Task 1

$$\underset{\sim}{u} = \begin{bmatrix} u \\ v \\ p \end{bmatrix}$$

u_x	v_x	p_x	u_y	v_y	p_y
$\frac{\partial u}{\partial x}$				$+\frac{\partial v}{\partial y}$	
$u\frac{\partial u}{\partial x}$		$+\frac{\partial p}{\partial x}$	$+v\frac{\partial u}{\partial y}$		
	$u\frac{\partial v}{\partial x}$			$+v\frac{\partial v}{\partial y}$	$+\frac{\partial p}{\partial y}$

$$\underline{\underline{A}} = \begin{bmatrix} 1 & 0 & 0 \\ u & 0 & 1 \\ 0 & u & 0 \end{bmatrix} \qquad \underline{\underline{B}} = \begin{bmatrix} 0 & 1 & 0 \\ v & 0 & 0 \\ 0 & v & 1 \end{bmatrix}$$

b) $$\underline{\underline{B}} - \lambda \underline{\underline{A}} = \begin{bmatrix} -\lambda & 1 & 0 \\ v-\lambda u & 0 & -\lambda \\ 0 & v-\lambda u & 1 \end{bmatrix}$$

$$\det(\underline{\underline{B}} - \lambda\underline{\underline{A}}) = -\lambda\left[0 + \lambda(v-\lambda u)\right] - 1\left[(v-\lambda u) - 0\right] + 0$$

$$= -\lambda^2(v-\lambda u) - (v-\lambda u)$$

$$= -(v-\lambda u)(\lambda^2+1)$$

$$\lambda = \pm i, \ \frac{v}{u}$$

this one give elliptic behavior

this one gives hyperbolic

Mixture of both.

Task 2 $\quad u_t + f_x = 0 \qquad f = au$

$$u_j^{n+1} = \frac{1}{2}(u_j^n + u_j^*) - \frac{\lambda}{2}\left[a u_j^* - a u_{j-1}^*\right]$$

$$u_j^{n+1} = \frac{1}{2}\left(u_j^n + \left[u_j^n - \lambda(a u_{j+1}^n - a u_j^n)\right]\right)$$
$$- \frac{\lambda a}{2}\left[\{u_j^n - \lambda a(u_{j+1}^n - u_j^n)\} - \{u_{j-1}^n - \lambda a(u_j^n - u_{j-1}^n)\}\right]$$

$$= \frac{1}{2}\left(2u_j^n - \lambda a(u_{j+1}^n - u_j^n)\right) - \frac{\lambda a}{2}\left[u_j^n - u_{j-1}^n - \lambda a(u_{j+1}^n - u_j^n - u_j^n + u_{j-1}^n)\right]$$

$$= \frac{1}{2}\left[2u_j^n - \lambda a(u_{j+1}^n - \cancel{u_j^n} + \cancel{u_j^n} - u_{j-1}^n - \lambda a(u_{j+1}^n - 2u_j^n + u_{j-1}^n))\right]$$

$$= u_j^n - \frac{\lambda a}{2}(u_{j+1}^n - u_{j-1}^n) + \frac{\lambda^2 a^2}{2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

I've honestly never seen this before in a lecture.
(I have missed a few)

But I asked some people and did a google

→ Lax-Wendroff

b) $$\begin{pmatrix} \rho \\ u \end{pmatrix}_t + \underbrace{\begin{pmatrix} u & \rho \\ K\gamma\rho^{\gamma-2} & u \end{pmatrix}}_{\underline{\underline{D}}}\begin{pmatrix} \rho \\ u \end{pmatrix}_x = 0$$

characteristics $dx/dt = \lambda$ with λ e'vals.

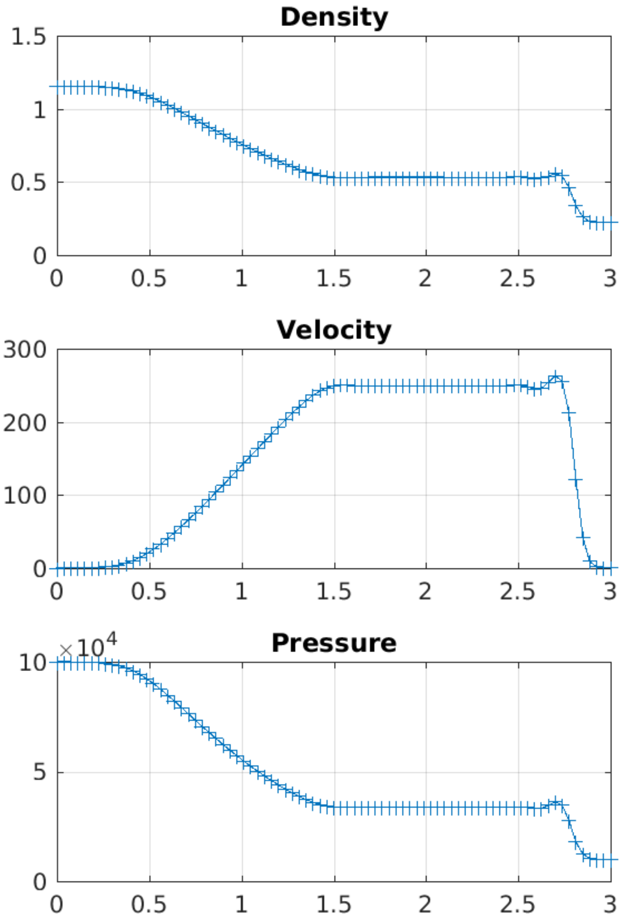
$$\det(\underline{\underline{D}} - \lambda I) = (u-\lambda)^2 - K\gamma\rho^{\gamma-1} = 0$$

$$\Rightarrow \lambda = u \pm \sqrt{K\gamma\rho^{\gamma-1}}$$

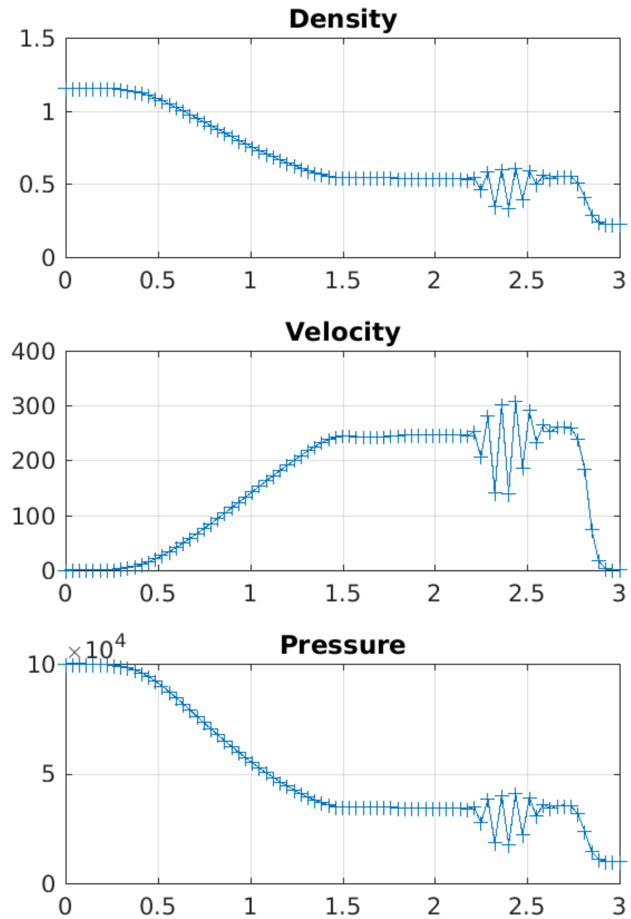
$$= u \pm c \qquad \text{with } c^2 = K\gamma\rho^{\gamma-1}$$

C0 = 0.05, C2 = 0.45

CN = 0.5

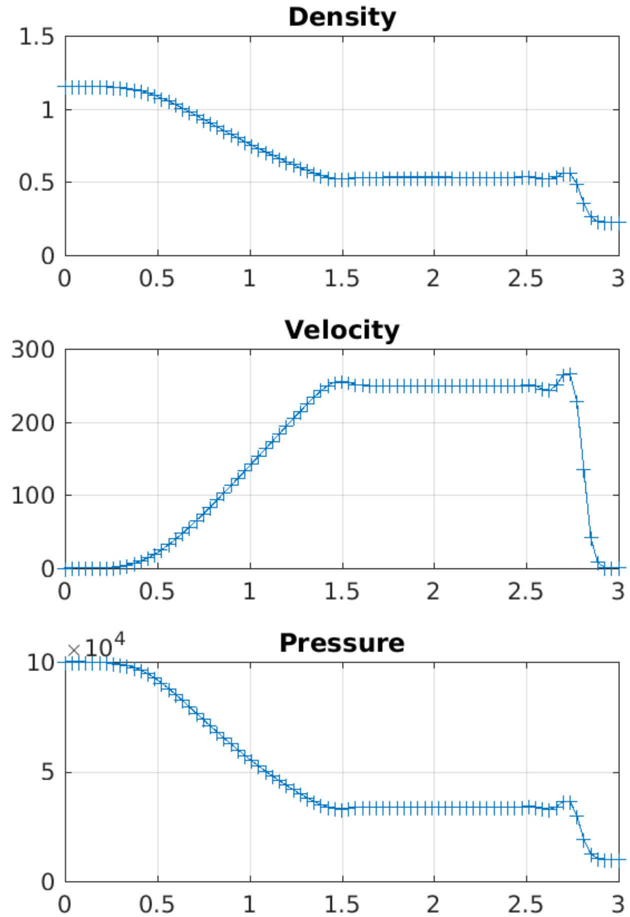


CN = 0.9

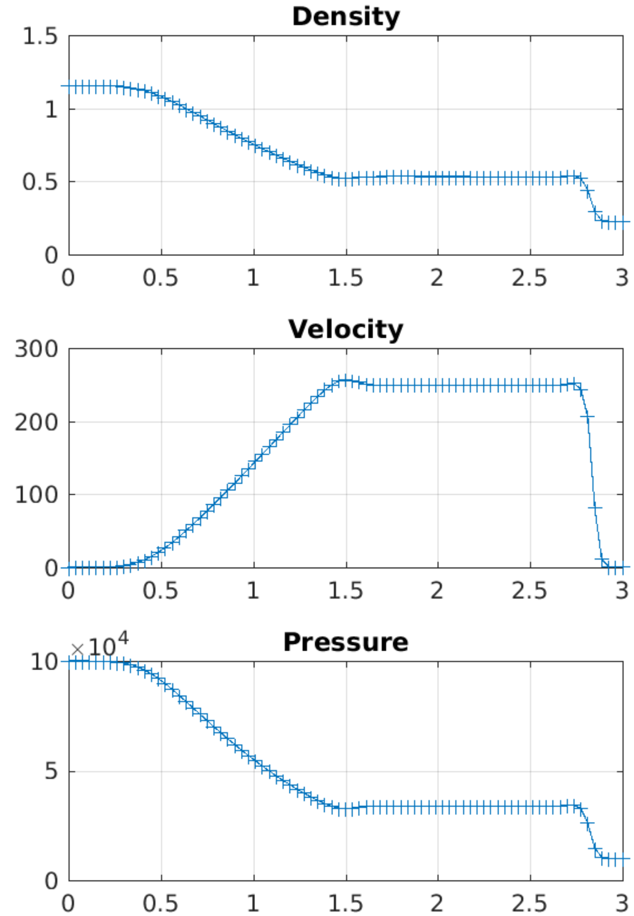


C0 = 0.05, C2 = 0.05

CN = 0.5

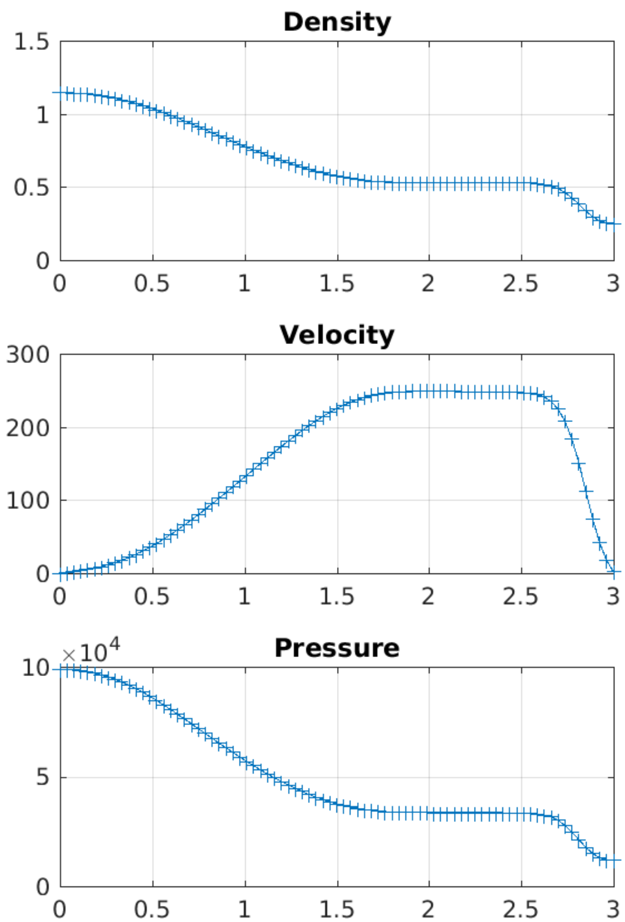


CN = 0.9

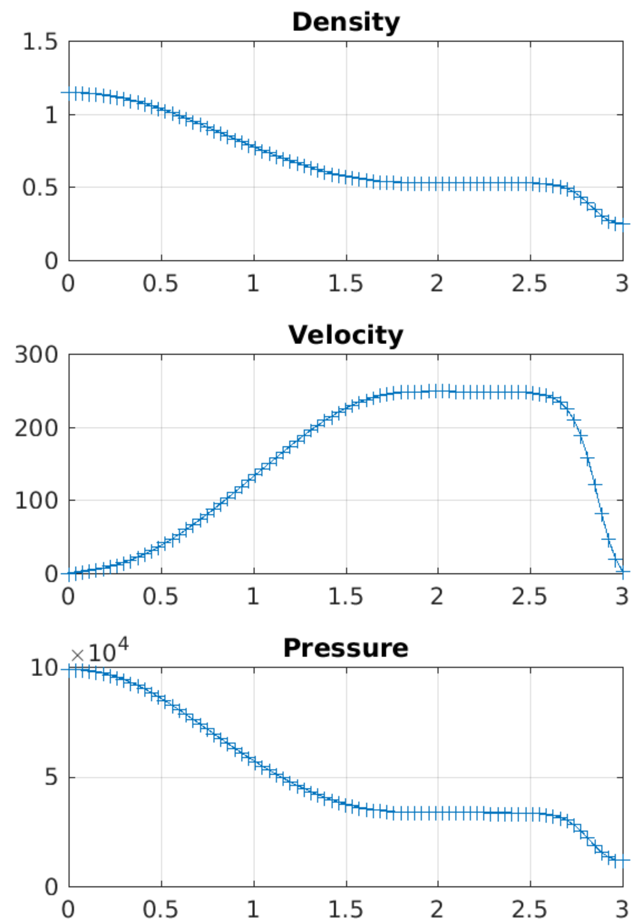


C0 = 0.4, C2 = 0.05

CN = 0.5

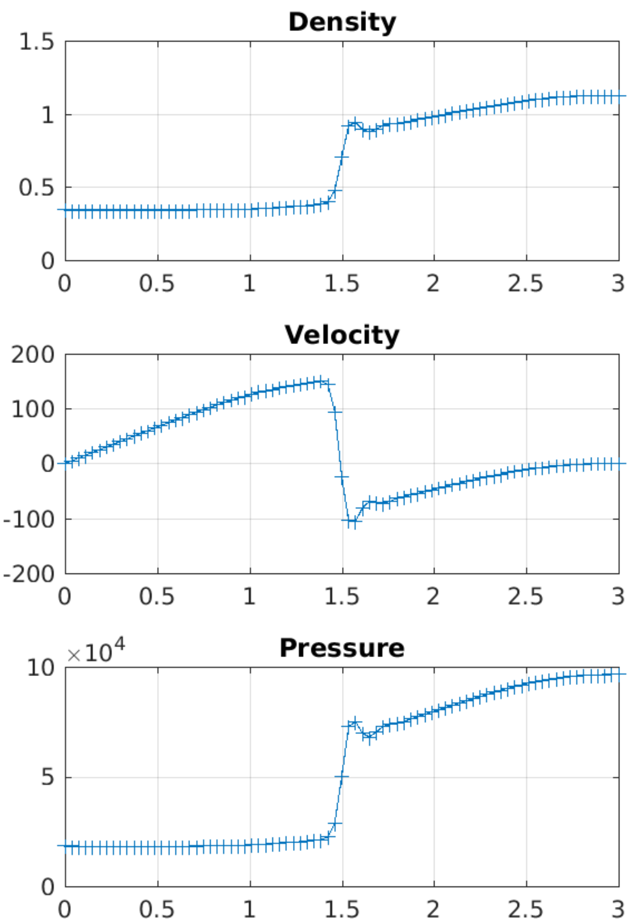


CN = 0.9

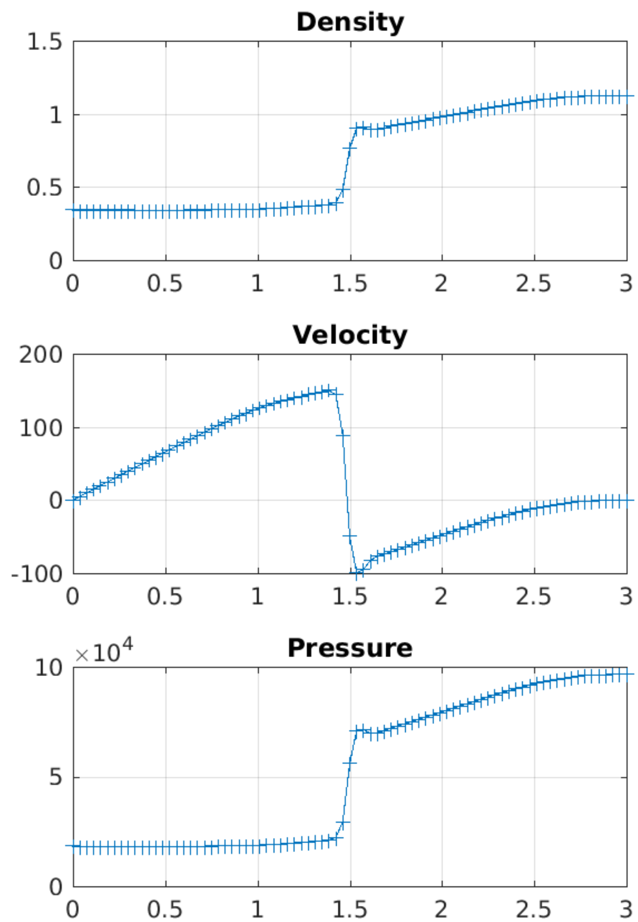


C0 = 0.05, C2 = 0.25

CN = 0.5



CN = 0.9



Part C:

An exerpt from my Matlab code for how I calculated the speed of the shock:

```
% ---------------------------------------------------------------------
% Track shock
% I know this is a dodgy way of tracking the shock.
% (it doesn't work after reflection)

% shock(:,1) x location of shock
% shock(:,2) time
% shock(:,3) estimated shock velocity from Rankine-Hugoniot condition

if t_cum > 0.001 && t_cum < 0.003
k = k+1;
d_rho = (rho(2:end) - rho(1:end-1)); % slope of density
shock_index = find(abs(d_rho) == max(abs(d_rho))); % assume shock has highest slope
shock(k,1) = X(shock_index);
shock(k,2) = t_cum;
d_rhoU = (U(2:end,2) - U(1:end-1,2))/DX; % slope of rho*U
rhoUL = mean(U([d_rhoU;0] < 10 & u > 10,2)); % mean of the flat part & non zero of rho*U
rhoL = mean(U([d_rhoU;0] < 10 & u > 10,1)); % mean of the flat part & non zero of rho*U
shock(k,3) = rhoUL/(rhoL-RHO1);
end
% ---------------------------------------------------------------------
```

The velocity is then estimated with a linear fit:

```
coefs = polyfit(shock(10:end,2),shock(10:end,1),1); % tendline for shock
s = mean(shock(10:end,3)); % average of analytic solution
disp(['CN:',num2str(CN(j)),', C0:',num2str(C0_in(i)),', C2:',num2str(C2_in(i)),...
        '   Measured: ',num2str(coefs(1)),'m/s   Analytic: ',num2str(s),'m/s'])
```

These were my results:

CN:0.5, C0:0.05, C2:0.45 Measured: 428.7524m/s Analytic: 426.2748m/s
CN:0.5, C0:0.05, C2:0.05 Measured: 426.0985m/s Analytic: 430.1961m/s
CN:0.5, C0:0.4, C2:0.05 Measured: 430.0152m/s Analytic: 416.1131m/s
CN:0.5, C0:0.05, C2:0.25 Measured: 429.1039m/s Analytic: 427.1592m/s
CN:0.9, C0:0.05, C2:0.45 Measured: 310.4498m/s Analytic: 428.6295m/s
CN:0.9, C0:0.05, C2:0.05 Measured: 430.3273m/s Analytic: 430.0618m/s
CN:0.9, C0:0.4, C2:0.05 Measured: 424.1953m/s Analytic: 416.4721m/s
CN:0.9, C0:0.05, C2:0.25 Measured: 426.6384m/s Analytic: 427.5941m/s

The CN:0.9, C0:0.05, C2:0.45 case doesn’t count because it had wiggles and I bet my assumption that “the shock has the highest slope” was violated in that case.