```
clearvars
clc
close all

% Method for finite difference operators
fdmethod = 'compact';

% Uniform mesh in transformed plane
nxi1 = 40;
nxi2 = 40;

xi1 = 0:1:nxi1;
xi2 = 0:1:nxi2;

x = zeros(nxi1+1,nxi2+1);
y = zeros(nxi1+1,nxi2+1);

% Define x and y on boundaries
%- bottom
x(:,1) = (xi1./xi1(end)).*4;
y(x(:,1)<=1,1) = 0;
y(((x(:,1)>=1)&(x(:,1)<=2)),1) = x(((x(:,1)>=1)&(x(:,1)<=2)),1)-1;
y(((x(:,1)>=2)&(x(:,1)<=3)),1) = 3-x(((x(:,1)>=2)&(x(:,1)<=3)),1);
y(x(:,1)>=3,1) = 0;

%- top
x(:,end) = (xi1./xi1(end)).*4;
y(:,end) = 2*ones(nxi1+1,1);

%- west
x(1,:) = zeros(1,nxi2+1);
y(1,:) = (xi2./xi2(end)).*2;

%- east
x(end,:) = 4*ones(1,nxi2+1);
y(end,:) = (xi2./xi2(end)).*2;

% Plot contour
msk = ones(nxi1+1,nxi2+1); msk(2:end-1,2:end-1) = 0;
xx = x(msk==1); yy = y(msk==1);

figure
hh=plot(xx,yy,'r.');
axis equal
grid on

% Initial condition
dx = (max(x(msk==1))-min(x(msk==1)))/nxi1;
x(2:end-1,2:end-1) = min(x(msk==1)) + dx*repmat([1:nxi1-1].',[1,nxi2-1]);
dy = (max(y(msk==1))-min(y(msk==1)))/nxi2;
y(2:end-1,2:end-1) = min(y(msk==1)) + dy*repmat(1:nxi2-1,[nxi1-1,1]);


%- Save position of boundary points in different form for convenience
xb = x(:,1); yb = y(:,1);
xt = x(:,end); yt = y(:,end);
xw = x(1,:); yw = y(1,:);
xe = x(end,:); ye = y(end,:);

options = optimoptions('fsolve','Display','iter','MaxFunctionEvaluations',
1000000,'OptimalityTolerance',1.0000e-13);
[xy,fval,exitflag] =
fsolve(@(xy)residual(xy,xb,yb,xt,yt,xw,yw,xe,ye,nxi1,nxi2,fdmethod),[x(:);y(:)],option
s);
N = length(xy);
x = reshape(xy(1:N/2),[nxi1+1,nxi2+1]);
y = reshape(xy(N/2+1:end),[nxi1+1,nxi2+1]);

figure
hold on
plot(x(:),y(:),'ro');
axis equal
for i=1:nxi1+1
```

```
71      h=plot(x(i,:),y(i,:),'b-');
72  end
73  for j=1:nxi2+1
74      h=plot(x(:,j),y(:,j),'b-');
75  end
76
77  %{
78    Mesh aspect ratio
79  %}
80  a = sqrt((x(3:end,2:end-1)-x(1:end-2,2:end-1)).^2 +
    (y(3:end,2:end-1)-y(1:end-2,2:end-1)).^2);
81  b = sqrt((x(2:end-1,3:end)-x(2:end-1,1:end-2)).^2 +
    (y(2:end-1,3:end)-y(2:end-1,1:end-2)).^2);
82  AR = b./a;
83
84  figure
85  [cs,h] =
    contourf(x(2:end-1,2:end-1),y(2:end-1,2:end-1),AR,min(AR(:)):(max(AR(:))-min(AR(:)))/3
    0:max(AR(:)));
86  set(h,'edgecolor','none');
87  shading interp
88  axis equal
89  colorbar
90
91  %{
92    Mesh skewness (mesh non-orthogonality)
93  %}
94  vxx = x(3:end,2:end-1)-x(1:end-2,2:end-1);
95  vyx = y(3:end,2:end-1)-y(1:end-2,2:end-1);
96  vxy = x(2:end-1,3:end)-x(2:end-1,1:end-2);
97  vyy = y(2:end-1,3:end)-y(2:end-1,1:end-2);
98  C   = acosd((vxx.*vxy + vyx.*vyy)./sqrt((vxx.^2+vyx.^2).*(vxy.^2+vyy.^2)));
99  Skw = abs(C-90)./90;
100
101 figure
102 [cs,h] =
    contourf(x(2:end-1,2:end-1),y(2:end-1,2:end-1),Skw,min(Skw(:)):(max(Skw(:))-min(Skw(:)
    ))/30:max(Skw(:)));
103 set(h,'edgecolor','none');
104 shading interp
105 axis equal
106 colorbar
107
108
109 % ************************************************************************** %
110 % ************************************************************************** %
111
112 function R = residual(xy,xb,yb,xt,yt,xw,yw,xe,ye,nxi1,nxi2,fdmethod)
113
114 N = length(xy);
115 x = reshape(xy(1:N/2),[nxi1+1,nxi2+1]);
116 y = reshape(xy(N/2+1:end),[nxi1+1,nxi2+1]);
117
118 % Metrics
119 dxdxi1 = cdiffv(x.',1,fdmethod).';
120 dxdxi2 = cdiffv(x,1,fdmethod);
121 dydxi1 = cdiffv(y.',1,fdmethod).';
122 dydxi2 = cdiffv(y,1,fdmethod);
123
124 D   = dxdxi1.*dydxi2 - dxdxi2.*dydxi1;
125 A   = dxdxi2.^2 + dydxi2.^2;
126 B   = dxdxi1.*dxdxi2 + dydxi1.*dydxi2;
127 C   = dxdxi1.^2 + dydxi1.^2;
128
129 % Second derivatives of x, y: computed either
130 % by applying twice the first derivative or directly
131 % a second derivative
132 if 0
133     d2xdxi1_2 = cdiffv(dxdxi1.',1,fdmethod).';
134     d2ydxi1_2 = cdiffv(dydxi1.',1,fdmethod).';
135     d2xdxi2_2 = cdiffv(dxdxi2,1,fdmethod);
136     d2ydxi2_2 = cdiffv(dydxi2,1,fdmethod);
137 else
```

```
    d2xdxi1_2 = cdiffd2v(x.',1,fdmethod).';
    d2ydxi1_2 = cdiffd2v(y.',1,fdmethod).';
    d2xdxi2_2 = cdiffd2v(x,1,fdmethod);
    d2ydxi2_2 = cdiffd2v(y,1,fdmethod);
end

% Mixed derivatives of x, y
d2xdxi1dxi2 = 0.5*(cdiffv(dxdxi1,1,fdmethod) + cdiffv(dxdxi2.',1,fdmethod).');
d2ydxi1dxi2 = 0.5*(cdiffv(dydxi1,1,fdmethod) + cdiffv(dydxi2.',1,fdmethod).');

Rx = A.*d2xdxi1_2 -2*B.*d2xdxi1dxi2 + C.*d2xdxi2_2;

Ry = A.*d2ydxi1_2 -2*B.*d2ydxi1dxi2 + C.*d2ydxi2_2;

Rx(:,1) = x(:,1)-xb;
Ry(:,1) = y(:,1)-yb;

Rx(:,end) = x(:,end)-xt;
Ry(:,end) = y(:,end)-yt;

Rx(1,:) = x(1,:)-xw;
Ry(1,:) = y(1,:)-yw;

Rx(end,:) = x(end,:)-xe;
Ry(end,:) = y(end,:)-ye;

R = [Rx(:);Ry(:)];

return;

end

% ********************************************************************************** %
% ********************************************************************************** %

%{
clear all
clc
close all

N = 10;
h = 1/(N-1);
x = 0:h:1;
f = exp(-x.^7); df = -7*x.^6.*f;
delf=cdiff(h,f);
figure
hf = plot(x,df,'b-');
hold on
hf = plot(x,delf,'rx');
hf = plot(x,[f(2)-f(1);(f(3:end).'-f(1:end-2).')./2;f(end)-f(end-1)]./h,'g--');
%}
function df=cdiffv(f,h,method)

N = size(f,1);

switch method
    case{'CD2'}
        df = zeros(size(f));
        df(2:end-1,:) = (f(3:end,:)-f(1:end-2,:))./(2*h);
        df(1,:)       = (f(2,:)-f(1,:))./h;
        df(N,:)       = (f(N,:)-f(N-1,:))./h;
    otherwise
        d = ones(N,1);
        u = [1;0.25*ones(N-2,1);1];
        A = spdiags([[u(2:end);1],d,[1;u(1:end-1)]],[-1,0,1],N,N);

        d = [-2;zeros(N-2,1);2];
        u = [2;0.75*ones(N-2,1);2];
        l = [2;-0.75*ones(N-2,1);-2];
        B = spdiags([[l(2:end);1],d,[1;u(1:end-1)]],[-1,0,1],N,N)./h;

        df = A\(B*f);
end
```

```
211
212   return
213
214   end
215
216   % ************************************************************************** %
217   % ************************************************************************** %
218
219   %{
220   clear all
221   clc
222   close all
223
224   N = 100;
225   h = 1/(N-1);
226   x = 0:h:1;
227   f = exp(-x.^7); ddf = 7*x.^5.*exp(-x.^7).*(7*x.^7 - 6);
228   del2f=cdiffd2v(h,f(:));
229   figure
230   hf = plot(x(:),ddf(:),'b-');
231   hold on
232   hf = plot(x(:),del2f(:),'rx');
233   %}
234   function df=cdiffd2v(f,h,method)
235
236   switch method
237
238       otherwise
239
240           N = size(f,1);
241           d = ones(N,1);
242           u = [11;0.1*ones(N-2,1);11];
243           A = spdiags([[u(2:end);1],d,[1;u(1:end-1)]],[-1,0,1],N,N);
244
245           d = [13;-2.4*ones(N-2,1);13];
246           u = [-27;1.2*ones(N-1,1)];
247           uu = [15;zeros(N-1,1)];
248           uuu = [-1;zeros(N-1,1)];
249           l = [1.2*ones(N-1,1);-27];
250           ll = [zeros(N-1,1);15];
251           lll = [zeros(N-1,1);-1];
252           B =
              spdiags([[lll(4:end);nan;nan;nan],[ll(3:end);nan;nan],[l(2:end);1],d,[1;u(1:en
              d-1)],[nan;nan;uu(1:end-2)],[nan;nan;nan;uuu(1:end-3)]],[-3:3],N,N)./(h^2);
253
254           df = A\(B*f);
255   end
256
257   return
258
259   end
260
261
262
263
```