

Computational Fluid Dynamics with Shocks

Matthew Vernacchia

May 16, 2016

1 ABSTRACT

I present a review of the governing equations of fluid flow, and of Computational Fluid Dynamics methods for the numerical solution of these equations. I describe my implementation of a 2D finite difference MacCormack solver for the Euler equations, and demonstrate my solver on 2 problems which contain shock waves: a shock tube and supersonic flow over a ramp.

2 INTRODUCTION

2.1 MOTIVATION

Many engineering problems in the military and transportation sectors involve supersonic fluid flow. A significant phenomenon in many supersonic flow problems is the formation of shock waves. Important design problems, such as a helmets to protect against traumatic brain injury from explosions, rocket nozzles for missiles and launch vehicles, or a spacecraft's reentry trajectory, depend on our ability to model and understand shocks.

Analytical solutions exist for some simple problems, and these were the primary engineering analysis tools during the 1940s, 50s and 60s. However, these tools restricted designers to simple geometries and a piecewise understanding of the problem. Consider an aircraft designer asked to predict the shock pattern and drag of a fighter jet in supersonic flight. He could solve the shock pattern from a conical nose and from a diamond-section wing or tail, and add these results together to get a rough estimate of the total vehicle's drag. However, the interaction of these shocks where the wing and fuselage meet would be left to guesswork. And a more complicated wing cross section (i.e. a proper airfoil) would not be possible to solve. Therefore, a drag prediction of useful fidelity would require expensive and time consuming wind tunnel tests.

Numerical solutions to fluid flow problems (Computational Fluid Dynamics) now enable engineers to analyze more complicated problems. This allows for the design of higher-performance vehicles. Although wind tunnel testing is often still necessary to calibrate or validate CFD results, the use of CFD means that much fewer physical experiments must be performed. This can substantially reduce the cost and time of development.

2.2 GOVERNING EQUATIONS

2.2.1 NAVIER-STOKES EQUATIONS

Fluid flow (whether subsonic or supersonic) is governed by the Navier Stokes equations. In n dimensions, these equations contain $4 + n$ unknowns at each point in space and time:

- Pressure, p
- Density, ρ
- Internal energy, e
- Temperature, T
- Velocity, \vec{u}

There are $4 + n$ equations which relate these variables:

CONTINUITY EQUATION The continuity equation enforces the conservation of mass.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \quad (2.1)$$

MOMENTUM EQUATIONS The momentum equations (one for each spatial dimension) enforce Newton's Laws. They relate the inertia of the fluid to the pressure, viscous, and external forces.

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{u} + \frac{1}{3} \nu \nabla (\nabla \cdot \vec{u}) + \vec{F} \quad (2.2)$$

where ν is the kinematic viscosity (a property of the fluid) and \vec{F} is an external force applied to the fluid per unit mass.

ENERGY EQUATION The energy equation enforces the conservation of energy. It relates the change in internal energy of the fluid to pressure work, viscous work, and thermal conduction.

$$\rho \left(\frac{\partial e}{\partial t} + \vec{u} \cdot \nabla e \right) = -p \nabla \cdot \vec{u} + \Phi + \nabla \cdot (k \nabla T) \quad (2.3)$$

where Φ is a viscous term dependent on the velocity second derivatives of velocity and ν , and k is the thermal conductivity (a property of the fluid).

THERMODYNAMIC RELATION The thermodynamic relation links the internal energy to the temperature and pressure.

$$e = e(T, p) \quad (2.4)$$

For a calorically perfect gas,

$$e = c_v T \quad (2.5)$$

where c_v is the gas's heat capacity at constant volume. Most real gases can be approximated as calorically perfect over small ranges of temperature.

EQUATION OF STATE The equation of state relates the fluid's pressure, density, and temperature.

$$\rho = \rho(p, T) \quad (2.6)$$

A common equation of state is the ideal gas law:

$$\rho = \frac{p}{RT} \quad (2.7)$$

where R is the specific gas constant (a property of the fluid). For a gas that is also calorically perfect, the ideal gas law can be written as:

$$\rho = \frac{p}{(\gamma - 1)e} \quad (2.8)$$

where $\gamma = c_p/c_v$ is the ratio of specific heats (a property of the fluid).

2.2.2 NON-DIMENSIONAL PARAMETERS

Two important non-dimensional parameters define flow regimes where the behavior of the Navier-Stokes equations are different.

MACH NUMBER The Mach number is the local velocity normalized by the local speed of sound.

$$M = \frac{|\vec{u}|}{a} \quad (2.9)$$

where a is the local speed of sound (a function of the temperature and fluid properties). Mach numbers less than 1 are subsonic; while Mach numbers greater than one are supersonic. For subsonic flow, the Navier-Stokes equations are classified as elliptic PDEs, and their solutions are smooth. For supersonic flow, the Navier-Stokes PDEs are parabolic: information does not propagate upstream in the flow, and sharp changes in the solution variables (shocks) can exist.

REYNOLDS NUMBER The Reynolds number gives the relative strength of inertia and viscous forces.

$$Re = \frac{|\vec{u}|L}{\nu} \quad (2.10)$$

Where L is a length scale of the problem, such as the chord of a wing or the diameter of a pipe. If the Reynolds number is high, then viscous forces are small compared to the inertia of the flow, and can be neglected for certain calculations. This leads to an important simplification of the Navier-Stokes equations.

2.2.3 EULER EQUATIONS

In the context of fluid dynamics, the Euler equations are a simplification of the Navier-Stokes equations, with no viscosity or thermal conductivity ($\nu = k = 0$). They provide a good approximation of certain high Reynolds number problems. The modified momentum and energy equations are written below. The continuity equation (2.1), thermodynamic relation (2.4), and equation of state (2.6) did not contain viscosity or conductivity terms, and are therefore unchanged.

EULER MOMENTUM EQUATION

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\frac{1}{\rho} \nabla p + \vec{F} \quad (2.11)$$

EULER ENERGY EQUATION

$$\rho \left(\frac{\partial e}{\partial t} + \vec{u} \cdot \nabla e \right) = -p \nabla \vec{u} \quad (2.12)$$

The Euler equations are hyperbolic PDEs.

2.2.4 EULER EQUATIONS IN CONSERVATIVE FORM

The Euler equations can be written in a conservative, vectorized, flux-based form. This form is convenient for numerical techniques. I will present the conservative form in 2 dimensions, but it can also be developed in 1 or 3. I assume a 2D Cartesian coordinate system with spatial directions x and y . The state of the fluid is represented by the vector U (not to be confused with the velocity vector, \vec{u}):

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix} \quad (2.13)$$

where:

- u and v are the velocity components in the x and y directions. $\vec{u} = [u, v]^T$.
- ρu and ρv are the momentum per unit volume in the x and y directions. They have units of mass per area-time (kgm^{-2}s in SI).

- E is the total energy per unit volume. $E = \rho(e + \frac{1}{2}(u^2 + v^2))$. E has units of energy per volume, which is equal to units of pressure (in SI, J m^{-3} or Pa).

The state of the fluid is a function of space and time:

$$U = U(x, y, t) \quad (2.14)$$

Next, define fluxes F and G for the x and y directions:

$$F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E + p)u \end{bmatrix} \quad (2.15)$$

$$G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E + p)v \end{bmatrix} \quad (2.16)$$

The Euler continuity, energy, and momentum equations can be written as:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0 \quad (2.17)$$

Although pressure and temperature are not included in the state vector U , they can be uniquely determined from U using the equation of state and thermodynamic relation. In the case of a calorically perfect gas,

$$p = (\gamma - 1) \left(E - \frac{1}{2} \rho (u^2 + v^2) \right) \quad (2.18)$$

$$T = \frac{1}{c_v} \left(\frac{E}{\rho} - (u^2 + v^2) \right) \quad (2.19)$$

2.3 NUMERICAL TECHNIQUES

Many numerical techniques exist for solving the Euler equations. I classify these techniques by 3 choices: the spatial flux scheme, the time update scheme, and the shock capturing technique. In all techniques, the solution is discretized spatially onto a grid or mesh. The state U of each grid node is stored, this set of U s represents the state of the system at a point in time. An update function uses this data to calculate the set of U s at the next time step.

2.3.1 SPATIAL FLUX SCHEME

Two major techniques exist for capturing the spatial derivative terms: finite differences and finite volume.

FINITE DIFFERENCES In the Finite Difference technique, the fluxes F and G are calculated at each grid point. To find the time derivative of U at grid point i, j , one uses the flux values at the neighbor points to estimate the spatial gradients of F and G . These gradients can be calculated using a central difference or upwind stencil. Finite difference techniques were the first to be developed, and are simpler to implement.

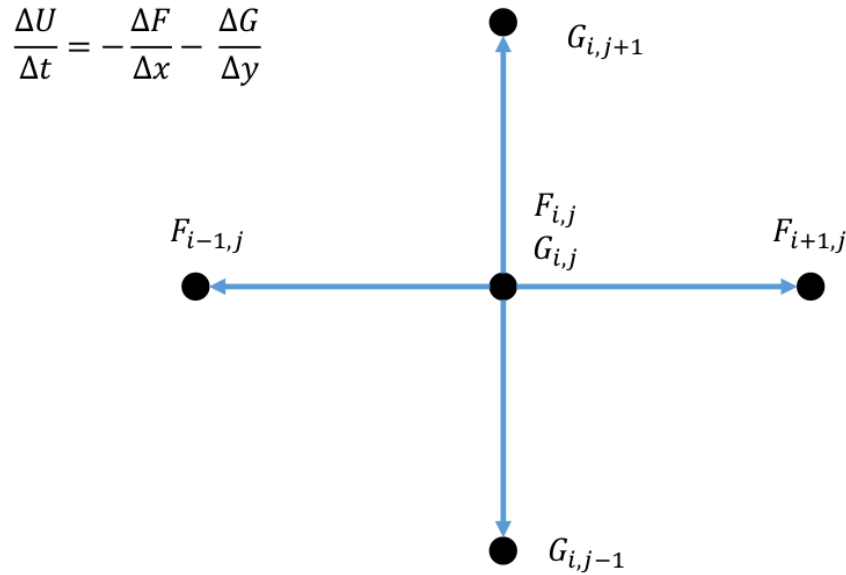


Figure 2.1: Finite Difference scheme.

FINITE VOLUMES In the Finite Volume technique, a boundary B is defined around each grid point, which encloses a finite volume V of space. The fluxes F and G are calculated between grid points, on the sides of B . Using the divergence theorem, we write the time rate of change of U within V as surface integral of the fluxes over B :

$$\frac{\partial}{\partial t} \left(\iint_V U dV \right) = \oint_B (F\hat{x} + G\hat{y}) \cdot \hat{n} dB \quad (2.20)$$

To discretized this equation, we choose V small enough that U can be assumed to have a uniform value within it. Then,

$$\frac{\Delta U}{\Delta t}_{i,j} = \frac{1}{V} \oint_{B_{i,j}} (F\hat{x} + G\hat{y}) \cdot \hat{n} dB \quad (2.21)$$

The finite volume scheme was developed after finite differences, and is somewhat harder to implement (on a Cartesian grid). However, it is conservative, and is easier to apply to unstructured meshes. Guaranteed conservation and applicability to complicated geometries are valuable for solving engineering problems. Therefore, finite volumes are used in most modern CFD software.

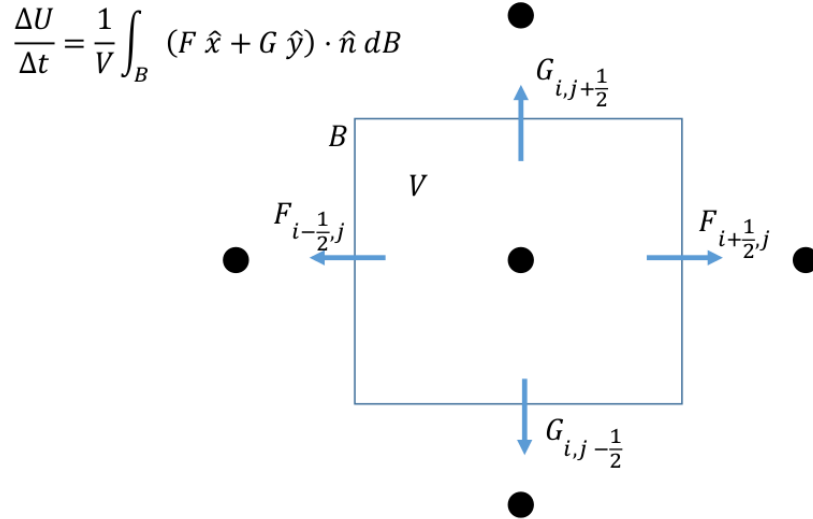


Figure 2.2: Finite Volume scheme.

2.3.2 TIME UPDATE SCHEME

Several algorithms are used to approximate the integration of U through time. These are broadly divided into explicit and implicit techniques.

EXPLICIT Explicit techniques define U^{n+1} as an explicit function of previous time steps U^n, U^{n-1}, \dots . They are usually easier to implement and require less computation per step than implicit techniques. However, their stability is limited to sufficiently small time steps. The Courant-Friedrichs-Lewy (CFL) condition provides an upper bound for the allowable time step:

$$\Delta t < (\Delta t)_{CFL} = \frac{1}{\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + a \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}}} \quad (2.22)$$

MacCormack is a common explicit method for the Euler and Navier-Stokes equations, but other explicit methods (Runge-Kutta, Lax-Wendroff) have been successfully used as well.

IMPLICIT Implicit techniques define an implicit relation between U^{n+1} and U^n . To find U^{n+1} , a (usually linear) system of equations must be solved. Implicit techniques are more difficult to implement and require more computation at each step (to solve the system of equations). However, their stability is not restricted to small time steps, so (if high resolution in time is not desired) less steps need to be computed, reducing the overall computation requirements.

Beam-Warming is a commonly used implicit method.

2.3.3 SHOCK CAPTURING

Hyperbolic PDEs present a challenge to numerical algorithms. In the case of the Euler equations, solvers tend to exhibit non-physical oscillations near shocks. These oscillations can impart the stability of the algorithm, and reduce the accuracy of the solution. To properly capture flow with shocks, modifications must be made to the algorithm to reduce these oscillations. This is achieved by adding artificial dissipation or limiting the flux.

CLASSICAL SCHEMES Classical schemes for shock handling use a classical time and gradient scheme (e.g. MacCormack) and add artificial dissipation.

MODERN SCHEMES A popular class of modern shock handling schemes are the Total Variation Diminishing (TVD) methods. These have the property the the magnitude of the local extrema of the solution monotonically decrease with time. They handle shocks well, but only provide 1st order spatial accuracy. Newer algorithms include the Monotonic Upstream-centered Schemes for Conservation Laws (MUSCL) and Essentially Non-Oscillatory (ENO) methods.

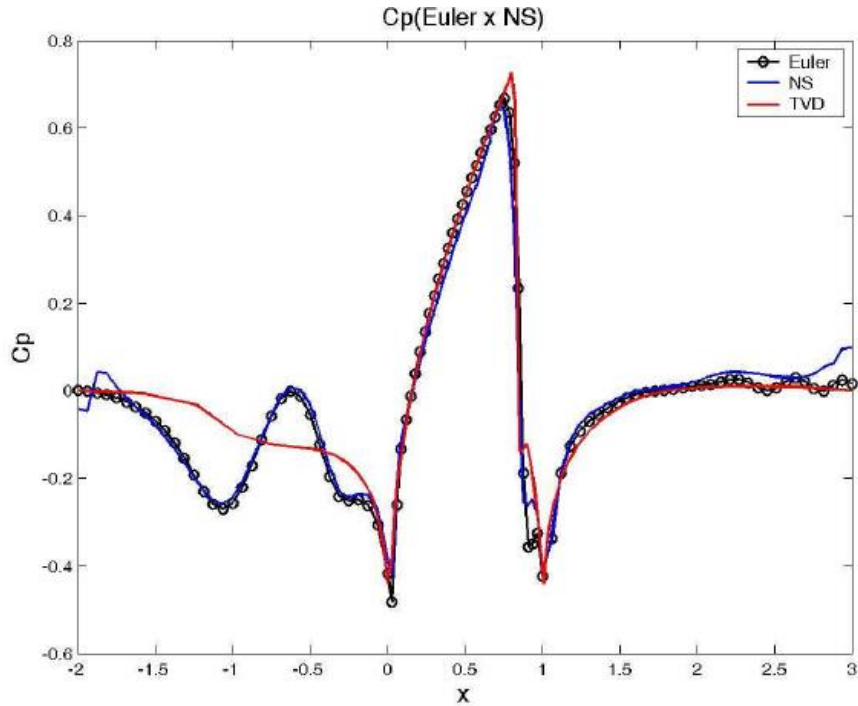


Figure 2.3: A comparison of shock capturing schemes. This figure shows the coefficient of pressure along the surface of a wall. The wall has a small parabolic bump that protrudes into the flow from $x = 0$ to $x = 1$. The inlet flow is subsonic ($M = 0.85$) but the fluid accelerates to sonic speed over the bump, and a shock exists at the rear of the bump. A MacCormack solver with artificial dissipation (black curve) gives non-physical pressure oscillations upstream of the bump. A TVD solver (red curve) does not. Figure reprinted from D. Lobao 2010.

3 IMPLEMENTATION

3.1 COMPUTATION PLATFORM

I implemented my solver in the Julia programming language (v 0.4.2). I performed my computations on a laptop computer with 6 GB RAM and an Intel i5-2410M CPU (2 cores at 2.3 GHz) running 64 bit Ubuntu 14.04.

My code is available at <https://github.mit.edu/mvernacc/18086-project>.

3.2 SOLVER

I chose a finite difference MacCormack scheme because of its implementation simplicity.

3.2.1 TIME UPDATE: MACCORMACK SCHEME

The MacCormack scheme operates in 2 steps: a predictor and a corrector.

$$U_{i,j}^* = U_{i,j}^n - \frac{\Delta t}{\Delta x}(F_{i+1,j}^n - F_{i,j}^n) - \frac{\Delta t}{\Delta y}(G_{i,j+1}^n - G_{i,j}^n) \quad (3.1)$$

$$U_{i,j}^{n+1} = \frac{1}{2}(U_{i,j}^n + U_{i,j}^*) - \frac{\Delta t}{2\Delta x}(F_{i,j}^* - F_{i-1,j}^*) - \frac{\Delta t}{2\Delta y}(G_{i,j}^* - G_{i,j-1}^*) \quad (3.2)$$

where F^* is the flux calculated using U^* .

3.2.2 SHOCK CAPTURING: ARTIFICIAL DISSIPATION

To reduce oscillations around shocks, I use the following artificial dissipation scheme, taken from D. Lobao 2010:

$$F_{i,j}^{ad} = \frac{1}{2}(d_{i+1,j} - d_{i-1,j}) \quad (3.3)$$

$$d_{i,j} = \epsilon_{i,j}^{(2)}(U_{i,j} - U_{i-1,j}) - \epsilon_{i,j}^{(4)}\Phi_{i,j} \quad (3.4)$$

$$\Phi_{i,j} = U_{i+1,j} - 3U_{i,j} + 3U_{i-1,j} - U_{i-2,j} \quad (3.5)$$

$$v_{i,j} = \frac{|p_{i+1,j} - 2p_{i,j} + p_{i-1,j}|}{p_{i+1,j} + 2p_{i,j} + p_{i-1,j}} \quad (3.6)$$

$$\epsilon_{i,j}^{(2)} = k^{(2)}v_{i-1,j}(|\vec{u}| + a)_{i,j} \quad (3.7)$$

$$\epsilon_{i,j}^{(4)} = \max(0, k^{(4)} - \epsilon_{i,j}^{(2)})(|\vec{u}| + a)_{i,j} \quad (3.8)$$

The artificial dissipation in the y direction is analogous. $k^{(2)}$ and $k^{(4)}$ are tuneable parameters. Lobao recommends values of $k^{(2)} = 1/4$ and $k^{(4)} = 1/256$. I found $k^{(2)} = 1/4$ and $k^{(4)} = 1/32$ to be more suitable.

While the MacCormack scheme only depends on neighboring cells, the artificial dissipation terms depend on cells up to 3 away. This wider spread introduces diffusion, which reduces the magnitude of local extrema.

3.2.3 CURVILINEAR COORDINATES

Curvilinear coordinates are required to handle domains which are not made up of rectangular segments. Consider a coordinate transform

$$(x, y) \rightarrow (\xi(x, y), \eta(x, y)) \quad (3.9)$$

where (x, y) are physical coordinates and (ξ, η) are grid coordinates, i.e. $i = \xi/\Delta\xi$, $j = \eta/\Delta\eta$.

We can write the Euler conservation equation in terms of the transformed state and flux:

$$J = \frac{1}{x_\xi y_\eta - x_\eta y_\xi} \quad (3.10)$$

$$U'' = \frac{U}{J} \quad (3.11)$$

$$F'' = y_\eta F = x_\eta G \quad (3.12)$$

$$G'' = -y_\xi F + x_\xi G \quad (3.13)$$

$$\frac{\partial U''}{\partial t} + \frac{\partial F''}{\partial \xi} + \frac{\partial G''}{\partial \eta} = 0 \quad (3.14)$$

3.2.4 BOUNDARY CONDITIONS

Boundary conditions are implemented by padding the U grid with ghost cells.

At walls, the ghost cells have the same ρ and E as their internal siblings, but the velocity vector is reflected about the wall tangent. This enforces a non-penetration condition at the wall. Non-penetration is the appropriate velocity wall condition for the Euler equations; for the full Navier-Stokes equations a non-slip condition would also be required.

At the inlet, the full state U is specified.

The outlet is free, i.e. the ghost cell is equal to the internal sibling cell.

4 RESULTS

4.1 SHOCK TUBE

I first demonstrate my solver on a shock tube to illustrate the necessity of artificial dissipation. A shock tube contains a high and low pressure region separated by a diaphragm. At $t = 0$, the diaphragm bursts, leaving a pressure discontinuity in the tube. A shock wave travels from the discontinuity into the low pressure region, and an expansion fan travels into the high pressure region. A region of uniform, intermediate pressure is left between the shock and the expansion fan.

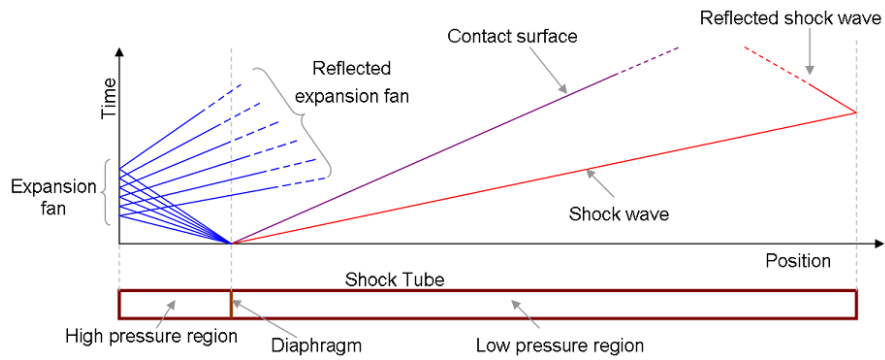


Figure 4.1: A shock tube. Reprinted from Wikipedia.

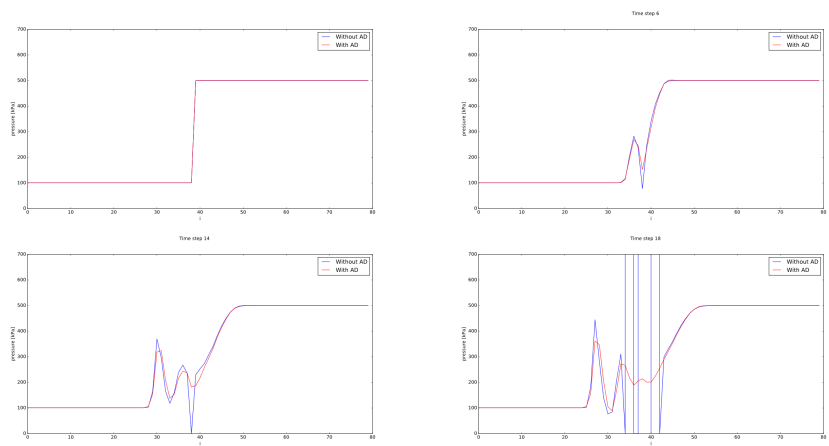


Figure 4.2: The evolution of pressure in the shock tube. Compare the (unstable) solution without artificial dissipation (blue) to the solution with artificial dissipation (red)

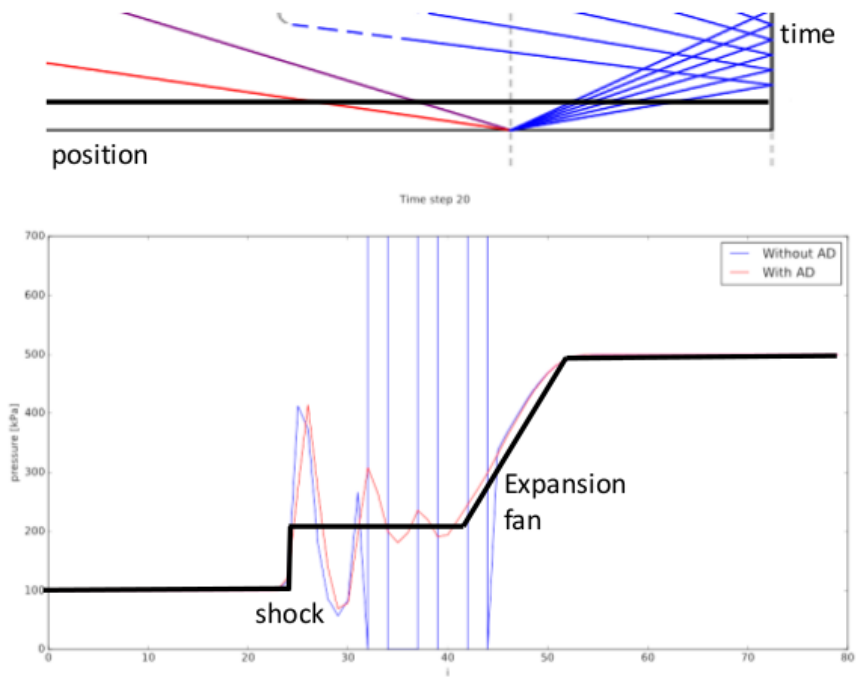


Figure 4.3: My simulated shock tube results compared to the theoretical solution. Although the artificial dissipation algorithm is stable, it shows non-physical oscillations between the shock and the expansion fan.

4.2 MACH 1.5 RAMP

Next, I simulate Mach 1.5 flow in a tunnel with a 10° ramp. This example is applicable to the design of intake ramps for supersonic air-breathing propulsion systems (see Figure 4.4). It illustrates several important features of supersonic flow: a shock, an expansion fan, and shock reflections. Figure 4.5 shows the expected locations of these features, and the boundary conditions of the problem.



Figure 4.4: The inlet ramps of a McDonnell-Douglas F-15E Strike Eagle. Reprinted from Wikipedia.

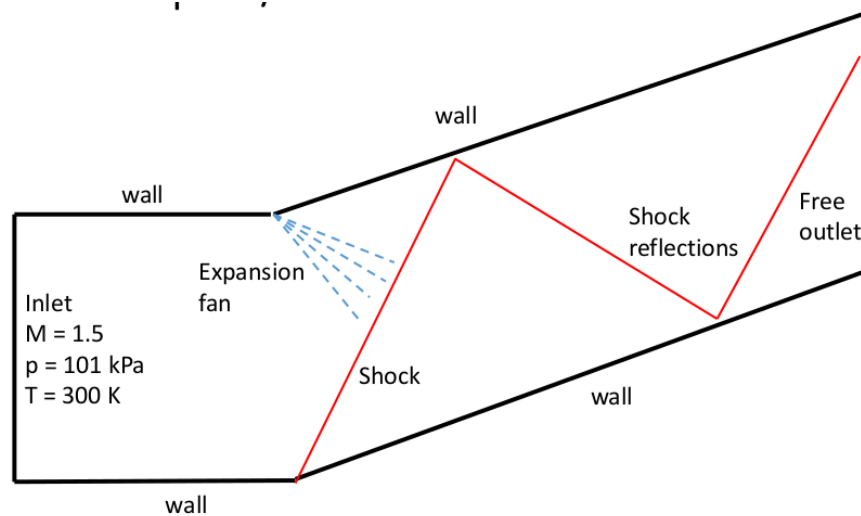


Figure 4.5: The boundary conditions and expected features of the ramp problem.

I started the problem from a uniform initial condition equal to the inlet, and ran 2,200 iterations of the MacCormack solver with artificial dissipation. I used a 120 by 40 grid, and a

time step equal to $0.4(\Delta t)_{CFL}$. The simulation ran in 51 min.

The evolution of the solution is shown in Figure 4.6. First, a low density region forms at the top wall as the fluid diverges away from the wall. A high density region forms at the lower wall as fluid impacts the wall. These regions are due to the non-physical initial conditions, which set the velocity to be horizontal everywhere, violating the non-penetration condition at the tilted walls.

Next, we see the growth of a shockwave from the bottom corner and an expansion fan from the top corner. The density increases across the shock and decreases across the fan. As the shock continues to grow, it reflects off of the top wall.

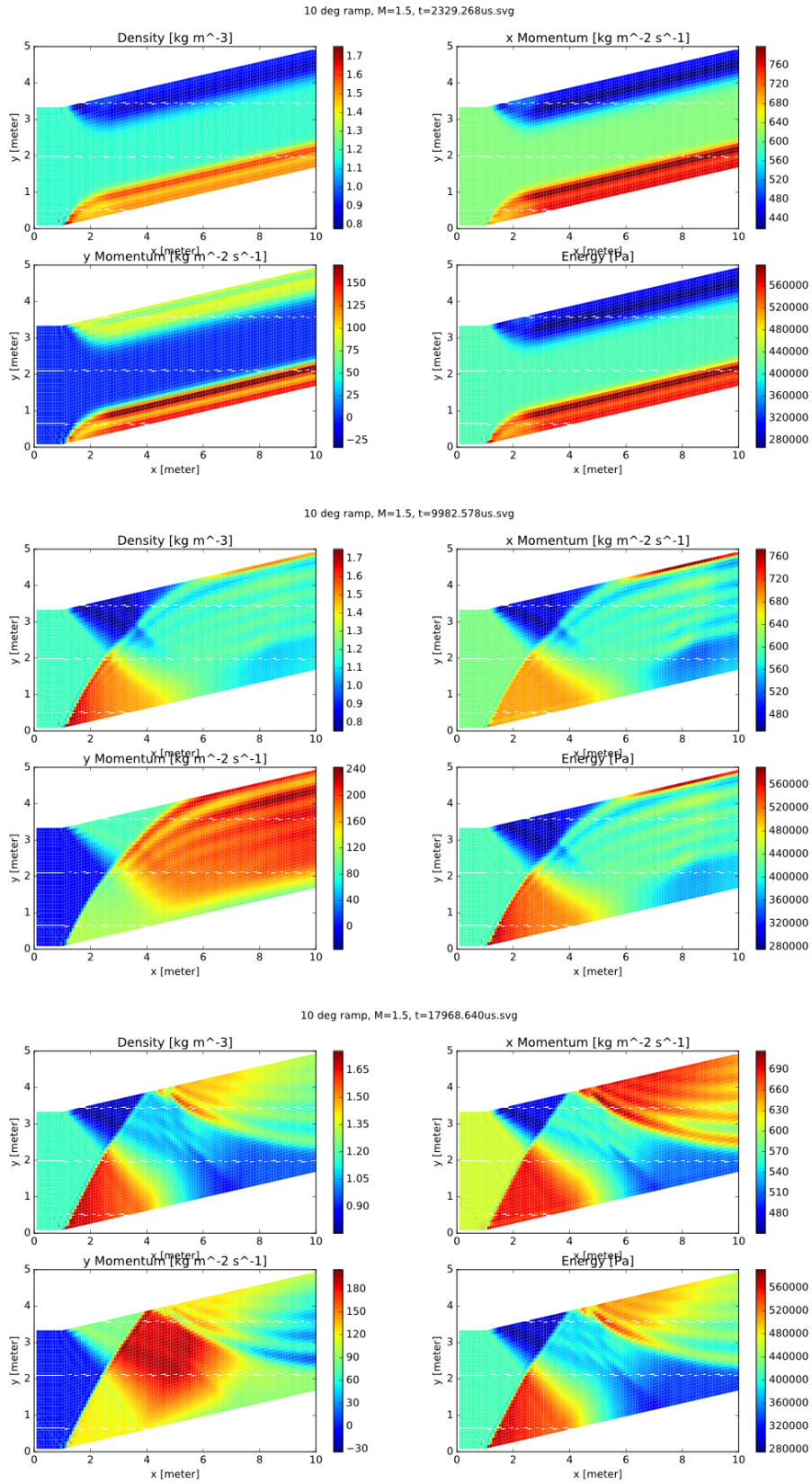


Figure 4.6: The evolution of the ramp problem solution.

Figure 4.7 shows the steady-state of the ramp problem solution. All of the expected features are present. The shocks are crisp, only 2 or 3 mesh points wide. However, non-physical oscillations are also present after the first and second shock reflections. These could be reduced by increases the artificial dissipation, but at the cost of blurring the shocks. To eliminated the oscillations, a more advanced solver, such as TVD, MUSCL or ENO must be used in place of MacCormack.

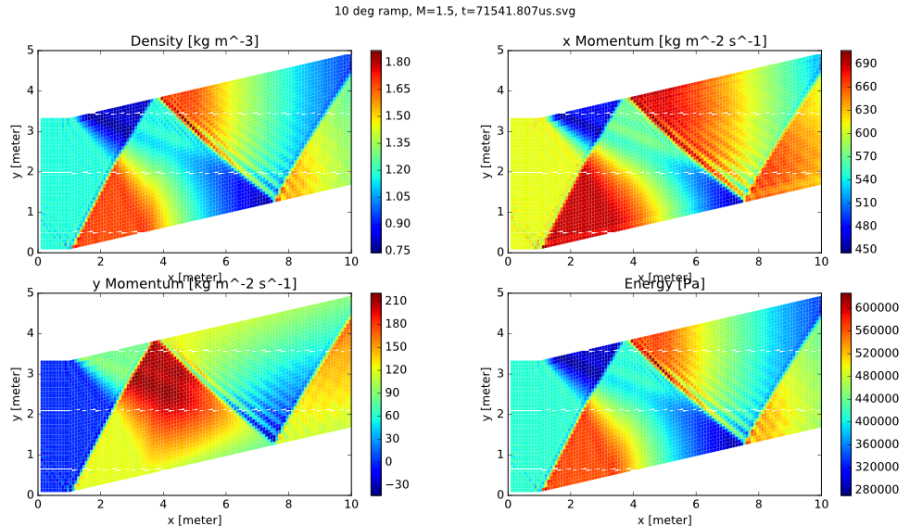


Figure 4.7: The steady-state ramp problem solution.

5 CONCLUSION

I have implemented a 2D finite difference MacCormack algorithm to solve the compressible Euler equations. The solver uses curvilinear coordinates and contains artificial dissipation terms. The solver is able to stably resolve shocks, both due to pressure discontinuities and due to supersonic flow. However, as is typical of MacCormack solvers, noticeable non-physical oscillations are present near shocks, even with artificial dissipation. I applied the solver to the simulation of a shock tube and of supersonic flow over a ramp.