Chapter 19

Finite Volume Method for Scalar Advection in 2D

19.1 Introduction

The purpose of this exercise is to code a program to integrate the scalar advection equation in two-dimensional flows.

19.2 The 2D equation

The 2D scalar advection equation in conservative form is:

$$T_t + (uT)_x + (vT_y) = 0. \tag{19.1}$$

where $\vec{u} = (u, v)$ is a two-dimensional flow field that obeys the mass conservation equation of an incompressible fluid $\nabla \cdot \vec{u} = u_x + v_y = 0$. The formulation of this partial differential equation into a finite volume formulation follows the steps outlined in chapter 17. Here we concern ourselves primarily with the FV 2D discretization of equation 19.1.

19.3 The 2D spatial discretization

The spatial discretization proceeds by deviding the domain into rectangular cells as depicted in figure 19.1. We are now concerned with assigning the different terms appearing in equation 17.7. Here we presume that diffusion is non-existent ($\alpha = 0$). We have the following remarks:

- The cell volume is actually an area in 2D and we denote by $\delta A = \Delta x \Delta y$. where $\Delta x \times \Delta y$ are the cell sizes in the $x \times y$ directions.

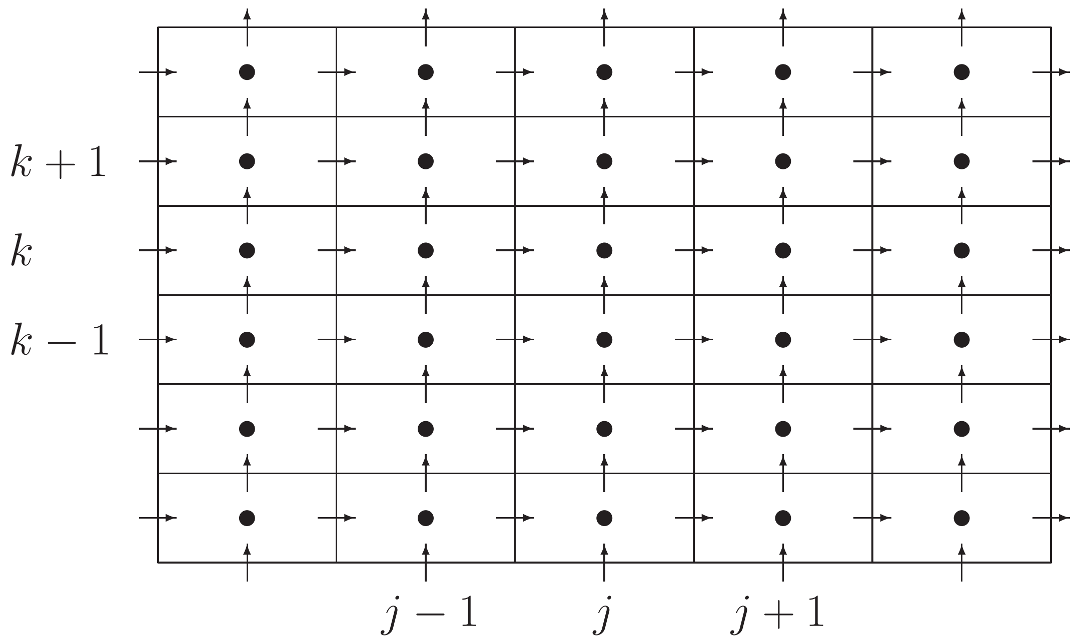

Figure 19.1: Cartesian finite volume grid with rectangular cells. The solid point represents the average of T over a cell while the arrows denote the $x-y$ advective fluxes into a cell through its 4 edges.

- The cells can be referenced by a pair of indices (j,k) along the (x,y) directions. The cell center has coordinate (x_j, y_k) and the cell walls are located at $(x_j \pm \frac{\Delta x}{2}, y_k \pm \frac{\Delta y}{2})$.
- Each cell has four edges with constant normal along each. The outward unit normal to cell (j,k) at $x = x_j + \Delta x/2$ points in the positive x-direction, and we have $\vec{u}\cdot\vec{n} = u$; whereas at $x = x_j - \Delta x/2$ it points in the negative x-direction and we have $\vec{u}\cdot\vec{n} = -u$. Similarly along $y = y_k + \Delta y/2$ we have $\vec{u}\cdot\vec{n} = v$, whereas along $y = y_k - \Delta y/2$ we have $\vec{u}\cdot\vec{n} = -v$.

With these remarks we can now write down the finite volume equation for cell (j,k):

$$\begin{aligned}
\delta A \frac{\mathrm{d}\overline{T}_{j,k}}{\mathrm{d}t} &+ \int_{y_k-\frac{\Delta y}{2}}^{y_k+\frac{\Delta y}{2}} \left[u\left(x_{j+\frac{1}{2}},y\right)T\left(x_{j+\frac{1}{2}},y\right) - u\left(x_{j-\frac{1}{2}},y\right)T\left(x_{j-\frac{1}{2}},y\right)\right]\,\mathrm{d}y \\
&+ \int_{x_j-\frac{\Delta j}{2}}^{x_j+\frac{\Delta x}{2}} \left[v\left(x,y_{k+\frac{1}{2}}\right)T\left(x,y_{k+\frac{1}{2}}\right) - v\left(x,y_{k-\frac{1}{2}}\right)T\left(x,y_{k-\frac{1}{2}}\right)\right]\,\mathrm{d}x \\
&= 0 \qquad (19.2)
\end{aligned}$$

where $x_{j\pm\frac{1}{2}} = x_j \pm \frac{\Delta x}{2}$, and $y_{k\pm\frac{1}{2}} = y_k \pm \frac{\Delta y}{2}$. In this form we see that the equation is nothing but an accounting of the flux entering/leaving cell j. For ease of notation

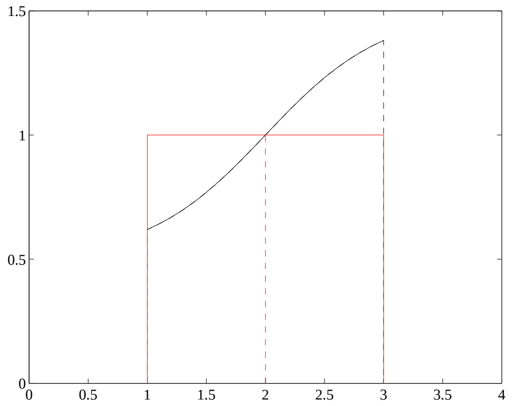

Figure 19.2: The midpoint rule approximates the area under the black curve with the area under the red rectangle. The rectangle height is determined by the function value at the mid-point of the interval, the dashed red curve.

we denote the x, y-fluxes by $(F, G) = (uT, vT)$, and the equation can be re-written as:

$$\begin{aligned}\delta A\frac{\mathrm{d}\overline{T}_{j,k}}{\mathrm{d}t} \quad &+ \quad \int_{y_k-\frac{\Delta y}{2}}^{y_k+\frac{\Delta y}{2}} \left[F\left(x_{j+\frac{1}{2}}, y\right) - F\left(x_{j-\frac{1}{2}}, y\right)\right]\ \mathrm{d}y \\ &+ \quad \int_{x_j-\frac{\Delta j}{2}}^{x_j+\frac{\Delta x}{2}} \left[G\left(x, y_{k+\frac{1}{2}}\right) - G\left(x, y_{k-\frac{1}{2}}\right)\right]\ \mathrm{d}x = 0\end{aligned}$$

An added wrinkle to the 2D finite volume formulation is the need to evaluate boundary integrals (which were not encountered in the 1D case). The silver lining is that all the integrals have the form:

$$\int_{z_m-\frac{\Delta z}{2}}^{z_m+\frac{\Delta z}{2}} H(z)\ \mathrm{d}z \tag{19.3}$$

where $H(z)$ is some function of z. Here we use a simple second order integration scheme, the mid-point rule, and write

$$\int_{z_m-\frac{\Delta z}{2}}^{z_m+\frac{\Delta z}{2}} H(z)\ \mathrm{d}z \approx H(z_m)\Delta z + O(\Delta z^2) \tag{19.4}$$

A geometric interpretation of the mid-point rule is shown in figure 19.2. The area under the function $H(z)$, shown in black is approximated by the area under the red triangle, and whose height is defined by the function value at the mid-point z_m. The integration reduces hence to an evaluation of the function at the edge center multiplied by the size of the edge, Δz.

Here we use a simple second order integration scheme, the mid-point rule, and write

$$\int_{y_k-\frac{\Delta y}{2}}^{y_k+\frac{\Delta y}{2}} F\left(x_{j+\frac{1}{2}}, y\right)\ \mathrm{d}y \quad \approx \quad F\left(x_{j+\frac{1}{2}}, y_k\right)\Delta y \tag{19.5}$$

$$\int_{x_j-\frac{\Delta x}{2}}^{x_j+\frac{\Delta x}{2}} G\left(x, y_{k+\frac{1}{2}}\right)\,\mathrm{d}x \quad\approx\quad G\left(x_j, y_{k+\frac{1}{2}}\right)\Delta x \tag{19.6}$$

The above integration formula are exact if F varies linearly across the cells. Higher order integration formula could be used but would involve substantially more work. The final form for the 2D FV advection equation becomes:

$$\frac{\mathrm{d}\overline{T}_{j,k}}{\mathrm{d}t} = -\frac{\left[F\left(x_{j+\frac{1}{2}}, y_k\right) - F\left(x_{j-\frac{1}{2}}, y_k\right)\right]\Delta y + \left[G\left(x_j, y_{k+\frac{1}{2}}\right) - G\left(x_j, y_{k-\frac{1}{2}}\right)\right]\Delta x}{\delta A} \tag{19.7}$$

19.4 Time discretization

Equation 19.7 is an ordinary differential equation governing the time evolution of the cell-averaged tracer. Its time integration can be performed by one of the time-stepping scheme discussed previously; an appropriate scheme is the third order Runge-Kutta method (RK3). The right hand of this ODE requires simply the computations of the flux divergence onto cell (j, k). The final piece missing is the reconstruction of the function values at the cell edges prior to computing the fluxes, a topic we follow up on in the next section.

19.5 Function reconstruction

The flux integration requires simply the calculation of:

$$F\left(x_j + \frac{\Delta x}{2}, y_k\right) = F_{j+\frac{1}{2},k} = u\left(x_j + \frac{\Delta x}{2}, y_k\right)\ T\left(x_j + \frac{\Delta x}{2}, y_k\right) \tag{19.8}$$

$$G\left(x_j, y_k + \frac{\Delta y}{2}\right) = G_{j,k+\frac{1}{2}} = v\left(x_j, y_k + \frac{\Delta y}{2}\right)\ T\left(x_j, y_k + \frac{\Delta y}{2}\right) \tag{19.9}$$

The scheme requires the evaluation of the T function from its cell averages in neighboring cells. Keeping within the context of second order schemes, we propose to use a piecewise linear reconstruction so that, with an obvious notation,

$$T_{j+\frac{1}{2},k} = \frac{\overline{T}_{j,k} + \overline{T}_{j+1,k}}{2} \tag{19.10}$$

$$T_{j,k+\frac{1}{2}} = \frac{\overline{T}_{j,k} + \overline{T}_{j,k+1}}{2} \tag{19.11}$$

19.6 Algorithm Summary

We are now in a position to summarize the solution process.

1. Read or Set the physical and numerical parameters of the problem.

2. Define the domain's geometry including the number of cells in the x, y directions, the grid sizes, and the cell areas. It would helpfull also to store the coordinates of the cell centers as well as that of the cell edges.

3. Define the flow field. The code for the Stommel gyre will be provided.

4. Define the output units, and the output format of the files, and compute some preliminary diagnostics, like the budget of T over the domain.

5. Define the initial distribution of the tracer $T^0_{j,k}$. The cell averages at the initial time can be also deduced. Here and keeping with the second order accuracy philosophy, we will take the cell averages to be the value of T at the center of the cell. It is easy to convince yourself using a two-dimensional version of the mid-point rule that $\overline{T}_{j,k} = T(x_j, y_k) + O(\Delta x^2, \Delta y^2)$.

6. Start a time loop that call a subroutine to perform a single time integration using the RK3 routine.

7. The RK3 routine calls a function to compute the right hand side of the ODE, equation 19.7. It needs to call three times for the 3 stages of the scheme. The right hand side should accept the cell averages and return the flux divergence.

8. The right hand side computation requires the evaluation of the fluxes at cell edges prior to performing the fluxes. These are obtained from reconstructing the function values at the cell edges and multiplying by the velocity field.

9. Output some diagnostics, like the T budget within the domain, and the extrema of the field. The solution needs to be saved intermitently for examination.

19.7 Code Design

You should start with the one-dimensional finite volume code that was presented in class as it already includes many of the elements listed above. What is required is to transform the code from 1D to 2D. One sequence of modification is as follows:

19.7.1 Data Structure

It is obvious that the most efficient way to store the various data is in two dimensional matrices. The dimensions of the matrices differ, the cell averages must be `Tb(M,N)` where M, N are the number of cells in the x, y directions respectively.

19.7.2 Domain Geometry

The domain description needs to be upgraded to account for the multi-dimensionality of the problem. The grid information is contained in module `grid.f90`. The following information requires to be added to it:

- The grid size in the y-direction.
- The y-coordinate of the cell edges: $y_{k+\frac{1}{2}} = (k-1)\Delta y$, with $k = 1, 2, \ldots, N+1$, where N is the number of cells in the y-directions. The y-coordinates of the cell centers may also be needed and should be stored.

19.7.3 Flow

The flow field information should be updated. The u velocity should be made two-dimensional and the y component of the velocity should also be added. Notice that u need to be defined at vertical cell edges, and consequently its dimension should be declared as `u(M+1,N)` whereas v should be declared `v(M,N+1)`. Refer to the figure for further information about the grid.

19.7.4 T initiations

The setting of the initial condition was done in a subroutine included in module `params.f90` that contains all the problem statements. This subroutine should be made two-dimensional, and be passed the proper data in its argument list or via the modules it includes.

19.8 Tracer Advection in a Stommel Gyre

Here, we revisit the Hecht advection test problem ([5, 6, 7]) to characterize the behavior of the high-order scheme in the under-resolved regime. The Hecht advection test consists of advecting a passive tracer in a Stommel Gyre, and in particular through an intense westen boundary current where the smooth looking Gaussian Hill undergoes intense deformation. Under-resolved features in this regime produce substantial noise in the solution and fail to propagate the solution downstream and out of the boundary current region where the Hill reconstitute itself.

19.8.1 The flow field

The flow field is given by the so-called Stommel gyre model which is an idealized version of the Gulf Stream system. The flow is characterized by an intense Western Boundary current that moves waters northward in a narrow zone whose width

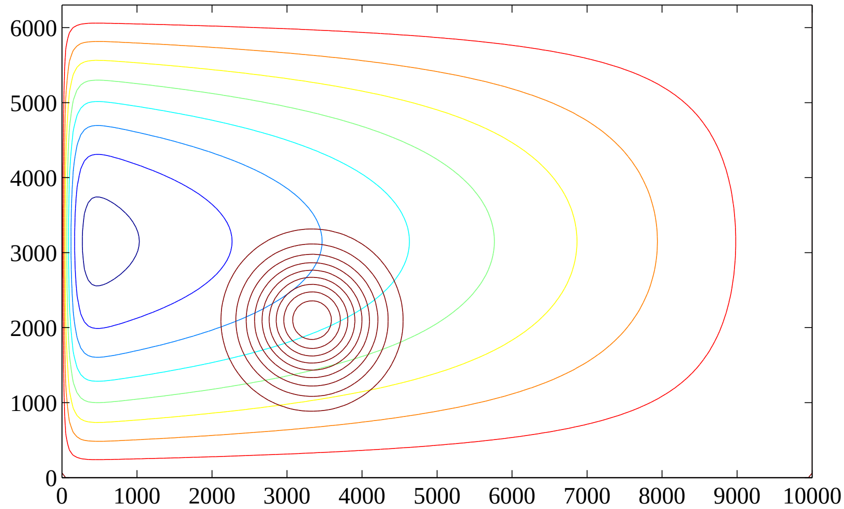

Figure 19.3: Streamlines of Stommel gyre flow field. The flow is characterized by an intense Western Boundary Current and a slow moving southerward current elsewhere in the basin. The bull's eye in the lower left corner is the initial distribution of T.

depends on the β-effect and the bottom friction value. The flow in the rest of the domain is slow and has little shear. The streamline for this flow is given by:

$$\psi = \Psi \sin \frac{y\pi}{b} \left[P e^{Ax} + (1-P) e^{Bx} - 1 \right] \tag{19.12}$$

$$P = \frac{1 - e^{Ba}}{e^{Aa} - e^{Ba}} \tag{19.13}$$

where a and b are the zonal and meriodinal size of the basin, respectively (the western and southern boundaries are located at $x = 0$ and $y = 0$, respectively). The parameters A and B determine the width of the western boundary current region and these, in turn, depend on the ratio of the β parameter, β, and drag coefficient, C_d as follows:

$$\alpha = \frac{\beta}{C_d} \tag{19.14}$$

$$A = -\frac{\alpha}{2} + \sqrt{\left(\frac{\alpha}{2}\right)^2 + \left(\frac{\pi}{b}\right)^2} \tag{19.15}$$

$$B = -\frac{\alpha}{2} - \sqrt{\left(\frac{\alpha}{2}\right)^2 + \left(\frac{\pi}{b}\right)^2} \tag{19.16}$$

The strength of the transport depends on the wind-stress, τ, the water density, ρ, the drag coefficient and the meridional size of the basin:

$$\Psi = \frac{\tau\pi}{\rho C_d b} \left(\frac{b}{\pi}\right)^2 \tag{19.17}$$

The advective velocity components can be obtained by differentiating the stream-function and dividing by the depth of the basin D

$$u = \frac{\psi_y}{D}, \quad v = -\frac{\psi_x}{D} \tag{19.18}$$

19.8.2 The initial condition

The initial condition for our tracer which we can think of as pollutant dumped in the ocean is a circular Gaussian hill distribution with a decay length scale of l

$$T = \exp\left(-\frac{(x-x_c)^2 + (y-y_c)^2}{l^2}\right) \tag{19.19}$$

and the center of the profile is located at $(x_c, y_c) = \left(\frac{a}{3}, \frac{b}{3}\right)$.

19.8.3 Expected result

The Gaussian hill will slowly makes its way to the western boundary region where it will be sheared and deformed substantially. The fidelity of the simulation will depend crucially on the spatial resolution used. The western boundary current with the current parameter settings will have a width of about 70 km. If the grid spacing is not enough to resolve the shear region, numerical noise will be generated and will manifest itself in either large positive and negative values outside the bounds of the initial condition ($0 \leq T \leq 1$). One of the aim of this exercise is to decide at what grid resolution, compared the width of the western boundary current, can one consider the simulation to be adequate. The report should thus include multiple runs with $\Delta x = 100, 50, 20, 10$ km.

The integration should be carried out for 3 (1080 days or years taking snapshots every month to record the time-evolution of T. The maximum speed is about 1.5 m/s; the time step should be scaled accordingly so that the Courant number, $C = u\Delta t/Dx$ does not exceed 1/2.

The solution after 3 years of integration is shown in figure 19.4.

19.8.4 Support Code

A number of matlab scripts will be provided to help in visualizing the results in matlab. The scripts are located in directory `~mohamed/Project` on metofis. Feel free to modify them to fit your needs.

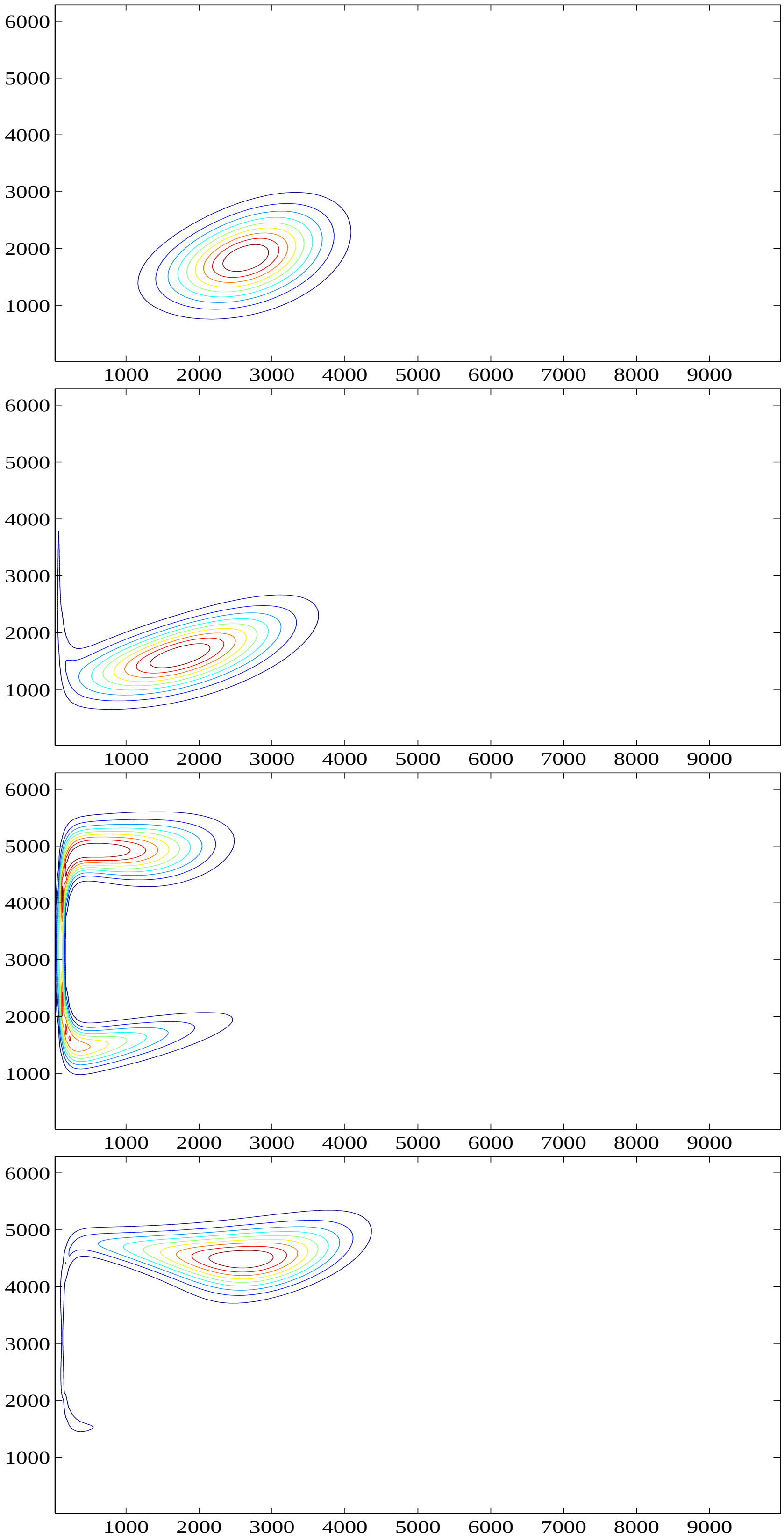

Figure 19.4: Solution by a method of characteristics that tracks the solution after 1/2 (first), 1 (second), 2 (third) and 3 (fourth) years of integration.