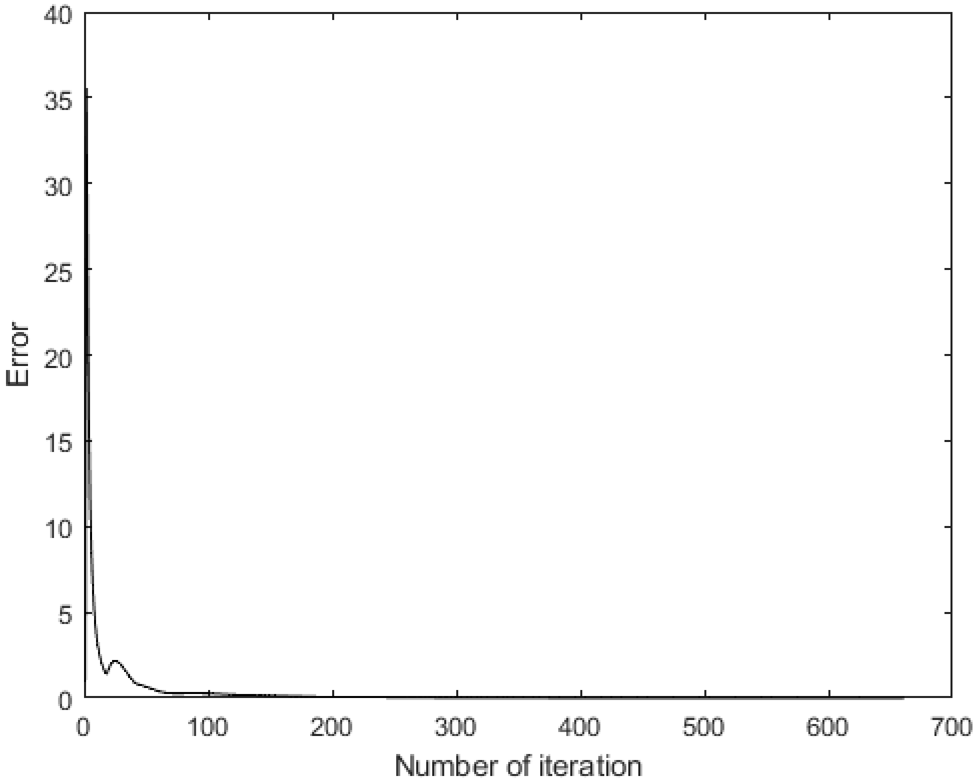
```matlab
% Shivam_Swarnakar_184106011_ME704
% MATLAB code for Stream-function vorticity method-based solution of
 2D incompressible flow

clc;
clear;

imax = 101;
jmax = 41;

Re = 100;
L = 1;
H = L/2;
a = 0.25;
nu = H/Re;

delx = L/(imax-1);
dely = H/(jmax-1);

L_len = linspace(0,L,imax);
H_len = linspace(0,H,jmax);

for i = 1:size(L_len,2)
    if (L_len(1,i) < (L/2+a/2))
        iRight = i;
    end
    if (L_len(1,i) < (L/2-a/2))
        iLeft = i;
    end
end

for i = 1:size(H_len,2)
    if (H_len(1,i) < (H/2+a/2))
        iTop = i;
    end
    if (H_len(1,i) < (H/2-a/2))
        iBottom = i;
    end
end

si(1:jmax,1:imax) = 0;
omega(1:jmax,1:imax) = 0;
u(1:jmax,1:imax) = 0;
v(1:jmax,1:imax) = 0;

si_old = si;
omega_old = omega;
u_old = u;
v_old = v;

u_0 = 1;
u(jmax,:) = u_0;
```

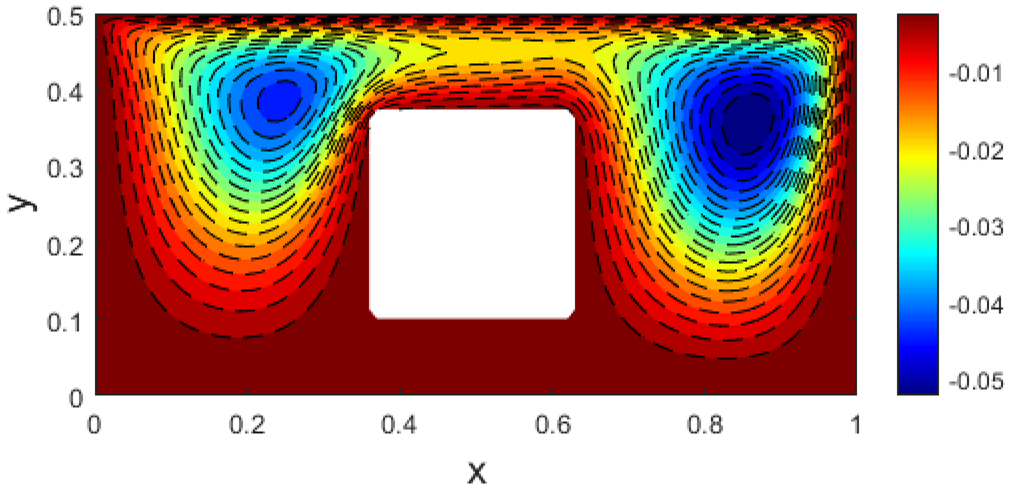
```
eps = 0.001;


si_relax = 1;
omega_relax = 1;

flag(1:jmax,1:imax) = 0;
flag(iBottom:iTop,iLeft:iRight) = 1;

n = 1;
err(1:n) = 1;
while(abs(err(end)) >= eps)

    for j = 2:jmax-1
        for i = 2:imax-1
            if (flag(j,i) == 0)
                u(j,i) = (si(j+1,i)-si(j-1,i))/(2*dely);
                v(j,i) = -(si(j,i+1)-si(j,i-1))/(2*delx);

                si(j,i) = 0.25*(si(j+1,i)+si(j-1,i)+si(j,i
+1)+si(j,i-1)+(delx^2)*omega(j,i));
                si(j,i) = si_old(j,i) + si_relax*(si(j,i)-
si_old(j,i));

                omega(jmax,:) = -(2*u_0*dely + 2*si(jmax-1,:))/
(dely^2); %TOP BC
                omega(1,:) =  -2*(si(2,:)/(dely^2)); %BOTTOM BC
                omega(:,imax) = -(2*si(:,imax-1))/(delx^2); %LEFT BC
                omega(:,1) = -(2*si(:,2)/(delx^2)); %RIGHT BC

                % inner square cylinder BC

                omega(iBottom:iTop,iLeft) = -
(2*si(iBottom:iTop,iLeft-1))/(delx^2); %LEFT BC
                omega(iBottom:iTop,iRight) = -
(2*si(iBottom:iTop,iRight+1)/(delx^2)); %RIGHT BC
                omega(iBottom,iLeft:iRight) =
 -2*(si(iBottom-1,iLeft:iRight)/(dely^2)); %BOTTOM BC
                omega(iTop,iLeft:iRight) = -2*(si(iTop
+1,iLeft:iRight)/(dely^2)); %BOTTOM BC

                aE = 1 - ((min(u(j,i),0)*delx)/nu);
                aW = 1 + ((max(u(j,i),0)*delx)/nu);
                aN = 1 - ((min(v(j,i),0)*dely)/nu);
                aS = 1 + ((max(v(j,i),0)*dely)/nu);
                aP = aE+aW+aN+aS;

                omega(j,i) = (aE*omega(j,i+1) + aW*omega(j,i-1) +
 aN*omega(j+1,i) + aS*omega(j-1,i))/aP;
                omega(j,i) = omega_old(j,i) + omega_relax*(omega(j,i)-
omega_old(j,i));
            else
                continue;
            end
        end
    end

    error_si = si -si_old;
    err_si = max(max(error_si));

    error_omega = omega -omega_old;
    err_omega = max(max(error_omega));

    error_u = u -u_old;
    err_u = max(max(error_u));

    error_v = v -v_old;
    err_v = max(max(error_v));

    u_old = u;
    v_old = v;
    si_old = si;
    omega_old = omega;

    err1 = max(err_u,err_v);
    err2 = max(err_omega,err_si);
    err(n) = max(err1,err2);
    plot(err,'-k','LineWidth',1)
    xlabel('Number of iteration');
    ylabel('Error');
    drawnow;
    n = n+1;

    if (n > 5000)
        break;
    else
        continue;
    end

end
```

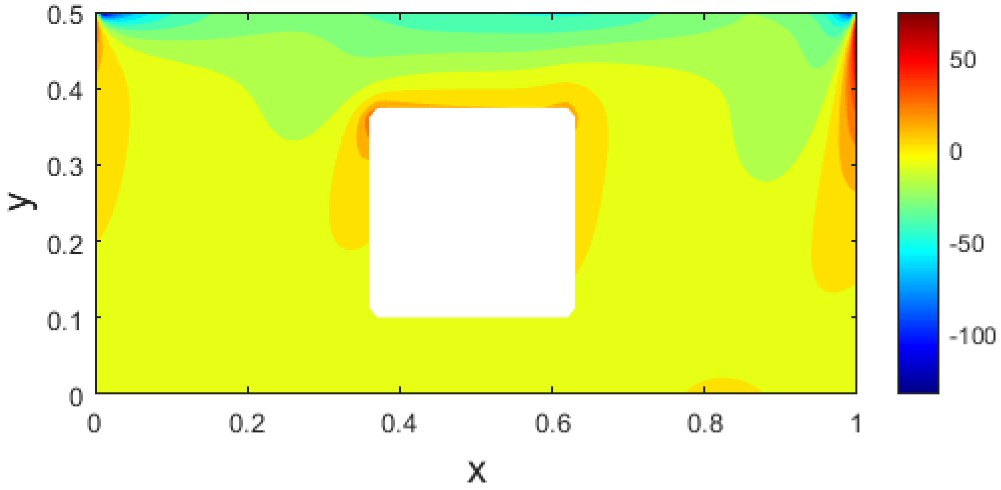


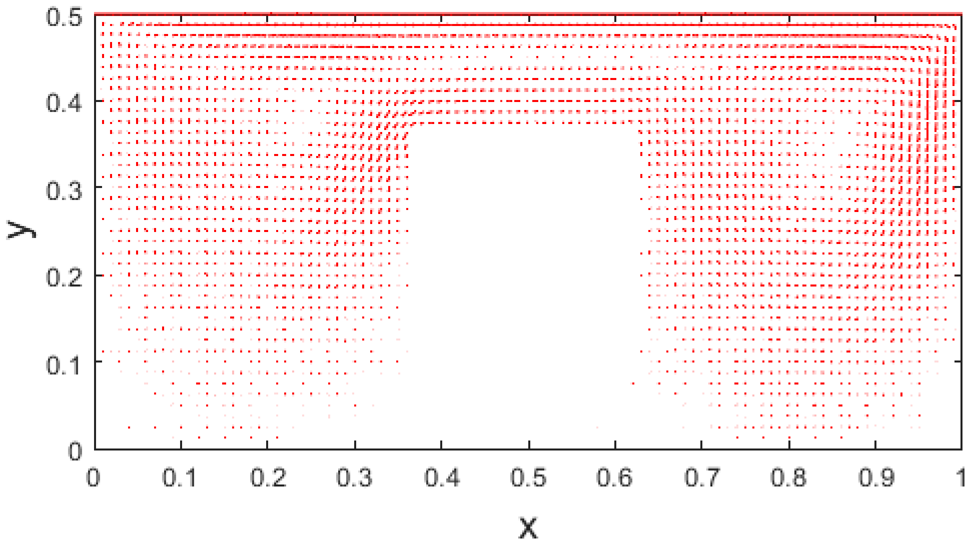
Plotting the data

```
for j = iBottom:iTop
    for i = iLeft:iRight
        si(j,i) = NaN;
        u(j,i) = NaN;
        v(j,i) = NaN;
        omega(j,i) = NaN;
    end
end
figure;
contourf(L_len,H_len,si,20,'LineStyle','--');
colormap('jet');
colorbar;
xlabel('x','FontSize',15);
ylabel('y','FontSize',15);
axis equal;
name = ('si_contour_plot.png');
saveas(gcf,name);
figure;
contourf(L_len,H_len,omega,20,'LineStyle','none');
colormap('jet');
colorbar;
xlabel('x','FontSize',15);
ylabel('y','FontSize',15);
```

```
axis equal;
name = ('omega_contour_plot.png');
saveas(gcf,name);
figure;
quiver(L_len,H_len,u,v,'LineWidth',1,'Color','r');
axis equal;
xlabel('x','FontSize',15);
ylabel('y','FontSize',15);
xlim([0,L]);
ylim([0,H]);
name = ('Vector_plot_contour_plot.png');
saveas(gcf,name);
```




0.5
0.4
0.3
0.2
0.1
0
y
0
0.2
0.4
0.6
0.8
1
x
50
0
-50
-100


0.5
0.4
0.3
0.2
0.1
0
y
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x

Published with MATLAB® R2020b