Introduction to fluid simulation in plasmas

Bhavesh Patel

Institute for Plasma Research, India

March 9, 2017

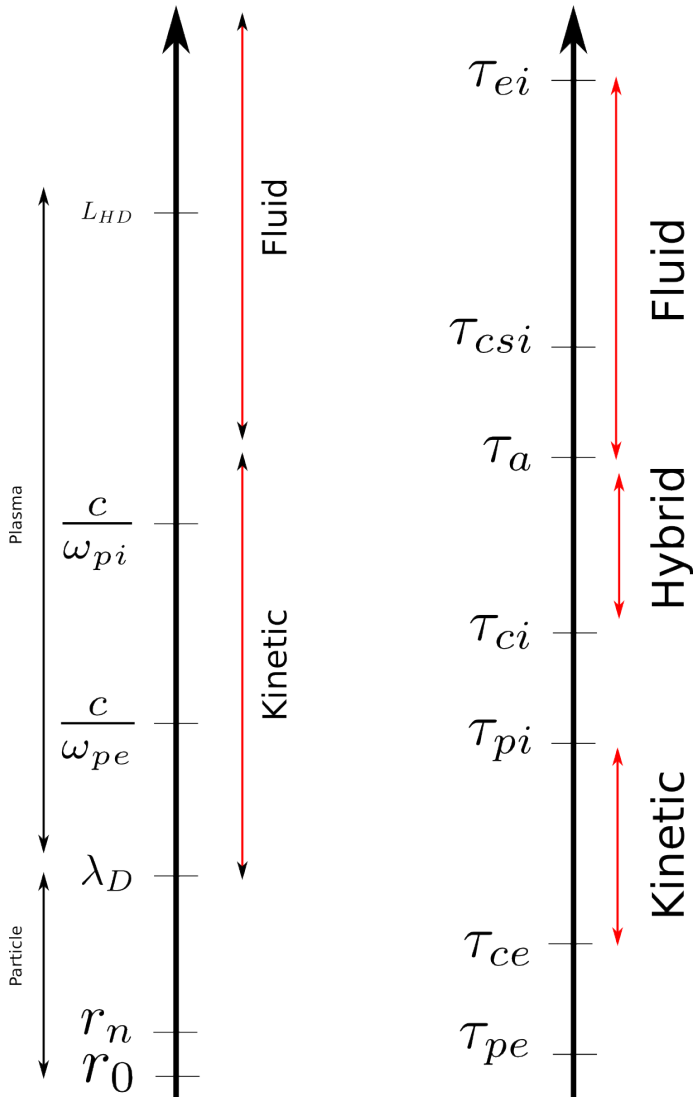
Plan of talk

1. Importance of numerical simulation
2. Types of differential equation
3. Finite difference form for derivatives
4. Flux-corrected-transport Algorithm
5. References

Importance of Numerical Simulations

- Analytical solutions exist only in limit cases.
  - Several theoretical approaches to describe plasmas—plasma equations. If analytic solution did exist for these equations, we could investigate physical phenomenon by applying these theories. However, plasma physics is full of nonlinearity and complexity and analytical solutions exist only in extremely rare cases.
- Simulations help in interpreting experiments and can provide detailed information which is difficult to measure
- Design and predict new experiments
- Innovative ideas can be tested on computers:
  - Most importantly, simulation can solve some problems which is out of analytical solutions or not realistic due to scale in experiments. For example, it is difficult to reproduce certain space plasma processes in laboratory but it is possible to have a numerical observatory. Using computer simulation we can attemp innovative ideas to predict new physics.

Numerical simulation methods



$$\omega_{pe} = \sqrt{\frac{4\pi ne^2}{m}} \quad \lambda_D = \frac{V_{thermal}}{\omega_{pe}} \propto \sqrt{\frac{T}{n}}$$

$$\lambda_{skindepth} = \frac{c}{\omega_{pe}} \quad \omega_c = \frac{eB}{mc}$$

- The fluid approach is suitable for treating large scale properties of plasmas involving mass, momentum, and flux transport
- Temperature, density, and net velocity of small volume/parcel of plasma is well defined and it is thus applicable for time-scales which allow enough collisions between plasma particles to thermalize the plasma
- In Hydrid model ions are treated kinetically, the electrons are assumed to be an inertia-less and quasi-neutral fluid, and the electromagnetic fields are treated in the low-frequency approximation
- kinetic description provides a more accurate treatment of many local and quasi-local processes

Fundamental steps in simulations

1. Set of equations with initial value and boundary conditions
2. Numerical (spatial and temporal discretization) of set of equations.
   - System of algebraic equations.
   - Various types of spatial discretization— finite-differences; finite element; finite volume; spectral methods.
   - Discretization leads to numerical error
3. Advance in time advance. For the time integration, various explicit or implicit schemes have been constructed. If implicit scheme is employed, we need to have a solver.
4. Diagnoses to analyse the approximate numerical solution

Types of differential equations

There are three main types of second-order partial differential equations

General prototypes of these are:
Wave equation : $q_{tt} - \gamma^2 q_{xx} - f(x) = 0 \longrightarrow$Hyperbolic
Heat equation : $q_t - \alpha^2 q_{xx} - f(x) = 0 \longrightarrow$Parabolic
Poisson equation : $q_{xx} - f(x) = 0 \longrightarrow$Elliptic

Why this geometric terminology??
All of the above equations can be written in a more general form:

$$a\Phi_{\xi\xi} + b\Phi_{\xi\chi} + c\Phi_{\chi\chi} + f(\Phi, \Phi_\xi, \Phi_\chi, \xi, \chi) = 0$$

The above equation is quite similar to the equation for conic sections:

$$ax^2 + bxy + cy^2 + d = 0$$

Relation between conic sections and PDEs

The conic section equation $ax^2+bxy+cy^2+d=0$ has discrminant b^2-4ac

$$\text{if } \begin{cases} b^2-4ac>0 & \text{we get hyperbola} \\ b^2-4ac=0 & \text{we get parabola} \\ b^2-4ac<0 & \text{we get ellopse} \end{cases}$$

The general second-order pde in two variables:

$$a\Phi_{\xi\xi}+b\Phi_{\xi\chi}+c\Phi_{\chi\chi}+f\left(\Phi,\Phi_{\xi},\Phi_{\chi},\xi,\chi\right)=0$$

is easily transformed to the hyperbolic equation $q_{tt}-\gamma^2 q_{xx}-f\left(x\right)=0$ in case $b^2-4ac>0$

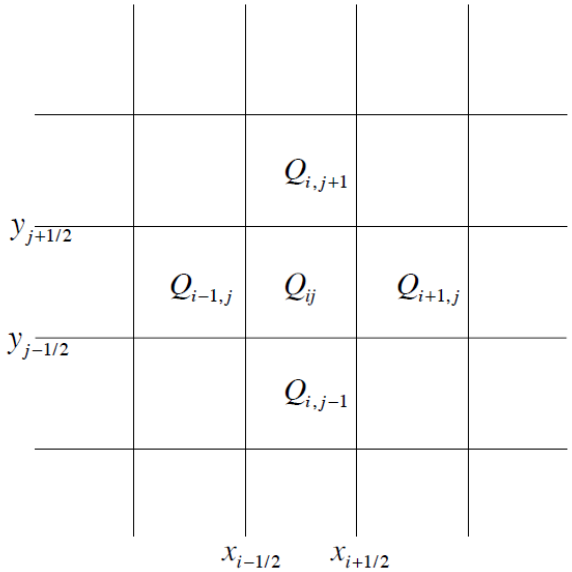
Properties of these PDEs

	Hyperbolic	Parabolic	Elliptic
Prototype	$q_{tt} - \gamma^2 q_{xx} - f(x) = 0$	$q_t - \alpha^2 q_{xx} - f(x) = 0$	$q_{xx} - f(x) = 0$
Eigenvalues	Real	Zero	Complex
Solutions	Wave-like, energy conserving	damping, diffusion,irreversibility	steady-state,no waves
Boundary Conditions	Cauchy(initial value problem)	Cauchy+Neumann/Dirichlet	Neumann/Dirichlet

- Although the classification of PDEs into hyperbolic, parabolic, and elliptic was designed around the appearance of second-order equations, we apply it to systems of first-order equations
- The methods we speak of in this talk are aimed at the solution of systems of first-order equations

Finite Volume vs Finite Difference

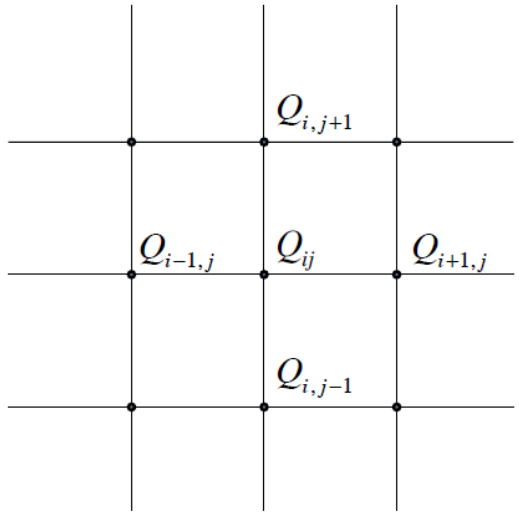
Finite Volume



Approximate values are averaged over cells:

$$Q_{ij}^n \approx \frac{1}{\Delta x \Delta y} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} q\left(x, y, t_n\right) dxdy$$

Finite Difference



Approximate values are evaluated on a grid:

$$Q_{ij}^n \approx q\left(x_i, y_j, t_n\right)$$

Methods to find finite difference

Methods for obtaining finite-difference expressions

- Taylor series expansion: the most common, but purely mathematical.
- Polynomial fitting or interpolation: the most general ways.

Taylor series is a subset of this method.
Example: let us find fourth order approximation to df/dx
For a five point stencil $[x_{i-2}, x_{i+2}]$ we have

$$\left.\frac{df}{dx}\right|_{x_i} = af_{i+2} + bf_{i+1} + cf_i + df_{i-1} + ef_{i-2} + O\left[(\Delta x)^4\right]$$

Now substitute Taylor series expansion for $f_{i\pm2}$and $f_{i\pm1}$in above expression and then equate the coefficients of like powers of Δx to get unknown coeifficients a, b, c *and* d.

continue...

$$
\begin{aligned}
a + b + c + d + e &= 0, \\
2a + b - d - 2e &= 1/\Delta x, \\
4a + b + d + 4e &= 0, \\
8a + b - d - 8e &= 0, \\
16a + b + d + 16e &= 0
\end{aligned}
$$

Solving above set of equations of form

$$
\begin{aligned}
\left.\frac{df}{dx}\right|_{x0} &= \frac{4}{3}\left(\frac{f_{i+1} - f_{i-1}}{2\Delta x}\right) \\
&\quad - \frac{1}{3}\left(\frac{f_{i+2} - f_{i-2}}{4\Delta x}\right) + O\left[(\Delta x)^4\right]
\end{aligned}
$$

Polynomial fitting

- We assume that the solution of the PDE can be approximated by a polynomial, and the values at the mesh points are exact.
- Over a three-point stencil $[x_i - 1, x_i + 1]$, we assume the local fitting function is $\bar{u}(x) = ax^2 + bx + c$. Applying the polynomials to the three points gives

$$u_{i-1} = ax_{i-1}^2 + bx_{i-1} + c, \quad u_i = ax_i^2 + bx_i + c, \quad u_{i+1} = ax_{i+1}^2 + bx_{i+1} + c,$$

- Solve for a, b, c, we obtain the Lagrange Interpolation Polynomial,

$$u(x) = u_{i-1}\frac{(x-x_i)(x-x_{i+1})}{(x_{i-1}-x_i)(x_{i-1}-x_{i+1})} + u_i\frac{(x-x_{i-1})(x-x_{i+1})}{(x_i-x_{i-1})(x_i-x_{i+1})} + u_{i+1}\frac{(x-x_{i-1})(x-x_i)}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)}$$

- Differentiating $u(x)$ with respect to x, we obtain

$$u_x = (u_{i+1} - u_{i-1})/(2h), \quad u_{xx} = (u_{i+1} - 2u_i + u_{i-1})/h^2$$

Examples of Difference Formula

Using either of the above approaches, we can obtain the finite-difference formula
The first-derivatives:

$$u'(x_0) = \frac{u_1 - u_0}{h} + O(h)$$
$$u'(x_0) = \frac{u_1 - u_{-1}}{2h} + O(h^2)$$
$$u'(x_0) = \frac{-u_2 + 8u_1 - 8u_{-1} + u_{-2}}{12h} + O(h^4)$$

Second -derivatives:

$$u''(x_0) = \frac{u_2 - 2u_1 + u_0}{h^2} + O(h)$$
$$u''(x_0) = \frac{u_1 - 2u_0 + u_{-1}}{h^2} + O(h^2)$$
$$u''(x_0) = \frac{-u_2 + 16u_1 - 30u_0 + 16u_{-1} - u_{-2}}{12h^2} + O(h^4)$$

continue...

Mixed derivatives:

$$u_{xy}(x_0, y_0) = \frac{1}{4h^2}\left(u_{1,1} - u_{1,-1} - u_{-1,1} + u_{-1,-1}\right) + O\left(h^2\right)$$

Laplacian :

$$\nabla^2 u(x_0, y_0) = \frac{1}{h^2}\left(u_{1,0} + u_{0,1} + u_{-1,0} + u_{0,-1} - 4u_{0,0}\right) + O\left(h^2\right)$$

$$\nabla^2 u(x_0, y_0) = \frac{1}{12h^2}\left(-60u_{0,0} + 16\left(u_{1,0} + u_{0,1} + u_{-1,0} + u_{0,-1}\right)\right.$$
$$\left. - \left(u_{2,0} + u_{0,2} + u_{-2,0} + u_{0,-2}\right)\right) + O\left(h^4\right)$$

Advection equation and FD Schemes

We now consider the finite-difference scheme for a simplest case: linear advection equation

$$u_t + a u_x = 0$$
$$u(x,0) = u_0(x)$$

where $a$is constant. We also assume $a > 0$. Its solution is simply a translation of the u_0 : $u(x,t) = u_0(x - at)$
Combine the spatial and temporal discretization together, we list some finite-difference scheme below

$$Upwind\,(first - order) \quad \frac{u_i^{n+1} - u_i^n}{k} + a\frac{u_i^n - u_{i-1}^n}{h} = 0$$

$$Lax - Friedrichs\,scheme, \quad \frac{u_i^{n+1} - \frac{1}{2}\left(u_{i+1}^n + u_{i-1}^n\right)}{k} + a\frac{u_{i+1}^n - u_{i-1}^n}{2h} = 0$$

$$Lax - Wendroff\,scheme, \quad u_i^{n+1} = u_i^n - \frac{ak}{2h}\left(u_{i+1}^n - u_{i-1}^n\right) + \frac{a^2k^2}{2h^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)$$

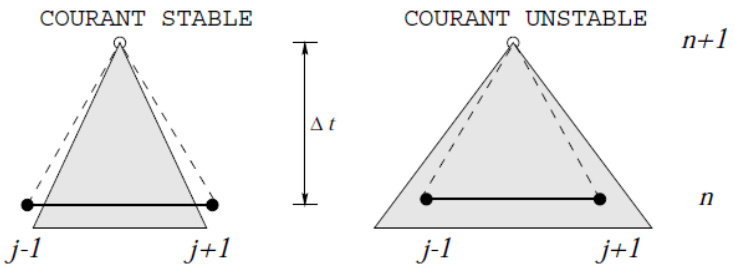
In general, an explicit finite difference scheme can be expressed as

$$u_i^{n+1} = F\left(u_{i-l}^n, u_{i-l+1}^n, ..., u_{i+r}^n\right) = \sum_{k=-l}^{r} a_k u_{i+k}^n$$

Properties of FD scheme

- **Consistency:** An finite-difference discretization of a PDE is consistent if the finite-difference equations converge to the PDE, i.e., the truncation error vanishes as grid spacing and time step tend to zero.
- **Stability:** the errors from any source will not grow unbounded with time
- **Convergence:** the solution of the finite-difference equations converge to the true solution of the PDE as grid spacing and time step tend to zero.

Courant-Fredrichs-Levy (CFL) Condition



- For a finite-difference scheme, $u_i^{n+1} = F(u_{i-l}^n, u_{i-l+1}^n, ..., u_{i+r}^n)$, the numerical domain dependence of (x_i, t_n) is $[x_{i-l}, x_{i+r}]$
- For stability, physical domain of dependence should be contained within numerical domain of dependence
- This gives a Courant-Fridrichs-Levy (CFL) condition on the ratio of h and k. For the linear advection equation with $a > 0$, the CFL condition is

$$0 \leq \frac{ak}{h} \leq 1$$

Stability Analysis: Von neumann Method

Theorem :

- A finite-difference scheme with constant coefficients

$u_i^{n+1} = \sum_{m=-l}^{r} a_m u_{i+m}^n$ is stable iff $\hat{G}(\xi) := \sum_{m=-l}^{r} a_k e^{-ik\xi}$ satisfies $\max_{-\pi \leq \xi \leq \pi} \left| \hat{G}(\xi) \right| \leq 1$

- Forward time central space scheme given by $\frac{u_i^{n+1} - u_i^n}{k} + a \frac{u_{i+1}^n - u_{i-1}^n}{2h} = 0$ is unstable
- For this case, $\hat{G}(\xi) = 1 + \iota \lambda sin\xi$, whose magnitude is not bounded by 1

Consistency and Truncation Error

Suppose our difference scheme is represented as $u^n = Fu^n$. The truncation error is defined as

$$e^n = \frac{u^{n+1} - Fu^n}{k}$$

A finite-difference scheme is called consistent if $e^n \longrightarrow 0$ as $k \longrightarrow 0$ and $h \longrightarrow 0$.
For finite difference scheme $u_i^{n+1} = \sum_{m=-l}^{r} a_m u_{i+m}^n$, the necessary condition for consistency is

$$\sum_{m=-l}^{r} a_m = 1$$

If $e^n = O\left(k^p, h^q\right)$,then the scheme is called of order (p, q). It is easy to check the upwind method has accuracy of order $(1, 1)$, and the Lax-Wendroff method has accuracy of order $(2, 2)$.

Lax-Richtmyer equivalence theorem

Lax-Richtmyer equivalence theorem : Given a properly posed initial-value problem and a finite difference approximation to it that satisfies the consistency condition, stability is the necessary and sufficient condition for convergence.

- It shows that for an initial-value problem which has been discretised with a consistent finite- difference operator, the concept of stability and convergence are interchangable.
- Thus, proving that the numerical solution is convergent will not only validate that the discrete form of the equations represents a faithful representation of the continuum ones, but also that the solution will be bounded at all times

Conservation form

- During evolution if the physical scale of the simulated disturbance approaches the minimum scale resolvable on the numerical mesh errors in the numerical solutions increase dramatically.
- For some equations smooth initial data guarantee a smooth solution at all later times. In such cases difficulties associated with poor numerical resolution can be avoided by using a sufficiently fine computational mesh.
- However, if the equations allow an initially smooth field to develop shocks or discontinuities no hope of maintaining adequate numerical resolution throughout the simulation
- Special numerical techniques must be used to control the development of overshoots and undershoots in the vicinity of the shock and ensure that the numerical solution converges to the correct solution as the spatial grid interval and the time step approach zero.

Case in point-Burger equation

Advective form:

$$\frac{\partial \psi}{\partial t} + \psi^2 \frac{\partial \psi}{\partial x} = 0$$

Upstream finite-difference approximation

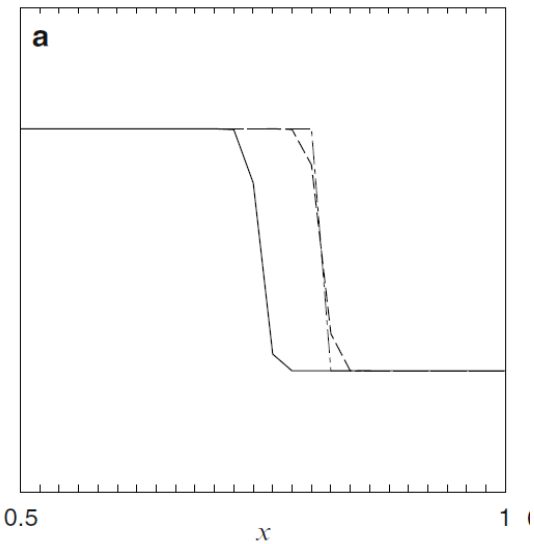
$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} + \left(\frac{\phi_i^n + \phi_{i-1}^n}{2} \right)^2 \left(\frac{\phi_i^n - \phi_{i-1}^n}{\Delta x} \right) = 0$$

Flux form:

$$\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x} \left(\frac{\psi^3}{3} \right) = 0$$

Upstream approximation

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} + \left(\frac{\left(\phi_i^n\right)^3 - \left(\phi_{i-1}^n\right)^3}{3\Delta x} \right) = 0$$



Conservation Laws and Weak solutions

Many of the partial differential equations arising in fluid dynamics can be expressed as a system of conservation laws of the form

$$\frac{\partial \boldsymbol{\rho}}{\partial t} + \sum_j \frac{\partial \mathbf{f}_j(\boldsymbol{\rho})}{\partial x_j} = 0$$

The rate of change of $\boldsymbol{\rho}$ at each point is determined by the convergence of the fluxes $\mathbf{f}_j$ at that point.
let us consider scalar conservatio law on the unbounded domain $-\infty < x < \infty$

$$\frac{\partial \psi}{\partial t} + \frac{\partial f(\psi)}{\partial x} = 0$$

continue...

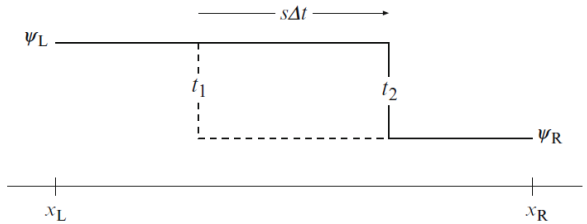
Integrating this conservation law over the intervals $[x_1, x_2]$and $[t_1, t_2]$, one obtains

$$\int_{x_1}^{x_2} \psi(x, t_2)\, dx = \int_{x_1}^{x_2} \psi(x, t_1)\, dx + \int_{t_1}^{t_2} f(\psi(x_1, t))\, dt - \int_{t_1}^{t_2} f(\psi(x_2, t))\, dt$$

Total change in ψ over the region$x_1 \leq x \leq x_2$ is determined by the time-integrated fluxes through the boundary of that region
Unlike the differential form, the integral form can be satisfied by piecewise-continuous functions

Weak Solution:
If ψ satisfies the integral equation on every subdomain $[x_1, x_2] \times [t_1, t_2]$, then ψ is a weak solution of the conservation law.
Differentiable weak solutions are also solutions to the PDE and are uniquely determined by the initial data.
Nondifferentiable weak solutions may, however, be nonunique

Reimann Problem

- Conservation law + Initial data in form of step function = Reimann problem
- 

$$\psi(x,t) = \begin{cases} \psi_L = \psi(x_L) & \text{if } x - st < 0, \\ \psi_R = \psi(x_R) & \text{otherwise,} \end{cases}$$

- Calculation of Shock Speed:

$$\int_{x_L}^{x_R} \psi(x,t)\,dx = (st - x_L)\,\psi_L + (x_R - st)\,\psi_R \longrightarrow \frac{d}{dt}\int_{x_L}^{x_R} \psi(x,t)\,dx = s(\psi_L - \psi_R)$$

Integrating conservation Eq over interval $[x_L, x_R]$, —> $\frac{d}{dt}\int_{x_L}^{x_R} \psi(x,t)\,dx = f(\psi_L) - f(\psi_R)$
Above two expressions imply that

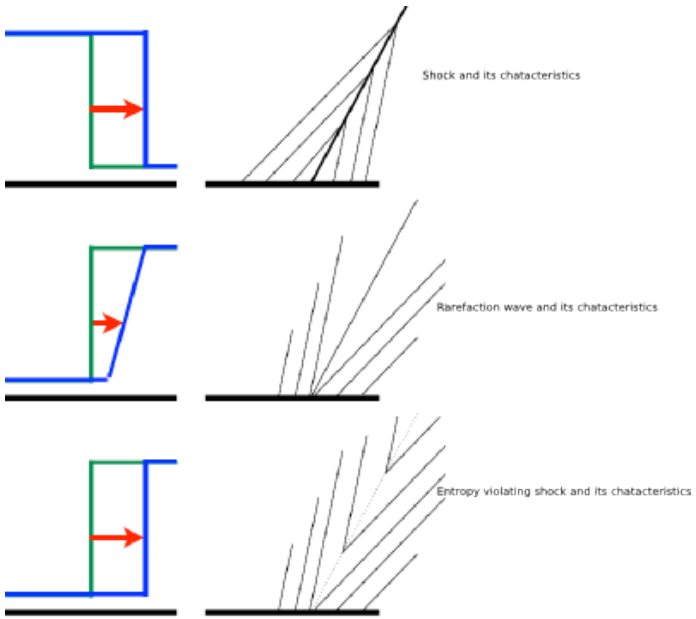
$$s = \frac{f(\psi_L) - f(\psi_R)}{\psi_L - \psi_R}$$

continue...

For given initial condition one may have may nondifferentiable weak solutions and it is necessary to select the physically relevant solution.
When the solutions to equations representing real physical systems develop discontinuities, one of the physical assumptions used to derive those equations is often violated.
One strategy for selecting the physically significant weak solution would be to conduct a series of viscous simulations with progressively smaller viscosities and choose the weak solution toward which the viscous solutions converge. This, of course, is a highly inefficient strategy, and it may be impossible to implement in actual simulations of high-Reynolds-number flow, where any realistic value for the molecular viscosity may be too low to significantly influence the solution on the spatial scales resolvable on the numerical grid. In addition, any attempt to include realistic viscosities in the numerical solution reintroduces precisely those mathematical complications that were eliminated when the full physical system was originally approximated by the simpler inviscid model

Entropy-Consistent Solutions

- Alternative criteria required for selecting the physically relevant weak solution.
- An entropy function is a function that is conserved when the solution is smooth, but changes in magnitude at a discontinuity.
- The thermodynamic entropy of a gas, for example, increases across a shock but is constant in smooth flow.



A discontinuity propagating with speed s must satisfy the entropy condition

$$f^{'}(q_l) < s < f^{'}(q_r)$$

Convergence of finite volume methods

Two difficulties arise in attempting to compute discontinuous solutions to PDE:

1. Numerical scheme might converge to a function that is not a weak solution of the conservation law
2. Numerical scheme may converge to a genuine weak solution, but it may fail to converge to the physically relevant entropy consistent solution

Conservative form:

$$\frac{\partial \psi}{\partial t} + \frac{\partial f(\psi)}{\partial x} = 0 \longrightarrow \frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} + \left(\frac{F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}}}{\Delta x} \right) = 0$$

Here,

$\phi_j^n \approx \frac{1}{\Delta x} \int_{x_j - \frac{\Delta x}{2}}^{x_j + \frac{\Delta x}{2}} \psi(x, t^n)\, dx$ and $F_{j+\frac{1}{2}} \approx \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} f\left(\psi\left(x_j + \frac{\Delta x}{2}, t\right)\right) dt$

For conservative form, Lax-Wendroff theorem ensures that if the numerical solution does converge, it will converge to a weak solution

Monotone Schemes

- Lax-Wendroff scheme does not gaurantee entropy-consistent weak solutions
- If a consistent numerical scheme in conservation form is monotone then it is bound to have entropy-consistent weak solutions

If the scheme is expressed in the functional form

$$\phi_j^{n+1} = H\left(\phi_{j-p}^n, ..., \phi_{j+q+1}^n\right)$$

the condition that the method be monotone is

$$\frac{\partial H\left(\phi_{j-p}, ..., \phi_{j+q+1}\right)}{\partial \phi_i} \geq 0$$

for each integer i in the interval $[j-p, j+q+1]$

- Monotone methods are restrictive in the sense that they are at most first order accurate

Total Variation Diminishing Methods

- First order methods are not efficient and better results can often be obtained using higher-order schemes

The total variation of a one-dimensional grid-point function is defined as

$$TV(\phi) = \sum_{j=1}^{N-1} |\phi_{j+1} - \phi_j|$$

where N is the total number of grid points in the numerical domain.

- A numerical method is total variation nonincreasing if

$$TV\left(\phi^{n+1}\right) < TV\left(\phi^{n}\right)$$

- For TVD scheme the solutions to a consistent finite-volume method in conservation form are guaranteed to converge to weak solutions of the exact conservation law
- It can be shown that a TVD scheme is also monotonicity preserving
- TVD can sometimes be a weakcondition since only the totalvariation decreases, it is possible to create small wiggles while still decreasing the total

Numerical schemes for conservation laws

Numerical schemes fall in two categories

- Flux manipulation methods: The basic idea is to add a switch such that the scheme becomes first order near discontinuity and remains high order in the smooth region.
  - artificial viscosity methods
  - total variation diminishing (TVD)
  - central scheme
  - relaxation schemes
  - flux-correction transport(FCT)
- Higher-order Godunov methods: MUSCL, piecewise parabolic method (PPM), essentially non-oscillatory (ENO) schemes

FCT algorithm

The conservation or flux form of the conservation equation

$$\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial f(\rho,x,t)}{\partial x} = 0$$

is

$$\rho_i^{n+1} = \rho_i^n - \Delta x_i^{-1}\left[F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}\right]$$

- Here ρ and f are defined on the spatial grid points x_i and temporal grid points t^n, and Δx_i is the cell width associated with cell i.
- The $F_{i+\frac{1}{2}}$ are called numerical fluxes, and are functions of f and ρ at one or more of the time levels t^n.
- The functional dependence of F on f and ρdefines the particular discrete approximation.

continue...

FCT algorithm can be implemented as follows:

1. Compute $F^L_{i+\frac{1}{2}}$, the "low order fluxes," using a method guaranteed not to generate unphysical values in the solution for the problem at hand.
2. Compute $F^H_{i+\frac{1}{2}}$, the "high order fluxes" using a method chosen to be accurate in smooth regions for the problem at hand.
3. Define the "anti-diffusive fluxes"

$$A_{i+\frac{1}{2}} \equiv F^H_{i+\frac{1}{2}} - F^L_{i+\frac{1}{2}}$$

4. Compute the time advanced low order ("transported and diffused") solution:

$$q_i^{td} = q_i^n - \Delta x_i^{-1} \left[F^L_{i+\frac{1}{2}} - F^L_{i-\frac{1}{2}} \right]$$

continue...

1.

5. Limit the anti-diffusive fluxes in a manner such that q_i^{n+1} as computed in step 6 below does not take on nonphysical values:

$$A^c_{i+\frac{1}{2}} = c_{i+\frac{1}{2}} A_{i+\frac{1}{2}}, \quad 0 \leq c_{i+\frac{1}{2}} \leq 1$$

6. Apply the limited anti-diffusive fluxes: equation here

$$q_i^{n+1} = q_i^{td} - \Delta x^{-1} \left[A^c_{i+\frac{1}{2}} - A^c_{i-\frac{1}{2}} \right]$$

It is important to realize that the success of FCT algorithm relies on the step 5.

Flux Correction: Boris and Book

Boris and Book did not actually give a formula for $c_{j+\frac{1}{2}}$, but offered the following expression for the corrected fluxes:

$$A^c_{j+\frac{1}{2}} = S_{j+\frac{1}{2}} \max\left\{0, \min\left[\left|A_{j+\frac{1}{2}}\right|, S_{j+\frac{1}{2}}\left(\phi^{td}_{j+2} - \phi^{td}_{j+1}\right)\frac{\Delta x}{\Delta t}, S_{j+\frac{1}{2}}\left(\phi^{td}_{j} - \phi^{td}_{j-1}\right)\frac{\Delta x}{\Delta t},\right]\right\}$$

where

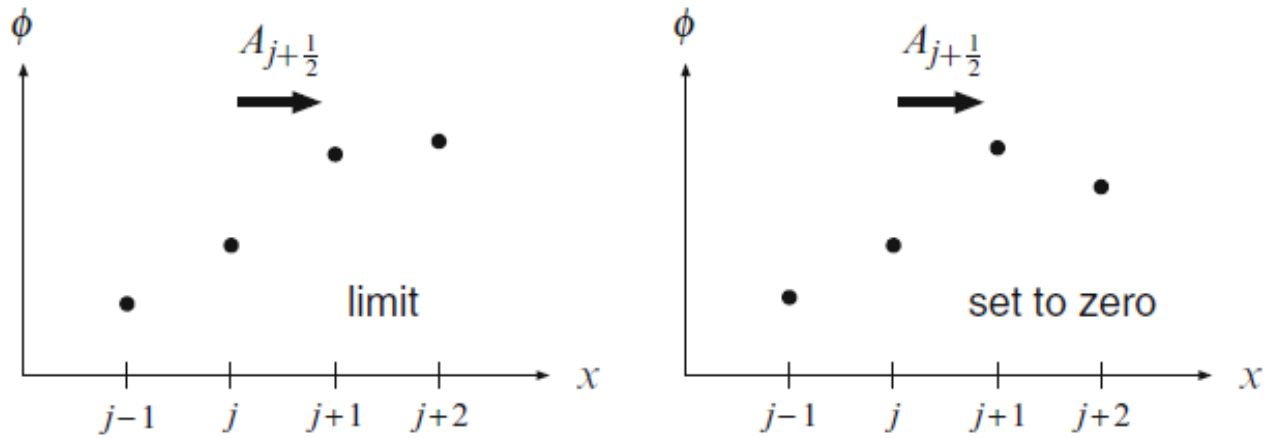
$$S_{j+\frac{1}{2}} = \text{sgn}\left(A_{j+\frac{1}{2}}\right)$$

The logic behind this formula can be understood by considering the case where $A_{j+\frac{1}{2}} \geq 0$, so the preceding may be written as

continue...

$$\frac{\Delta t}{\Delta x} A^c_{j+\frac{1}{2}} = \max\left\{0, \min\left[A_{j+\frac{1}{2}} \frac{\Delta t}{\Delta x}, \left(\phi^{td}_{j+2} - \phi^{td}_{j+1}\right), \left(\phi^{td}_{j} - \phi^{td}_{j-1}\right)\right]\right\}$$

Below figure shows two possible configurations in which the flux limiter modifies an antidiffusive flux



- The first argument of the "min" function in above expression gives the increase in ϕ^n_{j+1}, and the decrease in ϕ^n_j, that would be produced by the uncorrected antidiffusive flux $A_{j+\frac{1}{2}}$.
- The second argument of the min function renders $\phi_{j+1} > \phi_{j+2}$so that the corrected flux is not large enough to generate a new maximum

2D FCT

The generalized 2-D transport equation of interest in Eulerian form is:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} = c(x,y)\frac{\partial f}{\partial x} + d(x,y)\frac{\partial g}{\partial y} + s(x,y) \qquad (1)$$

Here ρ is the dependent flow property, x and y are the spatial dimensions, t is time and u and v are velocities in x and y directions respectively. The first and second term on the right hand side of the above equation represents flux in x and y direction respectively while the third term represents the algebraic source terms.

the spatial domain, over which the equation is evolved, is discretized into cells

The value of the flow variables remain uniform throughout the individual cell . The interface area of the cell surface oriented in x and y direction are represented by $A_{i,j}$ *and* $B_{i,j}$respectively and the cell volume is represented by $\Lambda_{i,j}$. Here we have assumed uniform mesh so that $A_{i,j} = \Delta y$ *and* $B_{i,j} = \Delta x$ and the cell volume $\Lambda_{i,j} = \Delta x \Delta y$.

Step 1

Let $\rho^0_{i,j}$represent the initial value of the dependent variable ρ. As discussed in earlier earlier, the first step in FCT methodology is to convect the initial field $\rho^0_{i,j}$. Thus the convected field is given by,

$$\begin{aligned}
\Lambda_{i,j}\rho^*_{i,j} &= \Lambda_{i,j}\rho^0_{i,j} \\
&+\Delta t\rho^0_{i-\frac{1}{2},j}A_{i-\frac{1}{2},j}u_{i-\frac{1}{2},j} - \Delta t\rho^0_{i+\frac{1}{2},j}A_{i+\frac{1}{2},j}u_{i+\frac{1}{2},j} \\
&+\Delta t\rho^0_{i,j-\frac{1}{2}}B_{i,j-\frac{1}{2}}v_{i,j-\frac{1}{2}} - \Delta t\rho^0_{i,j+\frac{1}{2}}B_{i,j+\frac{1}{2}}v_{i,j+\frac{1}{2}} \\
&+\gamma^x_{i+\frac{1}{2},j}\left(\rho^0_{i+1,j} - \rho^0_{i,j}\right) - \gamma^x_{i-\frac{1}{2},j}\left(\rho^0_{i,j} - \rho^0_{i-1,j}\right) \\
&+\gamma^y_{i,j+\frac{1}{2}}\left(\rho^0_{i,j+1} - \rho^0_{i,j}\right) - \gamma^y_{i,j-\frac{1}{2}}\left(\rho^0_{i,j} - \rho^0_{i,j-1}\right)
\end{aligned} \tag{2}$$

Here Δt is the time step and the parameter γ controls the phase-error.

continue...

The velocities and the areas at the interface are given by:

$$u_{i+\frac{1}{2},j} = \frac{1}{2}\left(u_{i,j} + u_{i+1,j}\right) \tag{3}$$

$$v_{i,j+\frac{1}{2}} = \frac{1}{2}\left(v_{i,j} + v_{i,j+1}\right) \tag{4}$$

$$A_{i+\frac{1}{2},j} = \frac{1}{2}\left(A_{i,j} + A_{i+1,j}\right) \tag{5}$$

$$B_{i,j+\frac{1}{2}} = \frac{1}{2}\left(B_{i,j} + B_{i,j+1}\right) \tag{6}$$

Step 2

The second step in the FCT methodology is to add source terms to the convected field calculated in step 1. We then have the transported field given by

$$\begin{aligned} \Lambda_{i,j}\rho^{T}_{i,j} &= \Lambda_{i,j}\rho^{*}_{i,j} \\ &+\tfrac{1}{4}\Delta t c_{i,j}\left(A_{i+\frac{1}{2},j}+A_{i-\frac{1}{2},j}\right)\left(f_{i+1,j}-f_{i-1,j}\right) \\ &+\tfrac{1}{4}\Delta t d_{i,j}\left(B_{i,j+\frac{1}{2}}+B_{i,j-\frac{1}{2}}\right)\left(g_{i,j+1}-g_{i,j-1}\right) \\ &+\Delta t \Lambda_{i,j} s_{i,j} \end{aligned} \tag{7}$$

Step 3

This step comprises of diffusing the transported solution.

$$\begin{aligned}\Lambda_{i,j}\tilde{\rho}_{i,j} &= \Lambda_{i,j}\rho^{T}_{i,j} \\ +\nu_{i+\frac{1}{2},j}\Lambda_{i+\frac{1}{2},j}\left(\rho^{0}_{i+1,j}-\rho^{0}_{i,j}\right) &- \nu_{i-\frac{1}{2},j}\Lambda_{i-\frac{1}{2},j}\left(\rho^{0}_{i,j}-\rho^{0}_{i-1,j}\right) \\ +\lambda_{i,j+\frac{1}{2}}\Lambda_{i,j+\frac{1}{2}}\left(\rho^{0}_{i,j+1}-\rho^{0}_{i,j}\right) &- \lambda_{i,j-\frac{1}{2}}\Lambda_{i,j-\frac{1}{2}}\left(\rho^{0}_{i,j}-\rho^{0}_{i,j-1}\right)\end{aligned} \tag{8}$$

Here $\tilde{\rho}_{i,j}$represents the transported and diffused density field profile. ν and λ are the diffusion coefficients and will be defined later. The interface-averaged volumes $\Lambda_{i+\frac{1}{2},j}$ and $\Lambda_{i,j+\frac{1}{2}}$ occurring in above expression are defined as

$$\begin{aligned}\Lambda_{i+\frac{1}{2},j} &= \frac{1}{2}\left(\Lambda_{i+1,j}+\Lambda_{i,j}\right) \\ \Lambda_{i,j+\frac{1}{2}} &= \frac{1}{2}\left(\Lambda_{i,j+1}+\Lambda_{i,j}\right)\end{aligned} \tag{9}$$

continue...

The boundary interface volumes in x and y direction are taken as

$$\begin{aligned} \Lambda_{\frac{1}{2},j} &= \Lambda_{1,j} \\ \Lambda_{M+\frac{1}{2},j} &= \Lambda_{M,j} \\ \Lambda_{i,\frac{1}{2}} &= \Lambda_{i,1} \\ \Lambda_{i,N+\frac{1}{2}} &= \Lambda_{i,N} \end{aligned} \tag{10}$$

Here M and N are the total number of cells in x and y direction respectively. It is important to note that the anti-diffusive fluxes are computed using $\rho^{T}_{i,j}$ rather than $\tilde{\rho}_{i,j}$. Hence, the evolution of the density field is divided into convective transport, source addition and diffusion stages.

Step 4

The raw, uncorrected anti-diffusive fluxes in x and y direction are given by

$$F^{ad}_{i+\frac{1}{2},j} = \mu_{i+\frac{1}{2},j}\Lambda_{i+\frac{1}{2},j}\left(\rho^T_{i+1,j} - \rho^T_{i,j}\right) \tag{11}$$

$$G^{ad}_{i,j+\frac{1}{2}} = \kappa_{i,j+\frac{1}{2}}\Lambda_{i,j+\frac{1}{2}}\left(\rho^T_{i,j+1} - \rho^T_{i,j}\right) \tag{12}$$

Here the anti-diffusive coefficients are defined as

$$\mu_{i+\frac{1}{2},j} = \frac{1}{6} - \frac{1}{6}\epsilon^2_{i+\frac{1}{2},j} \tag{13}$$

$$\kappa_{i,j+\frac{1}{2}} = \frac{1}{6} - \frac{1}{6}\alpha^2_{i,j+\frac{1}{2}} \tag{14}$$

and the modified local Courant numbers $\epsilon$and $\alpha$are defined as

$$\epsilon_{i+\frac{1}{2},j} = A_{i+\frac{1}{2},j}u_{i+\frac{1}{2},j}\frac{\Delta t}{2}\left(\frac{1}{\Lambda_{i+\frac{1}{2},j}}\right) \tag{15}$$

Step 5

Now using the raw anti-diffusive fluxes defined in equations 11 and 12 we compute the corrected anti-diffusion fluxes

$$F^c_{i+\frac{1}{2},j} = s_{i+\frac{1}{2},j} \max\left\{0, \min\left[|F^{ad}_{i+\frac{1}{2},j}|, s_{i+\frac{1}{2},j}\Lambda_{i+1,j}\left(\tilde{\rho}_{i+2,j} - \tilde{\rho}_{i+1,j}\right), s_{i+\frac{1}{2},j}\Lambda_{i,j}\left(\tilde{\rho}\right.\right.\right. \tag{17}$$

$$G^c_{i,j+\frac{1}{2}} = s_{i,j+\frac{1}{2}} \max\left\{0, \min\left[|G^{ad}_{i,j+\frac{1}{2}}|, s_{i,j+\frac{1}{2}}\Lambda_{i,j+1}\left(\tilde{\rho}_{i,j+2} - \tilde{\rho}_{i,j+1}\right), s_{i,j+\frac{1}{2}}\Lambda_{i,j}\left(\tilde{\rho}\right.\right.\right. \tag{18}$$

Here, $s_{i+\frac{1}{2},j}$ refers to sign of the quantity $[\tilde{\rho}_{i+1,j} - \tilde{\rho}_{i,j}]$ and $s_{i,j+\frac{1}{2}}$ refers to sign of the quantity $[\tilde{\rho}_{i,j+1} - \tilde{\rho}_{i,j}]$. The above form of flux correction method is the extension of the flux correction method proposed by Boris and Book []to two dimensions. Here the flux correction is first implemented in one direction (equation 17) and then it is implemented in other direction (equation 18). Truly two dimension flux correction was first proposed by Zalesak [].

Step 6

Once the corrected anti-diffusive fluxes are obtained, the evolved density field at advanced time is given by

$$\rho^{n}_{i,j} = \tilde{\rho}_{i,j} - \frac{1}{\Lambda_{i,j}} \left(F^{c}_{i+\frac{1}{2},j} - F^{c}_{i-\frac{1}{2},j} + G^{c}_{i,j+\frac{1}{2}} - G^{c}_{i,j-\frac{1}{2}} \right) \tag{19}$$

We have yet to define the diffusion coefficients $\nu$and λ. According to Boris and Book these diffusion coefficients are defined as

$$\gamma^{x}_{i+\frac{1}{2},j} = \gamma^{y}_{i,j+\frac{1}{2}} = 0 \tag{20}$$

$$\nu_{i+\frac{1}{2},j} \equiv \frac{1}{6} + \frac{1}{3}\epsilon^{2}_{i+\frac{1}{2},j} \tag{21}$$

$$\lambda_{i,j+\frac{1}{2}} \equiv \frac{1}{6} + \frac{1}{3}\alpha^{2}_{i,j+\frac{1}{2}} \tag{22}$$

continue...

As shown by Boris and Book, the above choice of diffusion coefficients reduce the relative phase errors in convection on a locally uniform grid to fourth order. The other choice of diffusion coefficients that reduce the phase error to sixth order [] is as given below.

$$\gamma^x_{i+\frac{1}{2},j} \equiv \frac{1}{5} + \frac{1}{5}\epsilon^2_{i+\frac{1}{2},j} \tag{23}$$

$$\gamma^y_{i,j+\frac{1}{2}} \equiv \frac{1}{5} + \frac{1}{5}\alpha^2_{i,j+\frac{1}{2}} \tag{24}$$

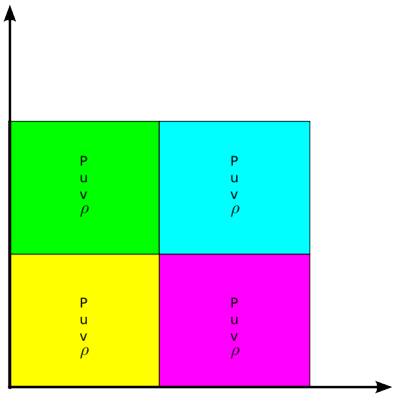
$$\nu_{i+\frac{1}{2},j} \equiv -\frac{1}{30} + \frac{4}{30}\epsilon^2_{i+\frac{1}{2},j} \tag{25}$$

$$\lambda_{i,j+\frac{1}{2}} \equiv -\frac{1}{30} + \frac{4}{30}\alpha^2_{i,j+\frac{1}{2}} \tag{26}$$

Subroutines in FCT package

- lcpgrd2d $\longrightarrow$This subroutine constructs grid for simulation. Cartesian, Cylindrical, Spherical
- lcpfct2d$\longrightarrow$This subroutine advances solution in time
- cnvfct2d $\longrightarrow$This subroutine is to be used when equation is in convective form.
- lcpsrc2d $\longrightarrow$This subroutine can be used when the equation contains source term. source term can either be in divergence form , gradient form , or constant.

Example: 2D Reimann problem



Computational domain is divided into four quadrants and each of them is then assigned the initial state.
Euler's equations of gas dynamics are then solved to evolve the initial state.
Symmetric or free boundary conditions for computational domain

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ E\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho u\\ \rho u^2\\ \rho uv\\ Eu\end{bmatrix}+\frac{\partial}{\partial y}\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2\\ Ev\end{bmatrix}=\begin{bmatrix}0\\ -\frac{\partial p}{\partial x}\\ -\frac{\partial p}{\partial y}\\ -\frac{\partial (pu)}{\partial x}-\frac{\partial (pv)}{\partial y}\end{bmatrix}$$

Here pressure p is related to the conserved quantities through the equation of state

$$p=(\gamma-1)\left[E-\frac{1}{2}\rho\left(u^2+v^2\right)\right]$$

$\gamma = 1.4$ and Symmetric or free boundary conditions haves been employed.

continue...

Cases		Left				Right			
		p	ρ	u	v	p	ρ	u	v
3	Top	0.3000	0.5323	1.2060	0.0000	1.5000	1.5000	0.0000	0.0000
	Bottom	0.0290	0.1380	1.2060	1.2060	0.3000	0.5323	0.0000	1.2060

```
clc; clear all; close all;
addpath('FluxCorrTrans_2July2016','UtilFun');
global rhon run rvn ergn
global xg yg dx dy
%-------------------------------------------------%
ny=400; Ly=1; nx=400; Lx=1;
maxstp=5000; courant=0.4; dtmax=1.0E-2; dt=1.0e-5;
config=3;

FCT2D_globalvar(nx,ny,0,0,1,1);
Rie_GlobalVar(nx,Lx,ny,Ly,config);
FCT2D_grid(xg,yg,1);

for istep=1:maxstp

    u(1:nx,1:ny)=run(1:nx,1:ny)./rhon(1:nx,1:ny);
    v(1:nx,1:ny)=rvn(1:nx,1:ny)./rhon(1:nx,1:ny);
    vmax=max(max(ergn(1:nx,1:ny)./rhon(1:nx,1:ny)));

    vmax=sqrt(vmax);
    dtnew=courant*min(dx,dy)/vmax;
    dt=min([dtmax 1.25*dt dtnew]);
```

```
23
24      %% plotting and diagnostics
25
26      Rie_gasdyn2d(dt,1,nx,1,ny);
27      time=time+dt;
28  end
29  close(movObj);
30  rmpath('FluxCorrTrans_2July2016','UtilFun');
```

```
function []=Rie_gasdyn2d(dt,i1,in,j1,jn)

global ii_1TOm1 ii_2TOm1 ii_1TOm1m
global jj_1TOm2 jj_2TOm2 jj_1TOm2m
global m1p m2p

global mpin_i mpin_j u v rhoo ruo rvo ergo
global uin vin pre mpuin mpvin unit
global rhon run rvn ergn gammam

rhoo(ii_1TOm1,jj_1TOm2)=rhon(ii_1TOm1,jj_1TOm2);
ruo(ii_1TOm1,jj_1TOm2)=run(ii_1TOm1,jj_1TOm2);
rvo(ii_1TOm1,jj_1TOm2)=rvn(ii_1TOm1,jj_1TOm2);
ergo(ii_1TOm1,jj_1TOm2)=ergn(ii_1TOm1,jj_1TOm2);

for it=1:2

    dtsub=0.5*dt*double(it);

    u(ii_1TOm1,jj_1TOm2)=run(ii_1TOm1,jj_1TOm2)./rhon(
      ii_1TOm1,jj_1TOm2);
    v(ii_1TOm1,jj_1TOm2)=rvn(ii_1TOm1,jj_1TOm2)./rhon(
```

```
    ii_1TOm1,jj_1TOm2);
22  pre(ii_1TOm1,jj_1TOm2)=gammam*(ergn(ii_1TOm1,jj_1TOm2)
    -...
23  0.5*(run(ii_1TOm1,jj_1TOm2).^2+rvn(ii_1TOm1,jj_1TOm2)
    .^2)./rhon(ii_1TOm1,jj_1TOm2));
24  mpuin(ii_2TOm1,jj_1TOm2)=1./(rhon(ii_2TOm1,jj_1TOm2)+
    rhon(ii_1TOm1m,jj_1TOm2));
25  uin(ii_2TOm1,jj_1TOm2)=(u(ii_2TOm1,jj_1TOm2).*rhon(
    ii_1TOm1m,jj_1TOm2)+...
26  u(ii_1TOm1m,jj_1TOm2).*rhon(ii_2TOm1,jj_1TOm2)).*mpuin
    (ii_2TOm1,jj_1TOm2);
27  mpin_i(ii_2TOm1,jj_1TOm2)=-(pre(ii_2TOm1,jj_1TOm2).*
    rhon(ii_1TOm1m,jj_1TOm2)+...
28  pre(ii_1TOm1m,jj_1TOm2).*rhon(ii_2TOm1,jj_1TOm2)).*
    mpuin(ii_2TOm1,jj_1TOm2);
29  mpuin(ii_2TOm1,jj_1TOm2)=-((pre(ii_2TOm1,jj_1TOm2).*u(
    ii_2TOm1,jj_1TOm2)).*...
30  rhon(ii_1TOm1m,jj_1TOm2)+(pre(ii_1TOm1m,jj_1TOm2).*u(
    ii_1TOm1m,jj_1TOm2)).*...
31  rhon(ii_2TOm1,jj_1TOm2)).*mpuin(ii_2TOm1,jj_1TOm2);
32  mpvin(ii_1TOm1,jj_2TOm2)=1./(rhon(ii_1TOm1,jj_2TOm2)+
    rhon(ii_1TOm1,jj_1TOm2m));
33  vin(ii_1TOm1,jj_2TOm2)=(v(ii_1TOm1,jj_2TOm2).*rhon(
```

```
    ii_1TOm1,jj_1TOm2m)+...
34  v(ii_1TOm1,jj_1TOm2m).*rhon(ii_1TOm1,jj_2TOm2)).*mpvin
    (ii_1TOm1,jj_2TOm2);
35  mpin_j(ii_1TOm1,jj_2TOm2)=-(pre(ii_1TOm1,jj_2TOm2).*
    rhon(ii_1TOm1,jj_1TOm2m)+...
36  pre(ii_1TOm1,jj_1TOm2m).*rhon(ii_1TOm1,jj_2TOm2)).*
    mpvin(ii_1TOm1,jj_2TOm2);
37  mpvin(ii_1TOm1,jj_2TOm2)=-((pre(ii_1TOm1,jj_2TOm2).*v(
    ii_1TOm1,jj_2TOm2)).*...
38  rhon(ii_1TOm1,jj_1TOm2m)+(pre(ii_1TOm1,jj_1TOm2m).*v(
    ii_1TOm1,jj_1TOm2m)).*...
39  rhon(ii_1TOm1,jj_2TOm2)).*mpvin(ii_1TOm1,jj_2TOm2);
40  uin(i1,jj_1TOm2)=u(i1,jj_1TOm2);
41  uin(m1p,jj_1TOm2)=u(in,jj_1TOm2);
42  mpin_i(i1,jj_1TOm2)=-pre(i1,jj_1TOm2);
43  mpin_i(m1p,jj_1TOm2)=-pre(in,jj_1TOm2);
44  mpuin(i1,jj_1TOm2)=mpin_i(i1,jj_1TOm2).*uin(i1,
    jj_1TOm2);
45  mpuin(m1p,jj_1TOm2)=mpin_i(m1p,jj_1TOm2).*uin(m1p,
    jj_1TOm2);
46  vin(ii_1TOm1,j1)=v(ii_1TOm1,j1);
47  vin(ii_1TOm1,m2p)=v(ii_1TOm1,jn);
48  mpin_j(ii_1TOm1,j1)=-pre(ii_1TOm1,j1);
```

```
49      mpin_j(ii_1TOm1,m2p)=-pre(ii_1TOm1,jn);          mpvin(
          ii_1TOm1,j1)=vin(ii_1TOm1,j1).*mpin_j(ii_1TOm1,j1);
50      mpvin(ii_1TOm1,m2p)=vin(ii_1TOm1,m2p).*mpin_j(ii_1TOm1
          ,m2p);
51
52      [rhon]=FCT2D_trans(rhoo,uin,vin,dtsub);
53
54      FCT2D_source(mpin_i,dtsub,3,unit);
55      [run]=FCT2D_trans(ruo,uin,vin,dtsub);
56
57      FCT2D_source(mpin_j,dtsub,4,unit);
58      [rvn]=FCT2D_trans(rvo,uin,vin,dtsub);
59
60      FCT2D_source(mpuin,dtsub,1,unit);
61      FCT2D_source(mpvin,dtsub,2,unit);
62      [ergn]=FCT2D_trans(ergo,uin,vin,dtsub);
63
64  end
65
66  mpin_i(:,:)=0.0; mpin_j(:,:)=0.0; u(:,:)=0.0; v(:,:)=0.0;
67  rhoo(:,:)=0.0;  ruo(:,:)=0.0; rvo(:,:)=0.0;
68  ergo(:,:)=0.0; uin(:,:)=0.0; vin(:,:)=0.0;
69  pre(:,:)=0.0; mpuin(:,:)=0.0; mpvin(:,:)=0.0;
```

```
70
71 end
```

References

- J. P. Boris, A. M. Landsberg, E. S. Oran, and J. H. Gardner, " LCPFCT—Flux-corrected transport algorithm for solving generalized continuity equations," Technical Report No. NRL/MR/6410-93-7192, Naval Research Laboratory, 1993.
- S. T. Zalesak, " Fully multidimensional flux-corrected transport algorithms for fluids," J. Comput. Phys. 31(3), 335–362 (1979).
- DeVore CR. "An improved limiter for multidimensional flux-corrected transport." NAVAL RESEARCH LAB WASHINGTON DC; 1998 Dec 31.

Thank you