 - 2v^{x2}}{1 - v^2}, \frac{v^x v^y \left(1 + v^2 - 2v^{y2}\right)}{\left(1 - v^{y2}\right)\left(1 - v^2\right)} \right], \tag{37}$$

$$\mathbf{l}_\pm^y = \frac{\pm}{\Delta_y} \begin{bmatrix} -c_s^2 \lambda_\mp^y v^{y3} v^2 + 3 c_s^2 v^y \left(v^{x2}\right)\left(v^y - \lambda_\mp^y\right) + \\ \left(c_s^2 + 1\right)\left(1 - v^{y2}\right) + \left(c_s^2 + 1\right)\left(v^{x2}\right) + \left(c_s^2 + 1\right) \lambda_\mp^y v^y \left(3v^{y2} - 2\right) + \\ \lambda_\mp^y v^y \left(3 - 4v^{y2}\right) + v^{y4} - \left(v^{x2}\right) - 1 \\ -2 c_s^2 v^x \left(1 - v^{y2}\right)\left(1 - 2\lambda_\mp^y v^y + v^{y2}\right) \\ \left[1 + \left(2c_s^2 - 1 + 2c_s^2 v^{y2}\right) v^{y2}\right] v^y + \left(-3c_s^2 - c_s^2 v^{y2}\right) v^y v^2 + \\ \left[-1 + \left(1 - 4c_s^2 v^{y2}\right) v^{y2} + \left(3 c_s^2 v^{y2} + c_s^2\right) v^2\right] \lambda_\mp^y \end{bmatrix}, \tag{38}$$

$$\Delta_y = \left(1 - v^{y2}\right)^3 c_s^2 \left(\lambda_-^y - \lambda_+^y\right) \left(1 - v^2\right) \left(1 - c_s^2 v^2\right). \tag{39}$$

Boundary Conditions and Cell Structure

In this section we explain how the boundary conditions are set in two dimensions. The procedure is completely analogous to the one in Project 2 using ghost cells. RNPL produces a mesh of points of size N_x by N_y that we will consider to be centred at positions (x_i, y_j). In Figure 3 we show the spatial cell structure, including ghost cells. This grid of points include the centres of the ghost cells. In order to set approximate outgoing boundary conditions we copy the values of the conservative variables in the last physical cell into the ghost cells:

$$q_{1\ j} = q_{3\ j} \tag{40}$$

$$q_{2\ j} = q_{3\ j} \tag{41}$$

$$q_{N_x - 1\ j} = q_{N_x - 2\ j} \tag{42}$$

$$q_{N_x\ j} = q_{N_x - 2\ j} \tag{43}$$

where q is one of the conservative variables and j take values on $1, ..., N_y$. Interchanging the x and y indexes we get the equations for the boundary conditions in y.

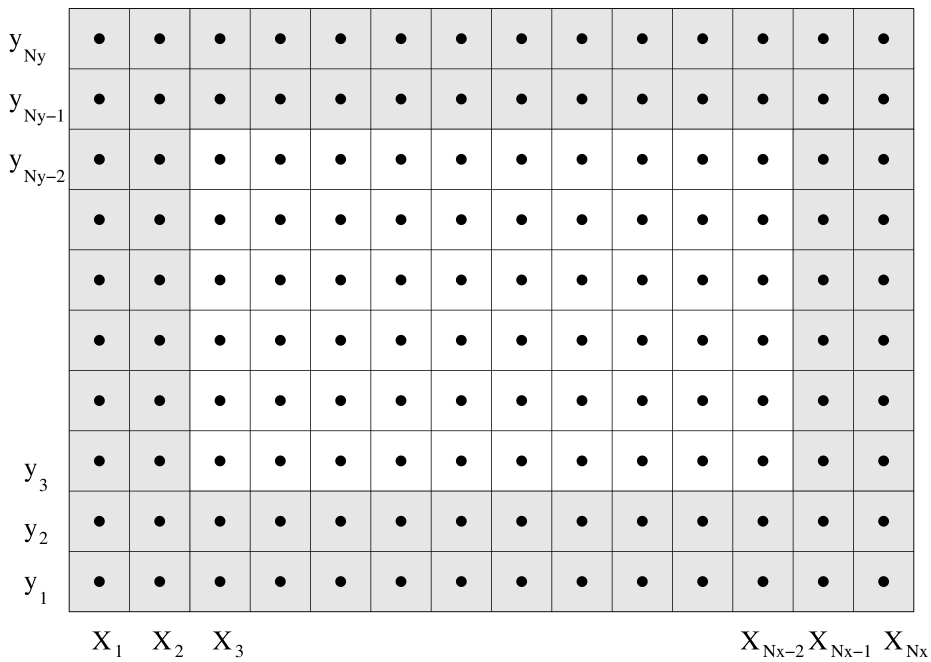

Figure 3: Cell structure on a spatial surface with 2 ghost cells per boundary and per direction ($N_{gx} = N_{gy} = 2$). The ghost cells are the shaded areas. Note that the points (x_i, y_j) are where the $\mathbf{q}^n_{i\ j}$ are calculated. In order to calculate the evolution of the interior points, points with coordinates (x_i, y_j) with $i \in \{N_{gx}+1, ..., N_x - N_{gx}\}$ and $j \in \{N_{gy}+1, ..., N_y - N_{gy}\}$, we need to calculate the x fluxes, $\mathbf{F^x}^{n+1/2}_{i+1/2\ j}$, at positions $(x_{i+1/2}, y_j)$ with $i \in \{N_{gx}, ..., N_x - N_{gx}\}$ and $j \in \{N_{gy}+1, N_y - N_{gy}\}$. Similarly we need to calculate the y fluxes, $\mathbf{F^y}^{n+1/2}_{i\ j+1/2}$, at positions $(x_i, y_{j+1/2})$ with $i \in \{N_{gx}+1, ..., N_x - N_{gx}\}$ and $j \in \{N_{gy}, ..., N_y - N_{gy}\}$. Note that $x_1 = x_{min}$, $x_{Nx} = x_{max}$, $y_1 = y_{min}$ and $y_{Ny} = y_{max}$.

PROBLEM 3 a) Using the 1-D ultrarelativistic code that you have produced in Project 2 as a template write a 2 dimensional code that solves equations (7). Note that now before updating the conservative variables you need to calculate both numerical fluxes in the x and y directions. After the conservative variables have been updated, both at the half and full step, you will need to floor τ, using the following equation:

$$\tau = \max\{\tau, \text{floor} + \sqrt{{S_x}^2 + {S_y}^2}\}. \tag{44}$$

You also need to find equations in order to calculate the primitive variables from the conservative ones inverting equations (4-6).

Ensure that your code can evolve both smooth and discontinuous initial data. Setting initial data that is constant along y and discontinuous in x solve for 1 dimensional Riemann problem and compare with the solution obtained from your code in Project 2. Interchange x and y and solve for the Riemann problem along the other direction. Make sure that your code also works for Riemann problems along $x+y$, setting initial data such that is constant along $x-y$.

The following problems are only a couple of suggestions of systems you can study. Choose the one, or ones, that interests you most and go for it, be adventurous:

Problem 3 b) There exists a very well known instability usually called Kelvin Helmholtz instability or instability of tangential discontinuities, see Landau and Lifshitz [4]. Although the *real* instability happens for fluids with the *ideal* equation of state (more generally fluids that can sustain a contact discontinuity) we can get an approximation of it using the ultrarelativistic fluid code. Set initial data such that the XY plane has two different regions divided by the following function:

$$x = \begin{cases} x_d & \text{if} \quad y < y_d \\ x_d + Asin\left(w\pi(y - y_d)\right) & \text{if} \quad y > y_d. \end{cases} \tag{45}$$

Try to choose values A, w, y_d, x_d such that the discontinuity is almost a straight line with a small curvature for $y > y_d$. Produce initial data for the primitive variables such that the density is constant across the discontinuity and the component of the velocity tangential to the surface is large (less than 1) in one of the region and zero on the other. The perpendicular component of the velocity should be zero on both regions. In the Lab web page you can find a couple of MPEG movies showing this evolution.

Problem 3 c) A very rudimentary way to produce a jet is to set the conservative variables in a region of the ghost cells to have constant values, independent of the values of the physical cells. Set initial conditions such that you have a constant density of $\rho = 6.0$ and zero velocity on all the integration range. Choose a small region region in the boundary at $x = x_{min}$ with values of y close to the centre of grid and set the conservative variables to have values: $\tau_b = 0.15, S_{xb} = 0.14, S_{yb} = 0$ for $\Gamma = 1.5$ and $floor = 10^{-10}$ at every time step. This should produce a jet of fluid coming into the range of integration. You can set another jet coming from the opposite boundary and make them collide one into the other. In the Lab web page you can find animations for this simulation.

References

[1] J.M. Marti, and E. Mueller, "Numerical Hydrodynamics in Special Relativity", Living Rev. Relativity 2, (1999), 3. [Online article]: cited on 25 Jun 2003 http://www.livingreviews.org/lrr-1999-3

[2] J. Font, "Numerical Hydrodynamics in General Relativity", Living Rev. Relativity 3, (2000), 2. [Online article]: cited on 25 Jun 2003 http://www.livingreviews.org/lrr-2000-2

[3] I. Hawke, University of Cambridge, Queens' College, Ph.D. thesis, http://www.aei-potsdam.mpg.de/~hawke/thesis/thesis.html

[4] L. D. Landau, and E. M. .Lifshifz, Fluid Mechanics, Pergamon Press, Addison-Wesley Publishing Company, inc. Reading, Massachusetts, 1959