Convergence of Fixed Geometry Nozzles Using the Pseudo-1D Euler Equations

Alex Le & Christopher Uyeda

March 13, 2015

1 Introduction

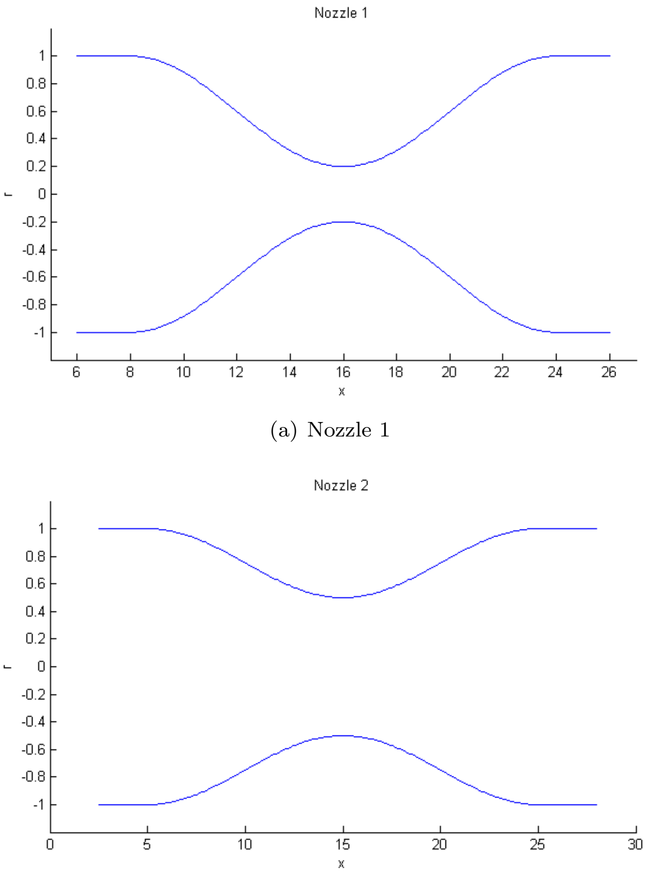
The pseudo-1D Euler equations will be explored for the case of an unsteady nozzle problem with fixed geometry. The simulations of the nozzle will be run until the steady state is reached for various pressure ratios that will result in purely subsonic flow, choked subsonic flow, and purely supersonic flow. In each case, the steady state results will be compared to the theoretical shock locations, if they exist, for different nozzles. Because the pseudo-1D Euler equations inherently contain a source terms, fractional splitting methods will be used to account for the source term. The fractional splitting method involves advecting using the Roe approximate Riemann solver, and then updating to the next time step by solving an ODE to account for the source terms.

2 Physical Model

The simulation will be modeling flow through a converging-diverging nozzle with two different geometry nozzles given by

$$\begin{aligned} r = 1 - 0.4(1 + cos(\pi(x-16)/8)) \\ \& \\ r = 0.75 - 0.25\sin(\pi(x-10)/10) \\ A = \pi r^2 \end{aligned} \tag{2.1}$$

which are shown in Fig. 1 where Nozzle 1 has $\frac{A}{A_t} = 25$ and Nozzle 2 has $\frac{A}{A_t} = 4$



(a) Nozzle 1

(b) Nozzle 2

Figure 1: Two different nozzle geometries used in the simulations.

There can be three different types of flow that exist in this nozzle that are dependent on the pressures at the inlet, opened to a pressure reservoir, and the outlet, opened to atmosphere. It is best to define these flow types by the Mach number

$$M = \frac{u}{c} \tag{2.2}$$

where u is the velocity and c is the speed of sound. The first type of flow results when the pressure ratio, p_{out}/p_{in}, is higher than a certain threshold such that the flow is purely subsonic flow, $M < 1$. Flow accelerates through the converging part, but never reaches a sonic speed, or $M = 1$, at the throat of the nozzle. This forces the flow to decelerate through the remaining diverging part of the nozzle such that the flow in the nozzle is entirely subsonic, $M < 1$. The second type of flow is supersonic when the pressure ratio is lower than a certain threshold, where flow accelerates through the converging section and reaches $M = 1$ at the throat. Physics dictates that the flow must now accelerate in the remaining diverging part of the nozzle which results in $M > 1$ at the outlet of the nozzle. These two conditions are a result of the pressure ratios being above a threshold for pure subsonic and below a threshold for supersonic. If the pressure ratio lies in between these thresholds, a shock will form. This type of flow will be considered the choked subsonic flow where the flow reaches sonic at the throat, but a shock appears in the diverging section of the nozzle. The shock exists to raise the pressure of the flow such that it equals the atmospheric pressure at the exit.

3 Pseudo-1D Euler Equations

The governing pseudo-1D Euler equation for the simulations is defined by

$$\frac{d}{dt}\begin{bmatrix} \rho A \\ \rho u A \\ eA \end{bmatrix} + \frac{d}{dx}\begin{bmatrix} \rho u A \\ (\rho u^2 + p)A \\ u(e+p)A \end{bmatrix} = \begin{bmatrix} 0 \\ p\frac{dA}{dx} \\ 0 \end{bmatrix} \tag{3.1}$$

where the area can be pulled out by

$$\begin{aligned} Aq_t + f(Aq)_x &= S \\ Aq_t + \frac{dA}{dx}f(q) + Af(q)_x &= S \\ q_t + f(q)_x &= \frac{1}{A}S - \frac{1}{A}\frac{dA}{dx}f(q) \end{aligned}$$

where S is the original source term vector. The reformulated equation then becomes

$$\frac{d}{dt}\begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix} + \frac{d}{dx}\begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(e+p) \end{bmatrix} = \begin{bmatrix} -\frac{1}{A}\frac{dA}{dx}\rho u \\ -\frac{1}{A}\frac{dA}{dx}\rho u^2 \\ -\frac{1}{A}\frac{dA}{dx}u(e+p) \end{bmatrix} \tag{3.2}$$

where the new conserved variables are $q = [\rho, \rho u, e]^T$, similar to the 1D Euler equations, which makes implementation with existing Riemann solvers much easier.

4 Boundary Conditions

In order to initiate flow through the nozzle, boundary conditions must be used at the inlet and outlet. The proper implementation of these conditions are crucial to avoid artificial reflections at the inlet or outlet. Both boundary conditions are purely driven by defined pressures.

4.1 Subsonic Inlet

In the case of a subsonic inlet, the 1-wave is traveling out of the domain while the 2-wave and 3-wave are moving into the domain. This means that two primitive variables must be defined from the boundary, u_0 and p_0 in the first ghost cell. Because pressure will be defined at the inlet from the reservoir, the corresponding u_0 can be determined by the Hugoniot-Loci for the 3-wave that links q_r to q_0, where q_0 are the conserved variables in the first ghost cell. The Rankine-Hugoniot conditions are applied in this case to derive an equation for u_0 in terms of p_0 and γ [1]

$$u_0 = u_1 - \frac{2}{\sqrt{2\gamma(\gamma-1)}}\sqrt{\frac{\gamma p_1}{\rho_1}}\left(\frac{1 - p_0/p_1}{\sqrt{1 + \beta p_0/p_1}}\right) \tag{4.1}$$

where $\beta = (\gamma+1)/(\gamma-1)$ and the subscript 1 refers to the first interior cell. The corresponding density is then defined as

$$\rho_0 = \left(\frac{1 + \beta p_0/p_1}{p_0/p_1 + \beta}\right)\rho_1 \tag{4.2}$$

The energy is defined by

$$e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2 \tag{4.3}$$

so the resulting conserved variables at the first ghost cell is

$$q_0 = \begin{bmatrix} \rho_0 \\ \rho_0 u_0 \\ \frac{p_0}{\gamma-1} + \frac{1}{2}\rho_0 u_0^2 \end{bmatrix} \tag{4.4}$$

where $q_j(i)$ refers to the ith component of q at the jth cell.

4.2 Subsonic Outlet

The subsonic outlet is similar to the subsonic inlet, but now the 1 wave is traveling into the domain while the 2-wave and 3-wave are moving out of the domain. In this case, only one primitive variable must be defined from the boundary, p_{atm}, but in order to satisfy the Riemann condition for the Roe solver, density and momentum are also defined along the 1-wave. The corresponding velocity is

$$u_{N+1} = u_N + \frac{2}{\sqrt{2\gamma(\gamma-1)}}\sqrt{\frac{\gamma p_N}{\rho_N}}\left(\frac{1 - p_{atm}/p_N}{\sqrt{1 + \beta p_{atm}/p_N}}\right) \tag{4.5}$$

where N refers to the last cell in the interior and N + 1 is the first ghost cell. The corresponding density on the 1-wave is

$$\rho_{N+1} = \left(\frac{1 + \beta p_{atm}/p_N}{p_{atm}/p_N + \beta}\right)\rho_N \tag{4.6}$$

Thus the resulting conserved variables defined at the first ghost cell at the outlet, q_n, is

$$q_n = \begin{bmatrix} \rho_{N+1} \\ \rho_{N+1}u_{N+1} \\ \frac{p_{atm}}{\gamma-1} + \frac{1}{2}\rho_{N+1}u_{N+1}^2 \end{bmatrix} \tag{4.7}$$

4.3 Supersonic Outlet

For the supersonic outlet, all the waves are now traveling out of the domain. This means the conserved variables at the first ghost cell at the outlet, q_n, is easily defined by an extrapolation

$$q_n = q_{n-1} \tag{4.8}$$

5 Numerical Method

5.1 Riemann Problem - Euler Equations

To solve the Euler equation Riemann problem, the waves are split into the left going and right going waves according to the eigenvalues. This means that if the flow is subsonic then λ^1 is a left going wave and λ^2 and λ^3 are right going waves, while if the flow is supersonic then all waves are right going waves. The flux Jacobian of the system is given by

$$f'(q) = \begin{bmatrix} 0 & 1 & 0 \\ p_\rho - u^2 & u(2-p_e) & p_e \\ u(p_\rho - H) & H - u^2 p_e & u(1+p_e) \end{bmatrix} \tag{5.1}$$

where $H = \frac{e+p}{\rho}$, $p_\rho = \frac{\gamma-1}{2}u^2$, and $p_e = \gamma - 1$. The eigenvalues may be solved by taking the determinant and setting it equal to zero

$$det\begin{bmatrix} -\lambda & 1 & 0 \\ p_\rho - u^2 & u(2-p_e) - \lambda & p_e \\ u(p_\rho - H) & H - u^2 p_e & u(1+p_e) - \lambda \end{bmatrix} = 0 \tag{5.2}$$

where the eigenvalues are then given by

$$\lambda^1 = u - c, \quad \lambda^2 = u, \quad \lambda^3 = u + c \tag{5.3}$$

Solving for the eigenvectors, r, results in

$$(f'(q) - \lambda I)r = 0$$
$$R = \left[\ r^1 \mid r^2 \mid r^3\ \right] = \begin{bmatrix} 1 & 1 & 1 \\ u-c & u & u+c \\ H-uc & H-\frac{c^2}{\gamma-1} & H+uc \end{bmatrix} \tag{5.4}$$

The coefficients of these eigenvectors, α, may be determined by the following which involves taking the inverse of the eigenvector matrix, R.

$$q_r - q_l = R\alpha \rightarrow R^{-1}(q_r - q_l) = \alpha \tag{5.5}$$

where

$$R^{-1} = \begin{bmatrix} \frac{c^2+H(\gamma-1)+cu+(\gamma-1)u^2}{2c^2} & -\frac{c+(\gamma-1)u}{2c^2} & \frac{\gamma-1}{2c^2} \\ \frac{(\gamma-1)(H-u^2)}{c^2} & \frac{(\gamma-1)u}{c^2} & -\frac{\gamma-1}{c^2} \\ \frac{c^2-H(\gamma-1)-cu+(\gamma-1)u^2}{2c^2} & \frac{c-(\gamma-1)u}{2c^2} & \frac{\gamma-1}{2c^2} \end{bmatrix} \tag{5.6}$$

The coefficients may then be solved to be

$$\begin{aligned} \alpha = R^{-1}(qr-ql) = R^{-1}\delta \\ = \begin{bmatrix} \frac{c^2+H(\gamma-1)+cu+(\gamma-1)u^2}{2c^2}\delta^1 - \frac{c+(\gamma-1)u}{2c^2}\delta^2 + \frac{\gamma-1}{2c^2}\delta^3 \\ \frac{(\gamma-1)(H-u^2)}{c^2}\delta^1 + \frac{(\gamma-1)u}{c^2}\delta^2 - \frac{\gamma-1}{c^2}\delta^3 \\ \frac{c^2-H(\gamma-1)-cu+(\gamma-1)u^2}{2c^2}\delta^1 + \frac{c-(\gamma-1)u}{2c^2}\delta^2 + \frac{\gamma-1}{2c^2}\delta^3 \end{bmatrix} \\ = \begin{bmatrix} (\gamma-1)\frac{(H+u^2)\delta^1-u\delta^2+\delta^3}{2c^2} + \frac{(c^2+cu)\delta^1-c\delta^2}{2c^2} \\ (\gamma-1)\frac{(H-u^2)\delta^1+u\delta^2-\delta^3}{c^2} \\ (\gamma-1)\frac{(u^2-H)\delta^1-u\delta^2+\delta^3}{2c^2} + \frac{(c^2-cu)\delta^1+c\delta^2}{2c^2} \end{bmatrix} \end{aligned} \tag{5.7}$$

where δ^i refers to the ith element of the $\delta = q_r - q_l$ vector. Because solving the exact Riemann solution is very expensive, the Roe approximation is used instead which estimates velocity, $\hat{u}$, total specific enthalpy, $\hat{H}$, and sound speed, $\hat{c}$ [1], by

$$\begin{aligned} \hat{u} &= \frac{\sqrt{\rho_{i-1}}u_{i-1} + \sqrt{\rho_i}u_i}{\sqrt{\rho_{i-1}} + \sqrt{\rho_i}} \\ \hat{H} = \frac{\sqrt{\rho_{i-1}}H_{i-1} + \sqrt{\rho_i}H_i}{\sqrt{\rho_{i-1}} + \sqrt{\rho_i}} &= \frac{(E_{i-1}+p_{i-1})/\sqrt{\rho_{i-1}} + (E_i+p_i)/\sqrt{\rho_i}}{\sqrt{\rho_{i-1}} + \sqrt{\rho_i}} \\ \hat{c} &= \sqrt{(\gamma-1)\left(\hat{H} - \frac{1}{2}\hat{u}^2\right)} \end{aligned} \tag{5.8}$$

These values are then plugged into Eqn. 5.3 and 5.7 to solve the Riemann problem with the approximated values.

5.2 Fractional Splitting

Because the psuedo-1D Euler equation given by Eqn. 3.1 contains a source term, $p\frac{dA}{dx}$, the typical Riemann solving method can't be applied. In order to account for this source term, a fractional splitting method is incorporated, which is chosen to be the Godunov splitting.

Starting with Gudonov splitting, at each timestep, the solution will be advected and then reacted. This two step method takes the form of

$$\begin{aligned} \text{Step 1: } & q_t + \bar{u}q_x = 0 \\ \text{Step 2: } & q_t = \psi(q) \end{aligned} \tag{5.9}$$

The first step will be solved first using the MC limiter method and the second step will be solved by using a two stage Runge-Kutta method to solve the ODE. The two steps when specified to the psuedo-1D Euler equations is given by

$$\begin{aligned} \text{Step 1: } & \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix}_t + \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(e+p) \end{bmatrix}_x = 0 \\ \text{Step 2: } & \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix}_t = \begin{bmatrix} -\frac{1}{A}\frac{dA}{dx}\rho u \\ -\frac{1}{A}\frac{dA}{dx}\rho u^2 \\ -\frac{1}{A}\frac{dA}{dx}(e+p) \end{bmatrix} \end{aligned} \tag{5.10}$$

5.3 Flux Limiter Methods

The first step in the split method is implemented as a modification to an existing Riemann solver within Clawpack, called 'rp1_euler_with_efix.f90'. This is a 1D Euler solver has the Roe-averaged quantities and wave solutions already programmed into it. This solver can be used with limiter methods specified by the 'setrun.py' file.

In general, the flux-limiter method is of the form:

$$Q_i^{n+1} = Q_i^n - \nu(Q_i^n - Q_{i-1}^n) - \frac{1}{2}\nu(1-\nu)[\phi(\theta_{i+1/2}^n)(Q_{i+1}^n - Q_i^n) - \phi(\theta_{i-1/2}^n)(Q_i^n - Q_{i-1}^n)] \tag{5.11}$$

If $\bar{u} > 0$, where $\nu = \frac{\bar{u}\Delta t}{\Delta x}$, and $\theta_{i-1/2}^n = \frac{\Delta Q_{i-3/2}^n}{\Delta Q_{i-1/2}^n}$.

The following function of θ was implemented for its corresponding limiter function.

$$\text{MC: } \phi(\theta) = max(0, min(1+\theta)/2, 2, 2\theta) \tag{5.12}$$

5.4 Source Terms

The second step implements an optional source function that uses a Runge-Kutta method to solve a set of three ODEs. The Runge-Kutta solver takes the form:

$$\begin{aligned} Q_i^{**} &= Q_i^* + \frac{\Delta t}{2}\psi(Q_i^*) \\ Q_i^{n+1} &= Q_i^* + \Delta t \psi(Q_i^{**}) \end{aligned} \tag{5.13}$$

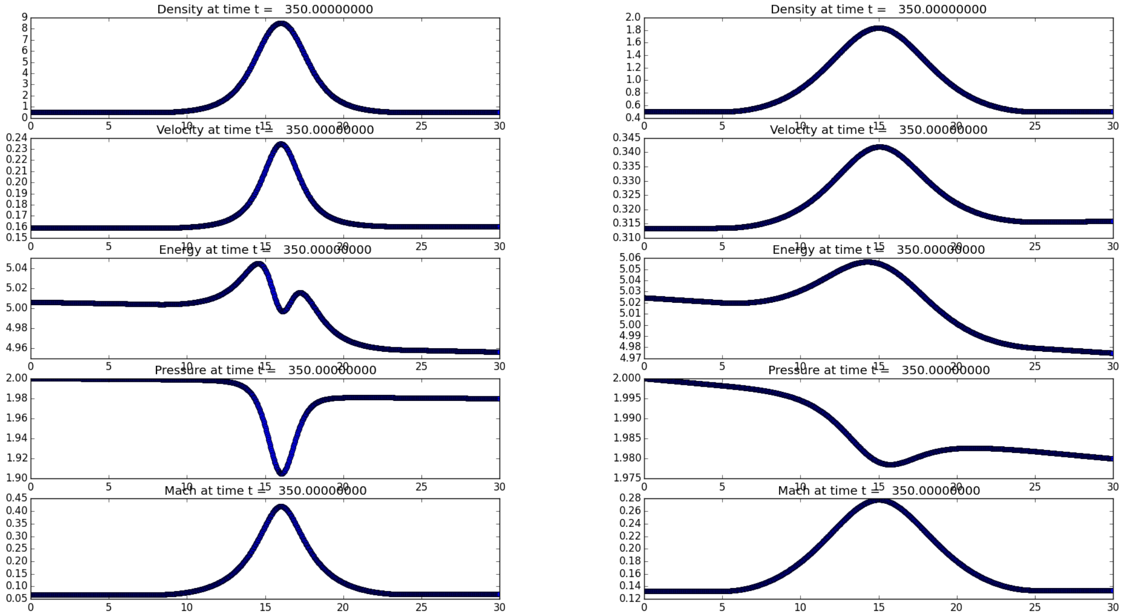
Here, Q^* is the conserved quantity vector after advection, and $\psi(Q^*)$ are the source terms from Eq. 5.10 evaluated using the quantities from the vector Q^*. This solver uses a forward Euler algorithm by approximating the conserved quantity vector at an intermediate state using a half-time step, and then using this midpoint state Q^{**} in order to update the solution to the next time level using a full time step, with $\psi(Q^{**})$ referring to the source terms evaluated using Q^{**}.

6 Numerical Results

All simulations were run with a 1000 mesh over a domain size of 0 to 30 to ensure that the cell spacing is consistent in each simulation and used the MC limiter as previously discussed. The following are the results for each simulation.

6.1 Subsonic

The results for nozzle 1 and 2 are shown at the steady state for pressure ratio $\frac{p_{atm}}{p_o} = 0.99$ in Fig. 2. It can be seen that the pressure ratio at the two boundaries are satisfied while Mach is always below 1 throughout the entire nozzle. As expected, the Mach and flow velocity increased through the converging section and decreased through the diverging section which agrees with physics. Also, because nozzle 1 has a higher area ratio, the flow velocity at the throat is expected to have accelerated faster than that of nozzle 2 which can be observed in the results.



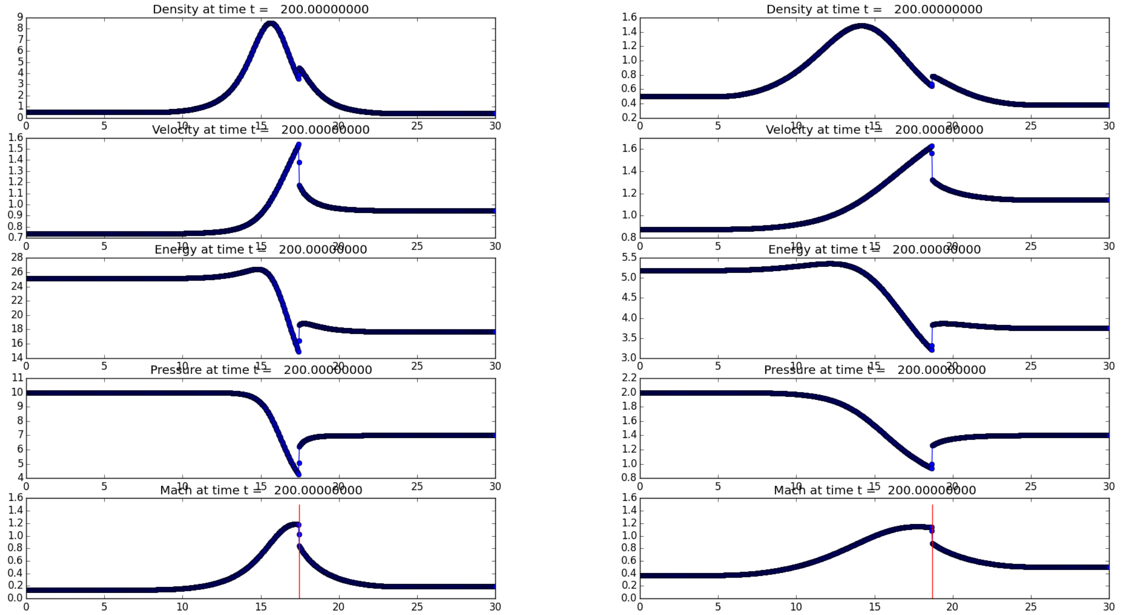
(a) Nozzle 1

(b) Nozzle 2

Figure 2: a). Nozzle 1 and b). Nozzle 2 converged at steady state for a high pressure ratio which results in subsonic flow.

6.2 Choked Subsonic

The results for nozzle 1 and 2 are shown in Fig. 3 at the steady state for pressure ratio, $\frac{p_{atm}}{p_o} = 0.7$. The theoretical shock location is calculated and plotted in the Mach plot of the results to compare how well the simulations were able to capture the shock.



(a) Nozzle 1

(b) Nozzle 2

Figure 3: a). Nozzle 1 and b). Nozzle 2 converged at steady state for a high pressure ratio which results in choked subsonic flow.

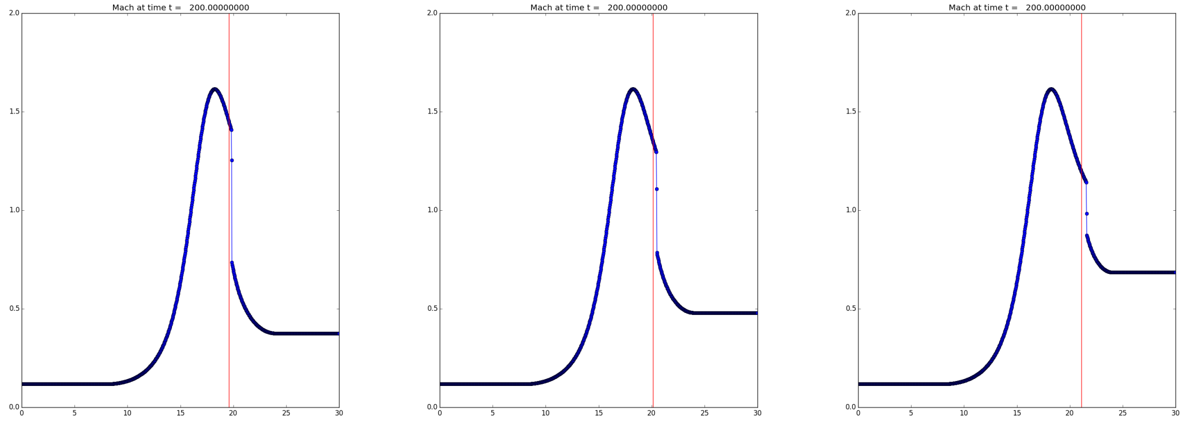
6.3 Supersonic

Unfortunately, the supersonic cases converged to a physically incorrect flow in the nozzle. After the flow becomes sonic at the throat, the flow should continue to accelerate through the diverging section of the nozzle until the constant area section. The simulation shows that the flow instead decreases after the throat and then increases all in the diverging section of the nozzle until the constant area section.

6.4 Convergence Issues

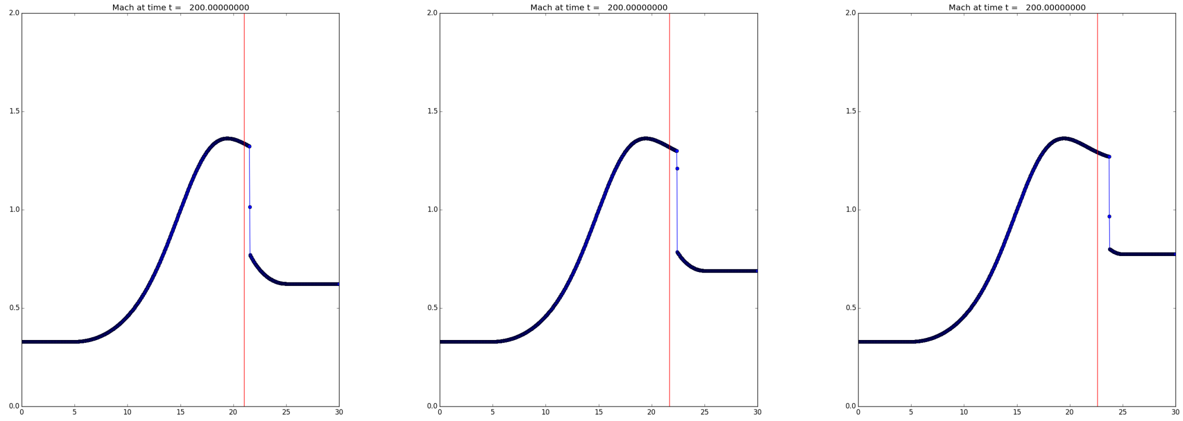
Simulations of the two geometry nozzles were able to converge for most cases that had pure subsonic and choked subsonic flows. Special care had to be taken when defining the initial reservoir pressure for each nozzle. It was discovered that a higher reservoir pressure needed to be defined for the higher area ratio nozzle 1 than the other nozzle 2 in order for nozzle 1 to converge properly at lower pressure ratios. For both nozzles, the reservoir pressure needed to be lowered in order to converge for lower pressure ratios that puts the shock close to the nozzle exit, the constant area region, or if the flow should be supersonic throughout the diverging section. Finally, care had to be taken to ensure that the reservoir pressure was not defined to be too low because it was discovered that this led to solutions that blew up.

Even by lowering the reservoir pressure in the cases that the shock approaches the nozzle exit, the steady state simulation put the shocks at incorrect distances when compared to the theoretical solution. This error in the simulations increased as the pressure ratio was lowered, or when the shock got closer to the nozzle exit. Results are shown in Fig. 4 and 5 where nozzle 2 shows further deviation of the shock location from the theoretical location as the pressure ratio is decreased.



(a) $\frac{p_{atm}}{p_0} = 0.2$ (b) $\frac{p_{atm}}{p_0} = 0.15$ (c) $\frac{p_{atm}}{p_0} = 0.1$

Figure 4: Mach of Nozzle 1 at steady state for $\frac{p_{atm}}{p_0}$ = a). 0.2, b). 0.15, c). 0.1. As the pressure ratio decreases, the simulation shock deviates further from the theoretical shock location.



(a) $\frac{p_{atm}}{p_0} = 0.45$ (b) $\frac{p_{atm}}{p_0} = 0.4$ (c) $\frac{p_{atm}}{p_0} = 0.35$

Figure 5: Mach of Nozzle 2 at steady state for $\frac{p_{atm}}{p_0}$ = a). 0.45, b). 0.4, c). 0.35. As the pressure ratio decreases, the simulation shock deviates further from the theoretical shock location.

References

[1] Leveque, Randall J., Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, New York, NY, 2011
[2] Liepmann, H.W. and Roshko, A., Elements of Gasdynamics, Dover Publications, Mineola, NY, 1985

Appendix

Analytical Shock Location

The theoretical shock location can be solved using isentropic relations, normal shock relations, and an iterative MATLAB code.

1. Guessing a shock location, x_s we can compute the area at the shock location A_s.

2. Assuming choked flow at the throat, and a known area profile, the cross-sectional area at the throat A^* and a downstream area at the guessed shock location A_s are known. From this area ratio, the Mach just upstream of the shock, M_1 can be calculated using the isentropic area relation:

$$\left(\frac{A_s}{A^*}\right)^2 = \frac{1}{M_1^2}\left[\frac{2}{\gamma+1}\left(1+\frac{\gamma-1}{2}M_1^2\right)\right]^{\frac{\gamma+1}{\gamma-1}} \tag{6.1}$$

3. Using a normal shock relation, we can calculate the Mach number M_2 just downstream of the shock:

$$M_2^2 = \frac{1+\frac{\gamma-1}{2}M_1^2}{\gamma M_1^2 - \frac{\gamma-1}{2}} \tag{6.2}$$

4. From M_2, and A_s, we can calculate a fictitious A^*_{exit} for some region downstream of the shock, using Eq. 6.1

5. Using Eq. 6.1 with A^*_{exit} and the known area at the exit A_{exit} , we can calculate the Mach number at the exit, M_{exit}, which should be a subsonic value.

6. We now need to compare the computed ratio of exit pressure to entrance pressure to the given pressure ratio. This is done through a product of pressure ratios, using isentropic and normal shock pressure relations.

   In general, for any point in the flow:

$$\frac{p_t}{p} = \left(1+\frac{\gamma-1}{2}M^2\right)^{\frac{\gamma}{\gamma-1}} \tag{6.3}$$

   Additionally, between any two points in a flow, total pressure remains constant, assuming isentropic flow, unless there is a shock, given by:

$$p_{1t} = p_{2t} \tag{6.4}$$

   In the case of a shock, the normal shock relation for two points on either side of the shock is given by:

$$\frac{p_2}{p_1} = 1+\left(\frac{2\gamma}{\gamma+1}(M_1^2-1)\right) \tag{6.5}$$

   We wish to solve for the pressure ratio $p_{exit}/p_{in} = 1.931$ using the product of pressure ratios:

$$\frac{p_{exit}}{p_{in}} = \underbrace{\frac{p_{exit}}{p_{exit,t}}}_{Eq.6.3}\underbrace{\frac{p_{exit,t}}{p_{2,t}}}_{1}\underbrace{\frac{p_{2,t}}{p_2}}_{Eq.6.3}\underbrace{\frac{p_2}{p_1}}_{Eq.6.5}\underbrace{\frac{p_1}{p_{1,t}}}_{Eq.6.3}\underbrace{\frac{p_{1,t}}{p_{in,t}}}_{1}\underbrace{\frac{p_{in,t}}{p_{in}}}_{Eq.6.3} \tag{6.6}$$

$$\frac{p_{exit}}{p_{in}} = \underbrace{\frac{p_{exit}}{p_{exit,t}}}_{isen}\underbrace{\frac{p_{exit,t}}{p_{2,t}}}_{1}\underbrace{\frac{p_{2,t}}{p_2}}_{isen}\underbrace{\frac{p_2}{p_1}}_{shock}\underbrace{\frac{p_1}{p_{1,t}}}_{isen}\underbrace{\frac{p_{1,t}}{p_{in,t}}}_{1}\underbrace{\frac{p_{in,t}}{p_{in}}}_{isen} \tag{6.7}$$

   In the last pressure ratio $\frac{p_{in,t}}{p_{in}}$, M_{inlet} is set to 0.

7. If this pressure ratio was larger than the prescribed pressure ratio, the new guess for x_s was increased by the MATLAB assignment $x_s = x_s + 0.01$. Conversely, if the pressure ratio was lower than the pressure ratio, the new guess for x_s was decreased by the MATLAB assignment $x_s = x_s - 0.01$. This process was iterated in a loop until convergence.