```
clc;clear;close all;

gam = 1.4;

% Grid Generation

nx = 60;
ny = 20;

Grid = GridGen(nx,ny);

x = Grid.x;         % 2D array [nx+1,ny+1], contains x node locations of grid
y = Grid.y;         % 2D array [nx+1,ny+1], contains y node locations of grid
xc = Grid.xc;       % 2D array [nx,ny], contains x cell center locations of grid
yc = Grid.yc;       % 2D array [nx,ny], contains y cell center locations of grid
area = Grid.area;   % 2D array [nx,ny], contains areas of all cells
edge = Grid.edge;   % 3D array [nx,ny,4], contains edge lengths of all cells
normx = Grid.normx; % 3D array [nx,ny,4], contains x-component of outward normals of↙
edges of all cells
normy = Grid.normy; % 3D array [nx,ny,4], contains y-component of outward normals of↙
edges of all cells

nnodes = nx*ny;

R = 287;
pL = 10^5;
TL = 300;
ML = 2;

rhoL = pL/(R*TL);
cL = sqrt(gam*pL/rhoL);
uL = ML*cL;
vL = 0;
EL = (pL/(gam-1)) + 0.5*rhoL*(uL^2);

VL = [rhoL; uL; vL; pL];
WL = [rhoL; rhoL*uL; vL; EL];

%% Setting Initial Conditions

W0 = zeros(nx,ny,4);
V0 = zeros(nx,ny,4);

for i = 1:nx
    for j = 1:ny
        W0(i,j,:) = WL;
        V0(i,j,:) = VL;
    end
end

W_old = W0;
V_old = V0;

W = W0;
V = V0;


%% Local Time Stepping
```

```
time = 0;

dtG = 1;
index = 1;
Res = ones(1,4);

while time < Inf
    % Internal Domain
    for i = 2:nx-1
        for j = 2:ny-1

            Wi = [W_old(i,j,1);W_old(i,j,2);W_old(i,j,3);W_old(i,j,4)];

            Wj = W_old(i+1,j,:);
            Wk = W_old(i,j+1,:);
            Wl = W_old(i-1,j,:);
            Wm = W_old(i,j-1,:);

            nij = edge(i,j,1)*[normx(i,j,1);normy(i,j,1)];
            nik = edge(i,j,2)*[normx(i,j,2);normy(i,j,2)];
            nil = edge(i,j,3)*[normx(i,j,3);normy(i,j,3)];
            nim = edge(i,j,4)*[normx(i,j,4);normy(i,j,4)];

            Fij = roe_solver_2d(Wi,Wj,nij);
            Fik = roe_solver_2d(Wi,Wk,nik);
            Fil = roe_solver_2d(Wi,Wl,nil);
            Fim = roe_solver_2d(Wi,Wm,nim);

            c = sqrt(gam*V(i,j,4)/V(i,j,1));
            max_c(i,j) = max(abs(V(i,j,2)-c),abs(V(i,j,2)+c));

            dtL = 0.95*(area(i,j))/max_c(i,j);

            W(i,j,:) = Wi - (dtL/area(i,j))*(Fij + Fik + Fil + Fim);
            V(i,j,:) = W_to_V(W(i,j,:));
            F(i,j,:) = Fij + Fik + Fil + Fim;

        end
    end

    % Left Far Field Boundary
    for i = 1
        for j = 2:ny-1

            Wi = [W_old(i,j,1);W_old(i,j,2);W_old(i,j,3);W_old(i,j,4)];

            Wj = W_old(i+1,j,:);
            Wk = W_old(i,j+1,:);
            Wm = W_old(i,j-1,:);

            nij = edge(i,j,1)*[normx(i,j,1);normy(i,j,1)];
            nik = edge(i,j,2)*[normx(i,j,2);normy(i,j,2)];
            nim = edge(i,j,4)*[normx(i,j,4);normy(i,j,4)];

            ni_inf = edge(i,j,3)*[normx(i,j,3);normy(i,j,3)];

            Fij = roe_solver_2d(Wi,Wj,nij);
```

```
        Fik = roe_solver_2d(Wi,Wk,nik);
        Fim = roe_solver_2d(Wi,Wm,nim);

        Fi_inf = roe_solver_2d(Wi,WL,ni_inf);

        c = sqrt(gam*V(i,j,4)/V(i,j,1));
        max_c(i,j) = max(abs(V(i,j,2)-c),abs(V(i,j,2)+c));

        dtL = 0.95*(area(i,j))/max_c(i,j);

        W(i,j,:) = Wi - (dtL/area(i,j))*(Fij + Fik + Fim + Fi_inf);
        V(i,j,:) = W_to_V(W(i,j,:));
        F(i,j,:) = (Fij + Fik + Fim + Fi_inf);



    end
end

% Right Far Field Boundary
for i = nx
    for j = 2:ny-1

        Wi = [W_old(i,j,1);W_old(i,j,2);W_old(i,j,3);W_old(i,j,4)];

        Wk = W_old(i,j+1,:);
        Wl = W_old(i-1,j,:);
        Wm = W_old(i,j-1,:);

        ni_inf = edge(i,j,1)*[normx(i,j,1);normy(i,j,1)];

        nik = edge(i,j,2)*[normx(i,j,2);normy(i,j,2)];
        nil = edge(i,j,3)*[normx(i,j,3);normy(i,j,3)];
        nim = edge(i,j,4)*[normx(i,j,4);normy(i,j,4)];

        Fik = roe_solver_2d(Wi,Wk,nik);
        Fil = roe_solver_2d(Wi,Wl,nil);
        Fim = roe_solver_2d(Wi,Wm,nim);

        Fi_inf = steger_warming_flux(W_to_V(Wi), ni_inf);

        c = sqrt(gam*V(i,j,4)/V(i,j,1));
        max_c(i,j) = max(abs(V(i,j,2)-c),abs(V(i,j,2)+c));

        dtL = 0.95*(area(i,j))/max_c(i,j);

        W(i,j,:) = Wi - (dtL/area(i,j))*(Fik + Fil + Fim + Fi_inf);
        V(i,j,:) = W_to_V(W(i,j,:));
        F(i,j,:) = (Fik + Fil + Fim + Fi_inf);

    end
end

% Bottom Wall Boundary
for i = 2:nx-1
    for j = 1

        Wi = [W_old(i,j,1);W_old(i,j,2);W_old(i,j,3);W_old(i,j,4)];
```

```
            Wj = W_old(i+1,j,:);
            Wl = W_old(i-1,j,:);
            Wk = W_old(i,j+1,:);

            nij = edge(i,j,1)*[normx(i,j,1);normy(i,j,1)];
            nil = edge(i,j,3)*[normx(i,j,3);normy(i,j,3)];
            nik = edge(i,j,2)*[normx(i,j,2);normy(i,j,2)];

            ni_wall = edge(i,j,4)*[normx(i,j,4);normy(i,j,4)];

            Fij = roe_solver_2d(Wi,Wj,0.5*nij);
            Fil = roe_solver_2d(Wi,Wl,0.5*nil);
            Fik = roe_solver_2d(Wi,Wk,0.5*nik);

            Fi_wall = wall_flux(Wi,ni_wall);

            c = sqrt(gam*V(i,j,4)/V(i,j,1));
            max_c(i,j) = max(abs(V(i,j,2)-c),abs(V(i,j,2)+c));

            dtL = 0.95*(area(i,j))/max_c(i,j);

            W(i,j,:) = Wi - (dtL/area(i,j))*(Fij + Fil + Fik+ Fi_wall);
            V(i,j,:) = W_to_V(W(i,j,:));
            F(i,j,:) = (Fij + Fil + Fik+ Fi_wall);

        end
    end

    % Top Wall Boundary
    for i = 2:nx-1
        for j = ny

            Wi = [W_old(i,j,1);W_old(i,j,2);W_old(i,j,3);W_old(i,j,4)];

            Wj = W_old(i+1,j,:);
            Wl = W_old(i-1,j,:);
            Wm = W_old(i,j-1,:);

            nij = edge(i,j,1)*[normx(i,j,1);normy(i,j,1)];
            nil = edge(i,j,3)*[normx(i,j,3);normy(i,j,3)];
            nim = edge(i,j,4)*[normx(i,j,4);normy(i,j,4)];

            ni_wall = edge(i,j,2)*[normx(i,j,2);normy(i,j,2)];

            Fij = roe_solver_2d(Wi,Wj,nij);
            Fil = roe_solver_2d(Wi,Wl,nil);
            Fim = roe_solver_2d(Wi,Wm,nim);

            Fi_wall = wall_flux(Wi,ni_wall);

            c = sqrt(gam*V(i,j,4)/V(i,j,1));
            max_c(i,j) = max(abs(V(i,j,2)-c),abs(V(i,j,2)+c));

            dtL = 0.95*(area(i,j))/max_c(i,j);

            W(i,j,:) = Wi - (dtL/area(i,j))*(Fij + Fil + Fim+ Fi_wall);
            V(i,j,:) = W_to_V(W(i,j,:));
```

```
        F(i,j,:) = (Fij + Fil + Fim+ Fi_wall);

    end
end

% Bottom Right Corner
for i = nx
    for j = 1

        Wi = [W_old(i,j,1);W_old(i,j,2);W_old(i,j,3);W_old(i,j,4)];

        Wl = W_old(i-1,j,:);
        Wk = W_old(i,j+1,:);

        W(i,j,:) = 0.5*(Wl + Wk);

        V(i,j,:) = W_to_V(W(i,j,:));

    end
end

% Top Right Corner
for i = nx
    for j = ny

        Wi = [W_old(i,j,1);W_old(i,j,2);W_old(i,j,3);W_old(i,j,4)];

        Wl = W_old(i-1,j,:);
        Wm = W_old(i,j-1,:);

        W(i,j,:) = 0.5*(Wl + Wm);

        V(i,j,:) = W_to_V(W(i,j,:));

    end
end

% Bottom Left Corner
for i = 1
    for j = 1

        Wi = [W_old(i,j,1);W_old(i,j,2);W_old(i,j,3);W_old(i,j,4)];

        Wj = W_old(i+1,j,:);
        Wk = W_old(i,j+1,:);

        W(i,j,:) = 0.5*(Wj + Wk);
        V(i,j,:) = W_to_V(W(i,j,:));

    end
end

% Top Left Corner
for i = 1
    for j = ny

        Wi = [W_old(i,j,1);W_old(i,j,2);W_old(i,j,3);W_old(i,j,4)];
```

```
            Wj = W_old(i+1,j,:);
            Wm = W_old(i,j-1,:);

            W(i,j,:) = 0.5*(Wj + Wm);
            V(i,j,:) = W_to_V(W(i,j,:));

        end
    end


    W_old = W;
    V_old = V;

    index = index+1;
    Res(index,:) = [norm(norm(F(:,:,1))),norm(norm(F(:,:,2))),norm(norm(F(:,:,3))),norm↙
(norm(F(:,:,4)))];

    time = time + dtG;

    dif1 = abs(((Res(end-1,1)-Res(end,1)))/Res(end,1));
    dif2 = abs(((Res(end-1,2)-Res(end,2)))/Res(end,2));
    dif3 = abs(((Res(end-1,3)-Res(end,3)))/Res(end,3));
    dif4 = abs(((Res(end-1,4)-Res(end,4)))/Res(end,4));

    dif(index,:) = [dif1,dif2,dif3,dif4];
    disp(dif(index,:))

    if dif1 < 1e-4 && dif2 < 1e-4 && dif3 < 1e-4 && dif4 < 1e-4
        break
    end

end

plotter(V,xc,yc)

figure
plot(dif(200:end,:))
set(gca, 'YScale', 'log')
grid on
title_text=('Residuals');
title(title_text,'interpreter','latex','Fontsize',14);
xlabel('Iterations','interpreter','latex','Fontsize',14)
ylabel('Residual','interpreter','latex','Fontsize',14);
legend('Continuity','X-Mom', 'Y-Mom','Energy')
```