Caleb Logemann
MATH667 Hyperbolic Partial Differential Equations
Final Project

1. Write out the Jacobian matrix and derive the eigenvalues and eigenvectors for the Euler equations.

   We have the following vector flux function written in terms of the conserved variables.

$$\mathbf{f}(\mathbf{w}) = \begin{bmatrix} m \\ \frac{m^2}{\rho} + P \\ \frac{m}{\rho}(E + P) \end{bmatrix}$$

   where

$$P = (\gamma - 1)\left(E - \frac{m^2}{2\rho}\right)$$

   or

$$E = \frac{P}{\gamma - 1} + \frac{\rho u^2}{2}$$

   In order to compute the Jacobian of $\mathbf{f}$ I will first compute the partial derivatives of P.

$$P_\rho = (\gamma - 1)\frac{m^2}{2\rho^2}$$
$$P_m = -(\gamma - 1)\frac{m}{\rho}$$
$$P_E = \gamma - 1$$

   Now the Jacobian of $\mathbf{f}$ can expressed as

$$\mathbf{f}'(\mathbf{w}) = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{m^2}{\rho^2} + P_\rho & 2\frac{m}{\rho} + P_m & P_E \\ P_\rho \frac{m}{\rho} - (E + P)\frac{m}{\rho^2} & P_m \frac{m}{\rho} + (E + P)\frac{1}{\rho} & \frac{m}{p}(1 + P_E) \end{bmatrix}$$

   Simplifying this results in

$$\mathbf{f}'(\mathbf{w}) = \begin{bmatrix} 0 & 1 & 0 \\ (\gamma - 3)\frac{m^2}{2\rho^2} & (3 - \gamma)\frac{m}{\rho} & \gamma - 1 \\ (\gamma - 1)\frac{m^3}{2\rho^3} - m\frac{E+P}{\rho^2} & \frac{E+P}{\rho} - (\gamma - 1)\frac{m^2}{\rho^2} & \gamma\frac{m}{p} \end{bmatrix}$$

   This can also be changed into primitive variables.

$$\mathbf{f}'(\mathbf{w}) = \begin{bmatrix} 0 & 1 & 0 \\ (\gamma - 3)\frac{u^2}{2} & (3 - \gamma)u & \gamma - 1 \\ \frac{1}{2}(\gamma - 2)u^3 - \frac{\gamma}{\gamma-1}\frac{uP}{\rho} & \frac{\gamma}{\gamma-1}\frac{P}{\rho} + \left(\frac{3}{2} - \gamma\right)u^2 & \gamma u \end{bmatrix}$$

   Now in order to find the eigenvalues and eigenvectors of this matrix we start by subtracting λI and solving $\det(\mathbf{f}'(\mathbf{w}) - \lambda I) = 0$.

$$\det(\mathbf{f}'(\mathbf{w}) - \lambda I) = \begin{vmatrix} -\lambda & 1 & 0 \\ (\gamma - 3)\frac{u^2}{2} & (3 - \gamma)u - \lambda & \gamma - 1 \\ \frac{1}{2}(\gamma - 2)u^3 - \frac{\gamma}{\gamma-1}\frac{uP}{\rho} & \frac{\gamma}{\gamma-1}\frac{P}{\rho} + \left(\frac{3}{2} - \gamma\right)u^2 & \gamma u - \lambda \end{vmatrix}$$
$$= -\lambda((3 - \gamma)u - \lambda)(\gamma u - \lambda) + (\gamma - 1)\left(\frac{1}{2}(\gamma - 2)u^3 - \frac{\gamma}{\gamma - 1}\frac{uP}{\rho}\right)$$
$$+\lambda(\gamma - 1)\left(\frac{\gamma}{\gamma - 1}\frac{P}{\rho} + \left(\frac{3}{2} - \gamma\right)u^2\right) - (\gamma - 3)\frac{u^2}{2}(\gamma u - \lambda)$$

I will separate out each of these terms as

$$
\begin{aligned}
a_1 &= -\lambda((3-\gamma)u-\lambda)(\gamma u-\lambda) \\
a_2 &= (\gamma-1)\left(\frac{1}{2}(\gamma-2)u^3 - \frac{\gamma}{\gamma-1}\frac{uP}{\rho}\right) \\
a_3 &= \lambda(\gamma-1)\left(\frac{\gamma}{\gamma-1}\frac{P}{\rho} + \left(\frac{3}{2}-\gamma\right)u^2\right) \\
a_4 &= -(\gamma-3)\frac{u^2}{2}(\gamma u-\lambda)
\end{aligned}
$$

so the determinant is $a_1+a_2+a_3+a_4$. Now I will simplify each of these terms

$$
\begin{aligned}
a_1 &= -\lambda((3-\gamma)u-\lambda)(\gamma u-\lambda) \\
&= \Big((\gamma-3)u\lambda+\lambda^2\Big)(\gamma u-\lambda) \\
&= (\gamma-3)\gamma u^2\lambda - (\gamma-3)u\lambda^2 + \gamma u\lambda^2 - \lambda^3 \\
&= (\gamma-3)\gamma u^2\lambda + 3u\lambda^2 - \lambda^3
\end{aligned}
$$

$$
\begin{aligned}
a_2 &= (\gamma-1)\left(\frac{1}{2}(\gamma-2)u^3 - \frac{\gamma}{\gamma-1}\frac{uP}{\rho}\right) \\
&= \frac{1}{2}(\gamma-1)(\gamma-2)u^3 - \gamma\frac{uP}{\rho}
\end{aligned}
$$

$$
\begin{aligned}
a_3 &= \lambda(\gamma-1)\left(\frac{\gamma}{\gamma-1}\frac{P}{\rho} + \left(\frac{3}{2}-\gamma\right)u^2\right) \\
&= \lambda\left(\gamma\frac{P}{\rho} + (\gamma-1)\left(\frac{3}{2}-\gamma\right)u^2\right)
\end{aligned}
$$

$$
\begin{aligned}
a_4 &= -(\gamma-3)\frac{u^2}{2}(\gamma u-\lambda) \\
&= \lambda(\gamma-3)\frac{u^2}{2} - \gamma(\gamma-3)\frac{u^3}{2}
\end{aligned}
$$

Adding these all back together gives

$$
\begin{gathered}
a_1+a_2+a_3+a_4 = -\lambda^3 + 3u\lambda^2 \\
+\left(\gamma\frac{P}{\rho} + (\gamma-1)\left(\frac{3}{2}-\gamma\right)u^2 + (\gamma-3)\left(\gamma+\frac{1}{2}\right)u^2\right)\lambda - \gamma(\gamma-3)\frac{u^3}{2} + \frac{1}{2}(\gamma-1)(\gamma-2)u^3 - \gamma\frac{uP}{\rho}
\end{gathered}
$$

Now I will simplify each of the coefficients of λ.

$$
\begin{aligned}
&\gamma\frac{P}{\rho} + (\gamma-1)\left(\frac{3}{2}-\gamma\right)u^2 + (\gamma-3)\left(\gamma+\frac{1}{2}\right)u^2 \\
&= \gamma\frac{P}{\rho} + \left((\gamma-1)\left(\frac{3}{2}-\gamma\right) + (\gamma-3)\left(\gamma+\frac{1}{2}\right)\right)u^2 \\
&= \gamma\frac{P}{\rho} + \left(-\gamma^2+\frac{5}{2}\gamma-\frac{3}{2}+\gamma^2-\frac{5}{2}\gamma-\frac{3}{2}\right)u^2 \\
&= \gamma\frac{P}{\rho} + -3u^2
\end{aligned}
$$

$$
\begin{aligned}
&- \gamma(\gamma-3)\frac{u^3}{2} + \frac{1}{2}(\gamma-1)(\gamma-2)u^3 - \gamma\frac{uP}{\rho} \\
&= (-\gamma(\gamma-3) + (\gamma-1)(\gamma-2))\frac{u^3}{2} - \gamma\frac{uP}{\rho} \\
&= \left(-\gamma^2 + 3\gamma + \gamma^2 - 3\gamma + 2\right)\frac{u^3}{2} - \gamma\frac{uP}{\rho} \\
&= u^3 - \gamma\frac{uP}{\rho}
\end{aligned}
$$

So now simplifying these terms by factoring into a difference cubed

$$
\begin{aligned}
a_1 + a_2 + a_3 + a_4 &= -\lambda^3 + 3u\lambda^2 + \left(\gamma\frac{P}{\rho} + -3u^2\right)\lambda + u^3 - \gamma\frac{uP}{\rho} \\
&= -\lambda^3 + 3u\lambda^2 - 3u^2\lambda + u^3 + \gamma\lambda\frac{P}{\rho} - \gamma\frac{uP}{\rho} \\
&= (u-\lambda)^3 - (u-\lambda)\frac{\gamma P}{\rho}
\end{aligned}
$$

Thus we have shown that

$$
\det(\mathbf{f}'(\mathbf{w}) - \lambda I) = (u-\lambda)^3 - (u-\lambda)\frac{\gamma P}{\rho}
$$

Setting this equal to zero and solving for λ gives.

$$
\begin{aligned}
(u-\lambda)^3 - (u-\lambda)\frac{\gamma P}{\rho} &= 0 \\
(u-\lambda)\left((u-\lambda)^2 - \frac{\gamma P}{\rho}\right) &= 0 \\
\lambda &= u \\
(u-\lambda)^2 - \frac{\gamma P}{\rho} &= 0 \\
(u-\lambda)^2 &= \frac{\gamma P}{\rho} \\
(u-\lambda) &= \pm\sqrt{\frac{\gamma P}{\rho}} \\
\lambda &= u \pm \sqrt{\frac{\gamma P}{\rho}}
\end{aligned}
$$

So the three eigenvalues of this system are

$$
\lambda_1 = u, \quad \lambda_2 = u + \sqrt{\frac{\gamma P}{\rho}}, \quad \lambda_3 = u - \sqrt{\frac{\gamma P}{\rho}}
$$

Now I will just check that we have the correct eigenvectors. The eigenvectors are

$$
\mathbf{w}_1 = \begin{bmatrix} 1 \\ u \\ \frac{1}{2}u^2 \end{bmatrix}, \quad \mathbf{w}_2 = \begin{bmatrix} 1 \\ u + \sqrt{\frac{\gamma P}{\rho}} \\ \frac{\gamma}{\gamma-1}\frac{P}{\rho} + \frac{u^2}{2} + u\sqrt{\frac{\gamma P}{\rho}} \end{bmatrix}, \quad \mathbf{w}_3 = \begin{bmatrix} 1 \\ u - \sqrt{\frac{\gamma P}{\rho}} \\ \frac{\gamma}{\gamma-1}\frac{P}{\rho} + \frac{u^2}{2} - u\sqrt{\frac{\gamma P}{\rho}} \end{bmatrix}
$$

To check

$$
\begin{aligned}
\mathbf{f}'\mathbf{w}_1 &= \begin{bmatrix} 0 & 1 & 0 \\ (\gamma-3)\frac{u^2}{2} & (3-\gamma)u & \gamma-1 \\ \frac{1}{2}(\gamma-2)u^3 - \frac{\gamma}{\gamma-1}\frac{uP}{\rho} & \frac{\gamma}{\gamma-1}\frac{P}{\rho} + \left(\frac{3}{2}-\gamma\right)u^2 & \gamma u \end{bmatrix} \begin{bmatrix} 1 \\ u \\ \frac{1}{2}u^2 \end{bmatrix} \\
&= \begin{bmatrix} u \\ (\gamma-3)\frac{u^2}{2} - (\gamma-3)u^2 + (\gamma-1)\frac{u^2}{2} \\ \frac{1}{2}(\gamma-2)u^3 - \frac{\gamma}{\gamma-1}\frac{uP}{\rho} + \frac{\gamma}{\gamma-1}\frac{uP}{\rho} + \left(\frac{3}{2}-\gamma\right)u^3 + \gamma\frac{u^3}{2} \end{bmatrix} \\
&= \begin{bmatrix} u \\ u^2 \\ \frac{1}{2}u^3 \end{bmatrix} \\
&= u \begin{bmatrix} 1 \\ u \\ \frac{1}{2}u^2 \end{bmatrix}
\end{aligned}
$$

Let $c = \sqrt{\frac{\gamma P}{\rho}}$

$$
\begin{aligned}
\mathbf{f}'\mathbf{w}_2 &= \begin{bmatrix} 0 & 1 & 0 \\ (\gamma-3)\frac{u^2}{2} & (3-\gamma)u & \gamma-1 \\ \frac{1}{2}(\gamma-2)u^3 - \frac{\gamma}{\gamma-1}\frac{uP}{\rho} & \frac{\gamma}{\gamma-1}\frac{P}{\rho} + \left(\frac{3}{2}-\gamma\right)u^2 & \gamma u \end{bmatrix} \begin{bmatrix} 1 \\ u+c \\ \frac{1}{\gamma-1}c^2 + \frac{u^2}{2} + uc \end{bmatrix} \\
&= \begin{bmatrix} u+c \\ (\gamma-3)\frac{u^2}{2} + (3-\gamma)u(u+c) + (\gamma-1)\left(\frac{1}{\gamma-1}c^2 + \frac{u^2}{2} + uc\right) \\ \frac{1}{2}(\gamma-2)u^3 - \frac{\gamma}{\gamma-1}\frac{uP}{\rho} + \left(\frac{\gamma}{\gamma-1}\frac{P}{\rho} + \left(\frac{3}{2}-\gamma\right)u^2\right)(u+c) + \gamma u\left(\frac{1}{\gamma-1}c^2 + \frac{u^2}{2} + uc\right) \end{bmatrix} \\
&= \begin{bmatrix} u+c \\ (\gamma-3)\frac{u^2}{2} + (3-\gamma)u^2 + (3-\gamma)uc + c^2 + (\gamma-1)\frac{u^2}{2} + (\gamma-1)uc \\ \frac{1}{2}(\gamma-2)u^3 + uc^2 + \frac{1}{\gamma-1}c^2(u+c) + \left(\frac{3}{2}-\gamma\right)u^3 + \left(\frac{3}{2}-\gamma\right)u^2c + \gamma\frac{u^3}{2} + \gamma u^2 c \end{bmatrix} \\
&= \begin{bmatrix} u+c \\ u^2 + 2uc + c^2 \\ \frac{1}{2}u^3 + \frac{1}{\gamma-1}c^2(u+c) + \frac{3}{2}u^2c + uc^2 \end{bmatrix} \\
&= \begin{bmatrix} u+c \\ (u+c)^2 \\ \left(\frac{u^2}{2} + uc\right)(u+c) + \frac{1}{\gamma-1}c^2(u+c) \end{bmatrix} \\
&= (u+c) \begin{bmatrix} 1 \\ u+c \\ \frac{1}{\gamma-1}c^2 + \frac{u^2}{2} + uc \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{f}'\mathbf{w}_3 &= \begin{bmatrix} 0 & 1 & 0 \\ (\gamma-3)\frac{u^2}{2} & (3-\gamma)u & \gamma-1 \\ \frac{1}{2}(\gamma-2)u^3 - \frac{\gamma}{\gamma-1}\frac{uP}{\rho} & \frac{\gamma}{\gamma-1}\frac{P}{\rho} + \left(\frac{3}{2}-\gamma\right)u^2 & \gamma u \end{bmatrix} \begin{bmatrix} 1 \\ u-c \\ \frac{1}{\gamma-1}c^2 + \frac{u^2}{2} - uc \end{bmatrix} \\
&= \begin{bmatrix} u-c \\ (\gamma-3)\frac{u^2}{2} + (3-\gamma)u(u-c) + (\gamma-1)\left(\frac{1}{\gamma-1}c^2 + \frac{u^2}{2} - uc\right) \\ \frac{1}{2}(\gamma-2)u^3 - \frac{\gamma}{\gamma-1}\frac{uP}{\rho} + \left(\frac{\gamma}{\gamma-1}\frac{P}{\rho} + \left(\frac{3}{2}-\gamma\right)u^2\right)(u-c) + \gamma u\left(\frac{1}{\gamma-1}c^2 + \frac{u^2}{2} - uc\right) \end{bmatrix} \\
&= \begin{bmatrix} u-c \\ (\gamma-3)\frac{u^2}{2} + (3-\gamma)u^2 - (3-\gamma)uc + c^2 + (\gamma-1)\frac{u^2}{2} - (\gamma-1)uc \\ \frac{1}{2}(\gamma-2)u^3 + uc^2 + \frac{1}{\gamma-1}c^2(u-c) + \left(\frac{3}{2}-\gamma\right)u^3 - \left(\frac{3}{2}-\gamma\right)u^2c + \gamma\frac{u^3}{2} - \gamma u^2 c \end{bmatrix} \\
&= \begin{bmatrix} u-c \\ u^2 - 2uc + c^2 \\ \frac{1}{2}u^3 + \frac{1}{\gamma-1}c^2(u-c) - \frac{3}{2}u^2c + uc^2 \end{bmatrix} \\
&= \begin{bmatrix} u-c \\ (u-c)^2 \\ \left(\frac{u^2}{2} - uc\right)(u-c) + \frac{1}{\gamma-1}c^2(u-c) \end{bmatrix} \\
&= (u-c)\begin{bmatrix} 1 \\ u-c \\ \frac{1}{\gamma-1}c^2 + \frac{u^2}{2} - uc \end{bmatrix}
\end{aligned}
$$

2. The following is my function for 3rd order MUSCL scheme on a system of equations.

```
function [L] = muscl3System(w, f, deltaX, deltaT, RFunc, LambdaFunc)
    [n, nGridCells] = size(w);
    L = zeros(n, nGridCells);
    nu = 1/deltaX;
    %a = deltaX/(2*deltaT);

    boundaryConditions = 'zeroFlux';

    % flux array, F(:, i) is flux at i + 1/2 interface
    F = zeros(n, nGridCells);

    % compute fluxes at boundaries
    for j = 1:nGridCells
        % boundary conditions
        jm1 = j-1;
        if (j == 1)
            if (strcmp(boundaryConditions,'periodic'))
                jm1 = nGridCells;
            elseif (strcmp(boundaryConditions,'zeroFlux'))
                jm1 = 1;
            end
        end

        jp1 = j+1;
        jp2 = j+2;
        if (j == nGridCells)
            if (strcmp(boundaryConditions,'periodic'))
                jp1 = 1;
                jp2 = 2;
```

```
        elseif (strcmp(boundaryConditions,'zeroFlux'))
            jp1 = nGridCells;
            jp2 = nGridCells;
        end
    elseif (j == nGridCells - 1)
        if (strcmp(boundaryConditions,'periodic'))
            jp2 = 1;
        elseif (strcmp(boundaryConditions,'zeroFlux'))
            jp2 = nGridCells;
        end
    end
    wjm1 = w(:,jm1);
    wj = w(:,j);
    wjp1 = w(:,jp1);
    wjp2 = w(:,jp2);

    % Component wise
    %wminus = -(1/6)*wjm1 + (5/6)*wj + (1/3)*wjp1;
    %wplus = (1/3)*wj + (5/6)*wjp1 - (1/6)*wjp2;
    %wtilde = wminus - wj;
    %wdoubletilde = wplus + wjp1;
    %wtildemod = minmod3System(wtilde, wjp1 - wj, wj - wjm1);
    %wdoubletildemod = minmod3System(wdoubletilde, wjp2 - wjp1, wjp1 - wj);

    %wminusmod = wj + wtildemod;
    %wplusmod = wjp1 - wdoubletildemod;
    %F(:,j) = a*(wminusmod - wplusmod) + 0.5*(f(wminusmod) + f(wplusmod));

    % Characteristic Wise
    % reference solution
    wtilde = 0.5*(wj + wjp1);
    R = RFunc(wtilde);
    Lambda = LambdaFunc(wtilde);
    a = max(max(abs(Lambda)));

    vjm1 = R\wjm1;
    vj = R\wj;
    vjp1 = R\wjp1;
    vjp2 = R\wjp2;

    vminus = -(1/6)*vjm1 + (5/6)*vj + (1/3)*vjp1;
    vplus = (1/3)*vj + (5/6)*vjp1 - (1/6)*vjp2;
    vtilde = vminus - vj;
    vdoubletilde = vplus + vjp1;

    vtildemod = minmod3System(vtilde, vjp1 - vj, vj - vjm1);
    vdoubletildemod = minmod3System(vdoubletilde, vjp2 - vjp1, vjp1 - vj);

    vminusmod = vj + vtildemod;
    vplusmod = vjp1 - vdoubletildemod;

    wminus = R*vminusmod;
    wplus = R*vplusmod;
    %w1 = wminus - wj;
    %w2 = wplus - wj;

    %F(:,j) = f(wj + minmodSystem(w1,w2));

    F(:,j) = a*(wminus - wplus) + 0.5*(f(wminus) + f(wplus));
end
```

```
    % update solution
    for j = 1:nGridCells
        % boundary conditions
        jm1 = j-1;
        if (j == 1)
            if (strcmp(boundaryConditions,'periodic'))
                jm1 = nGridCells;
            elseif (strcmp(boundaryConditions,'zeroFlux'))
                jm1 = 1;
            end
        end

        L(:,j) = nu*(F(:,jm1) - F(:,j));
    end
end
```

I use the following RK3 method.

```
function [sol] = rungeKutta3(L, t, w0)
    [nEqns, nCells] = size(w0);
    nTimeSteps = length(t)-1;
    deltaT = diff(t);
    k = zeros(nEqns, nCells, 3);
    sol = zeros(nEqns, nCells, nTimeSteps+1);
    sol(:,:,1) = w0;

    alpha = [1/6, 2/3, 1/6];
    lambda = [0, 0, 0; 1/2, 0, 0; -1, 2, 0];
    for n = 1:nTimeSteps
        k(:,:,1) = L(t(n), sol(:, :, n));
        k(:,:,2) = L(t(n) + sum(lambda(2,:))*deltaT(n), sol(:,:,n) + deltaT(n)*lambda
            ↪ (2,1)*k(:,:,1));
        k(:,:,3) = L(t(n) + sum(lambda(3,:))*deltaT(n), sol(:,:,n) + deltaT(n)*(lambda
            ↪ (3,1)*k(:,:,1) + lambda(3,2)*k(:,:,2)));
        sol(:,:,n+1) = sol(:,:,n) + deltaT(n)*(alpha(1)*k(:,:,1) + alpha(2)*k(:,:,2) +
            ↪ alpha(3)*k(:,:,3));
        disp(n/nTimeSteps);
    end

%     alpha = [1, 0, 0; 3/4, 1/4, 0; 1/3, 0, 2/3];
%     beta = [1, 0, 0; 0, 1/4, 0; 0, 0, 2/3];
%     for n = 1:nTimeSteps
%         k(:,:,1) = sol(:,:,n) + beta(1,1)*deltaT(n)*L(t(n), sol(:,:,n));
%         k(:,:,2) = alpha(2,1)*sol(:,:,n) + alpha(2,2)*k(:,:,1) + beta(2,2)*deltaT(n)*
    ↪ L(t(n),k(:,:,1));
%         k(:,:,3) = alpha(3,1)*sol(:,:,n) + alpha(3,3)*k(:,:,2) + beta(3,3)*deltaT(n)*
    ↪ L(t(n),k(:,:,2));
%         sol(:,:,n+1) = k(:,:,3);
%         disp(n/nTimeSteps);
%     end
end
```

This script now uses the previous methods to solve the given problem.

```
g = 1.4;
energyFuncPrimitive = @(rho, u, P) P/(g - 1) + 0.5*rho*u^2;
energyFunc = @(rho, m, P) P/(g - 1) + 0.5*m^2/rho;
pressureFuncPrimitive = @(rho, u, E) (g - 1)*(E - 0.5*rho*u^2);
pressureFunc = @(rho, m, E) (g - 1)*(E - 0.5*m^2/rho);
```

```
rhoL = 1;
uL = 0;
mL = rhoL*uL;
PL = 1;
EL = energyFuncPrimitive(rhoL, uL, PL);
rhoR = 0.125;
uR = 0;
mR = rhoR*uR;
PR = 0.1;
ER = energyFuncPrimitive(rhoR, uR, PR);

w0func = @(x) [rhoL, mL, EL]'*(x <= 0) + [rhoR, mR, ER]'*(x > 0);
a = -5;
b = 5;
fConservative = @(rho, m, E) [m; m^2/rho + pressureFunc(rho, m, E); m/rho*(E +
    ↪ pressureFunc(rho, m, E))];
fPrimitive = @(rho, u, P) [rho*u; rho*u^2 + P; u*(energyFuncPrimitive(rho, u, P) + P)];
f = @(w) fConservative(w(1), w(2), w(3));
N = 1000;
tFinal = 2.0;
deltaX = (b - a)/N;
x = linspace(a+0.5*deltaX, b-0.5*deltaX, N);
w0 = w0func(x);

RPrimitive = @(rho, u, P) [1, 1, 1; u - sqrt(g*P/rho), u, u + sqrt(g*P/rho); g/(g - 1)*
    ↪ P/rho + u^2/2 - u*sqrt(g*P/rho), 0.5*u^2, g/(g - 1)*P/rho + u^2/2 - u*sqrt(g*P/
    ↪ rho)];
RConservative = @(rho, m, E) RPrimitive(rho, m/rho, pressureFunc(rho, m, E));
RFunc = @(w) RConservative(w(1), w(2), w(3));
multByR = @(w) RFunc(w)*w;
%multByRInverse = @(w) RFunc(w)\w;

LambdaPrimitive = @(rho, u, P) [u - sqrt(g*P/rho), 0, 0; 0, u, 0; 0, 0, u + sqrt(g*P/
    ↪ rho)];
LambdaConservative = @(rho, m, E) LambdaPrimitive(rho, m/rho, pressureFunc(rho, m, E));
LambdaFunc = @(w) LambdaConservative(w(1), w(2), w(3));
multByLambda = @(w) LambdaFunc(w)*w;

cfl = 0.1;
deltaT = cfl*deltaX;
nTimeSteps = ceil(tFinal/deltaT);
deltaT = tFinal/nTimeSteps;

t = 0:deltaT:tFinal;

%v0 = multByRInverse(w0);

L = @(t, u) muscl3System(u, f, deltaX, deltaT, RFunc, LambdaFunc);
%rk3 = NumericalAnalysis.ODES.standardRK3Method;
%sol = rk3.solveSystem(L, t, w0);
sol = rungeKutta3(L, t, w0);
rho = sol(1,:,end);
u = sol(2,:,end)./sol(1,:,end);
p = (g - 1)*(sol(3,:,end) - 0.5*rho.*u.^2);
subplot(3, 1, 1);
plot(x, rho, 'LineWidth', 2);
xlabel('x');
ylabel('rho');
title('Density');
subplot(3, 1, 2);
```

```
plot(x, u, 'LineWidth', 2);
xlabel('x');
ylabel('u');
title('Velocity');
subplot(3, 1, 3);
plot(x, p, 'LineWidth', 2);
xlabel('x');
ylabel('P');
title('Pressure');
saveas(gcf, 'Figures/finalProject.png', 'png');
figure;
plot(x, sol(:, :,end), 'k--', 'LineWidth', 2);
xlabel('x');
ylabel('w');
title(strcat('Euler equations at t = ', num2str(tFinal)));
saveas(gcf, 'Figures/finalProject_2.png', 'png');
```

The following image is output.