Computing Deformation Measures from Digital Image Correlation Data using Matlab

Md Mahmudul Hasan Mollah-W10389619
Department of Mechanical Engineering, University of Wyoming

Abstract

Digital Image Correlation (DIC) is a powerful tool to measure material properties (like: stress, strain) without physical measurement of displacement under any unknown stress. It just needs an undeformed image, a deformed image as well as some experimental parameters. Using the Matlab environment for DIC is a convenient way to visualize different results and process the input file. This project will cover the fundamental aspects of DIC using Matlab GUI. It also covers how to make the code ubiquitous for all types of images and corresponding circumstances as well as several validation processes of the code.

1. Introduction

In standard empirical stress-strain measurement, a known force is applied to an undeformed body, and the displacement is measured in a prescribed way. After that, the specimen dimension and other engineering data help to determine the required mechanical properties of the materials. But what if the force is unknown and the displacement is too small that it is hard to measure(infinitesimal)? Here comes the optical process which is called DIC process.

This DIC process takes a small subset from the undeformed image and then tries to find its best matching position in the deformed image. At the same time, it gives a co-relation constant c to every point that it searches for a probable match based on its similarity. This value varies from -1 to +1 where +1 means best matching position and -1 is vice-versa. This small subset is a combination of some pixels. Matlab image processing toolbox expedites this process by defining predefined functions for doing this like 'normxcorr2'. From Matlab documentation of that function, it is clear that the output array of c is bigger than the image original array. So, after some modification, it gives the displaced position of that subset in the deformed image which will help to determine the displacement of that subset. However, this is pixel displacement. Now if we know the pixel size of the camera that took both of these pictures then we can multiply the pixel displacement by the pixel dimensions and get the total displacement. After getting total displacement, it is easier to get the deformation gradient and subsequent Lagrange strain. Matlab GUI helps to plot the displacement and strain of each subset in a plot which is helpful for visualization. So to complete the whole precess, some steps were followed in the Matlab code. Which are:

Step 1: Collecting file

Here, it must be ensured that the data file is in RGB format because matlab Image processing toolbox only collects data from RGB images. However, if the image is not in RGB, there are many matlab function to convert that image to RGB. Both images (undeformed and deformed) must be on same size.

```
R = imread('referenceimage.tif');
C = imread('deformedimage.tif');
```

Step 2: Exclude unnecessary Regions and define ROI

There might be many regions that remain undeformed in the final image. So, at that region displacement is zero. Moreover, there might be black border or a structured or unstructured hole in the reference image which does not contribute to the displacement. So after excluding these regions, we will get the region of interest (ROI). For that here a mask is created for defining these regions which might be vary from picture to picture and it is easy to adjust with different images. That ROI will be used in the future steps to create a meshgrid in the confined ROI region. Here multiple regions might be excluded using the "drawfreehand" function and the final ROI is shown in the terminal.

```
figure;
imshow(R);
title('Select the region to be excluded in the grid');
mask_for_exclude = false(size(R, 1), size(R, 2));

for i = 1:5% the second value might be equal to number of excluded area
    h = drawfreehand('Label', ['Region ' num2str(i)]);
    wait(h);
    mask = createMask(h);
    mask_for_exclude = mask_for_exclude | mask;
end

ROI= ~mask_for_exclude;

figure;
imshow(R);
hold on;
h = imshow(~ROI);
set(h, 'AlphaData', 0.3);
title('Excluded and ROI on reference image');
```

Step 3: Meshgrid & initial variable for Iteration and Ploting

Here subset will be taken only from the ROI. For that a meshgrid will be defined throughout the reference image and only the meshgrid point that fall inside the ROI will be considered. Moreover, for visualization some variable will be declared which will be used in future iteration. Some intial zeros vector will be defined which have the same purpose as like visualization variable. Subset size and grid spacing is a deal braker in this situation since reducing the subset size and grid-spacing creates better results but at the meantime in increases computational cost and time. It might vary based on the picture size and required accuracy.

```
grid_spaceing=20;
[rows,columns] = size(R);
[xGrid, yGrid] = meshgrid(1:grid_spaceing:columns, 1:grid_spaceing:rows);
```

```
figure('Name','Reference Image Grid','NumberTitle','off');
ax = axes;
imshow(R, 'Parent', ax);

figure('Name','Moved subset in deformed image','NumberTitle','off');
ay = axes;
imshow(C, 'Parent', ay);

w=10;
h=10;
R_template_size=[w,h];

x_displacement = zeros(size(xGrid));
y_displacement = zeros(size(yGrid));
final_displacement =zeros(size(xGrid));
```

Step 4: normxcorr2: Displacement

It is time to create a small subset and check if whether it is inside the ROI or not. If is inside ROI then it will be the input of Matlab image processing toolbox. The "normxcorr2" will find the peak value point for the matched position. Then this value will be modified and compare to the middle of the subset to determine the displacement. Now the whole process will be done using a for loop throughout the image. In the meantime, it will draw each subset in the reference image in the same axis. In the current image subset, the initial position and deformed position of the subset will be shown.

```
for i = 1:size(xGrid, 1)
    for j = 1:size(xGrid, 2)
            x=xGrid(i,j);
            y=yGrid(i,j);
            xMin = max(1, x - R_template_size(1)/2);
            xMax = min(columns, x + R_template_size(1)/2 -1);
            yMin = max(1, y - R_template_size(2)/2);
            yMax = min(rows, y + R_template_size(2)/2-1);

         if all(ROI(yMin:yMax, xMin:xMax), 'all')
            R_template = R(yMin:yMax, xMin:xMax);

            c = normxcorr2(R_template, C);
            [ypeak, xpeak] = find(c==max(c(:)));

            yoffSet = ypeak-(size(R_template,1)/2)+1;
            xoffSet = xpeak-(size(R_template,2)/2)+1;

            x_displacement(i, j) = xoffSet - x;
            y_displacement(i, j) = yoffSet - y;

            final_displacement(i,j) =sqrt(x_displacement(i,j).^2
+y_displacement(i,j).^2);
            rectangle(ax, 'Position', [x, y, size(R_template,2), size(R_template,1)],
'EdgeColor', 'g','LineWidth', .5);

            rectangle(ay, 'Position', [xoffSet, yoffSet, size(R_template,2),
size(R_template,1)], 'EdgeColor', 'r','LineWidth', .5);
```

```
            rectangle(ay, 'Position', [x, y, size(R_template,2), size(R_template,1)],
'EdgeColor', 'g','LineWidth', 1);


        end
    end
end
```

Step 5: Displacement Plot

Here, the "countourf" function will be used to plot the x, y, and overall displacement. The overall code is in 2D throughout the image. The maximum and minimum value ranges will be divided into several ranges (trial and error for "levels") for better visualization. There will be no plot when displacement is zero in each point i.e. if no standard deviation. If we change the variable "x_displacement" to y_displacements or "final_displacement" then all plots will be obtained.

```
if std(x_displacement(:) ~= 0)

     figure('Name','xdisplacement contour plot','NumberTitle','off');
     levels = linspace(min(x_displacement(:)), max(x_displacement(:)), 10);
     contourf(xGrid, yGrid, x_displacement, levels, 'LineColor', 'r');
     colorbar;
     title('X Displacement Contour Plot');
     xlabel('X-pixel');
     ylabel('Y -pixel');
     axis equal;
     set(gca, 'YDir', 'reverse');
     hold on;
     plot(xGrid, yGrid, 'g+', 'MarkerSize', .5, 'LineWidth', .5);
     hold off;
else
    disp('x displacement all value are zero or constant no standard Deviation in
data: Contour plot not generated.');
end
```

Step 6: Deformation Gradient & Jacobian matrix

Now the partial derivative of displacement at each point will be derived to get Deformation Gradient. After that, it is also possible to determine the Jacobian matrix. These two matrices will be used to measure the Left and Right Tensor, Rotation Tensor, Spin Tensor, etc. (this will not be covered in this project)

```
[dx_dx, dx_dy]=gradient(x_displacement,grid_spaceing,grid_spaceing);
[dy_dx, dy_dy]=gradient(y_displacement,grid_spaceing,grid_spaceing);
[a,b]=size(xGrid);
F=zeros(2,2,a,b);
for m=1:size(xGrid,1)
    for n=1:size(xGrid,2)
```

```
        F(:,:,a,b)= [dx_dx(i,j), dx_dy(i,j);
                     dy_dx(i,j), dy_dy(i,j)];
    end

end
fprintf('The Deformation Gradient array or Tensor (F) are:');
Tensor_F= F


J=dx_dx.*dy_dy- dx_dy.*dy_dx;
```

Step9: Lagrange strain & corresponding plot

Now, It is easy to get the different Lagrange strain and their corresponding plot to visualize the strain.

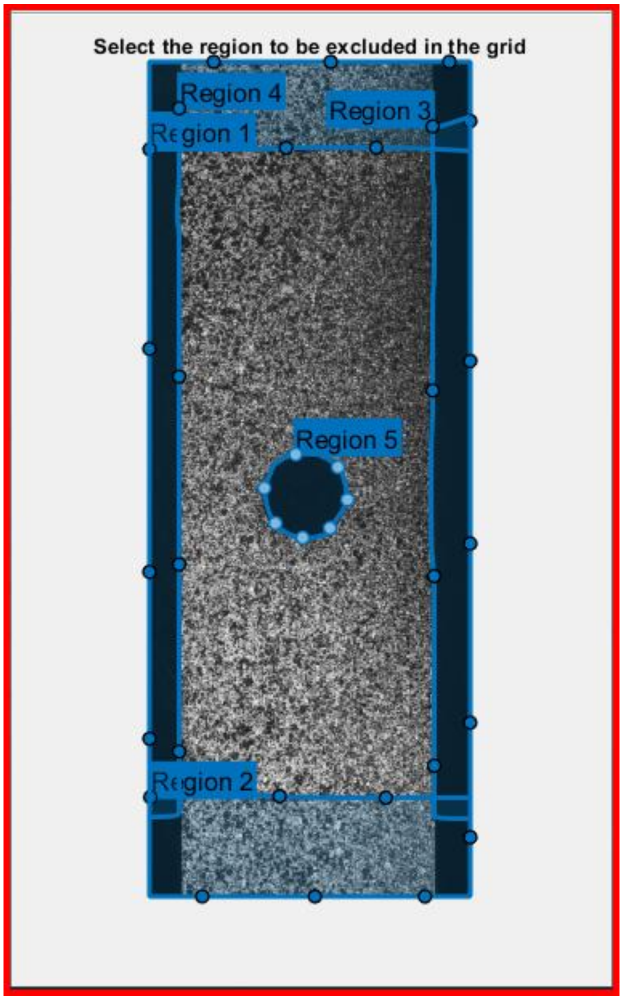
```
Exx=dx_dx+ 0.5*(dx_dx.^2+dy_dy.^2);
Eyy = dy_dy + 0.5*(dx_dy.^2 + dy_dy.^2);
Exy = 0.5*(dx_dy + dy_dx) + 0.5*(dx_dx.*dx_dy + dy_dx.*dy_dy);
```

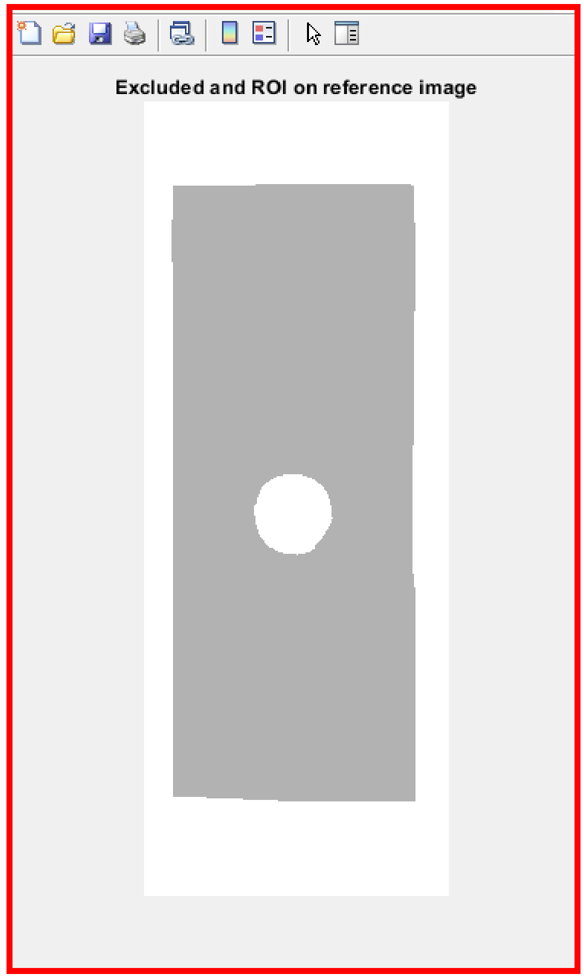
Plotting the Exx, Eyy, Exy is the same except the variable Exx will be changed to Eyy or Exy when drawing the corresponding plot. Sometimes the “levels” number might be adjusted based on the maximum and minimum value of the output strain observed in the workspace terminal. If there is no stain or no standard deviation in the date there will be no plot

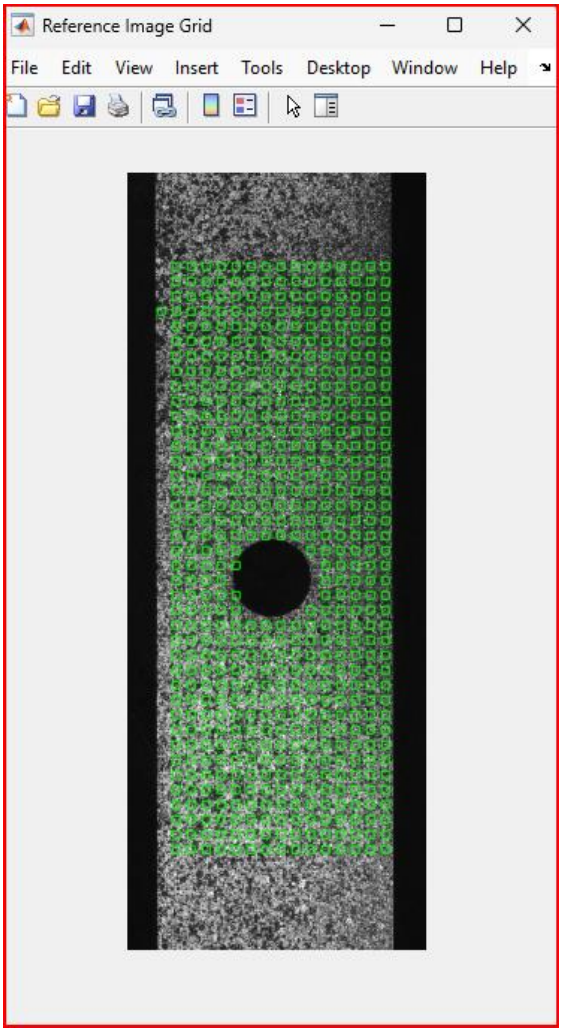
```
if std(Exx(:) ~= 0)

     figure('Name','Exx contour plot','NumberTitle','off');
     levels = linspace(min(Exx(:)), max(Exx(:)), 5);
     contourf(xGrid, yGrid, Exx, levels, 'LineColor', 'r');
     colorbar;
     title('Exx Contour Plot');
     xlabel('X-pixel');
     ylabel('Y -pixel');
     axis equal; % original size
     set(gca, 'YDir', 'reverse');
     hold on;
     plot(xGrid, yGrid, 'g+', 'MarkerSize', .5, 'LineWidth', .5);
     hold off;
else
    disp('Exx all value is constant or zeros no standard deviation: Contour plot not
generated.');
end
```

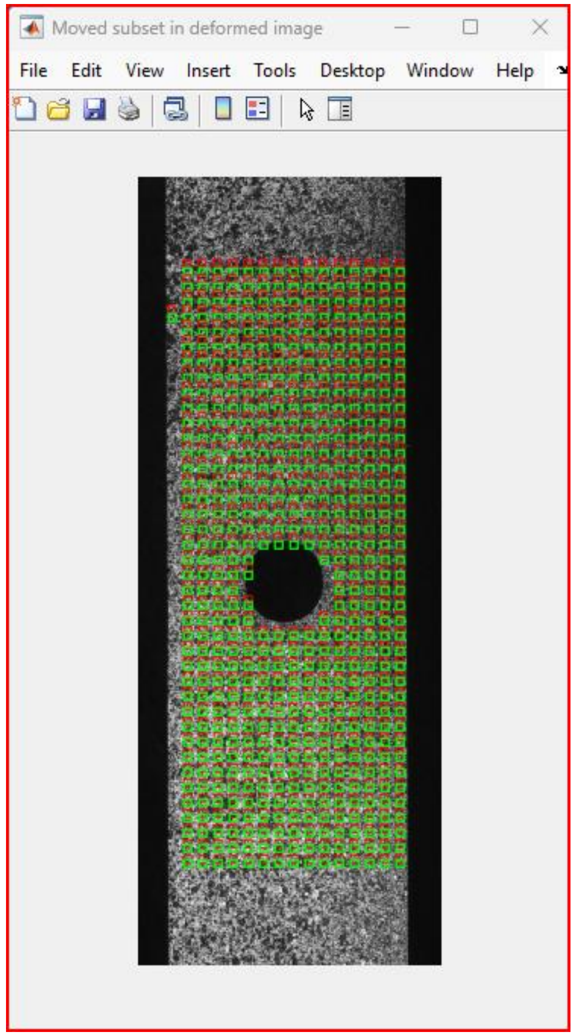
2. Application of Code

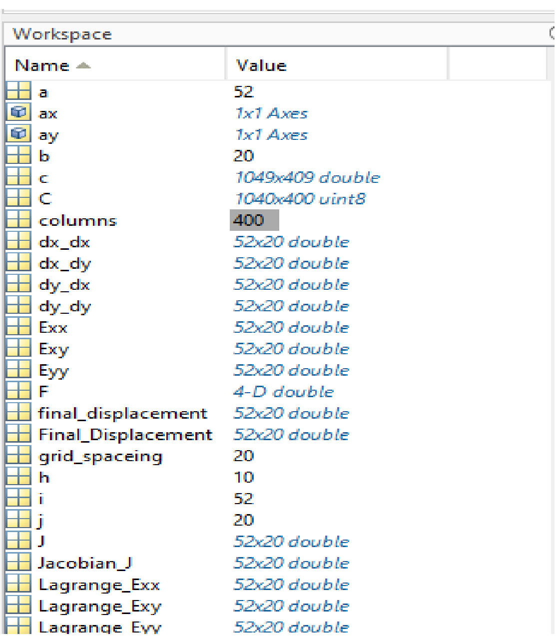
This code output for an open-hole image example (collected from www.ncorr.com) is given below. The full details code is provided in the Appendix. If image descriptions are given at the top of the image no description will be provided at the down of the image.





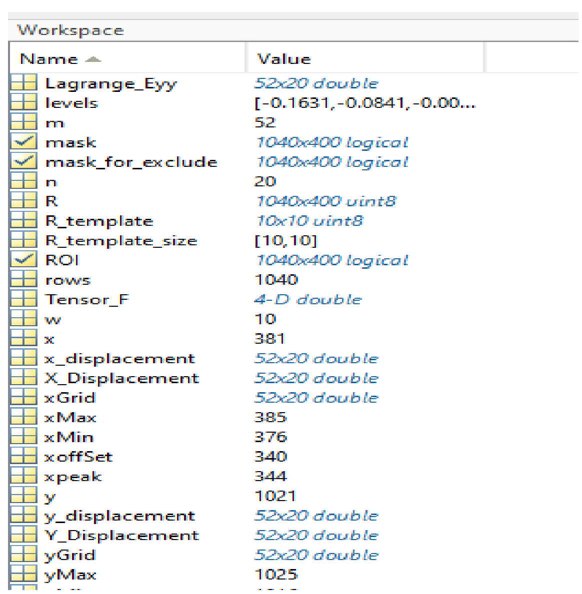






Workspace

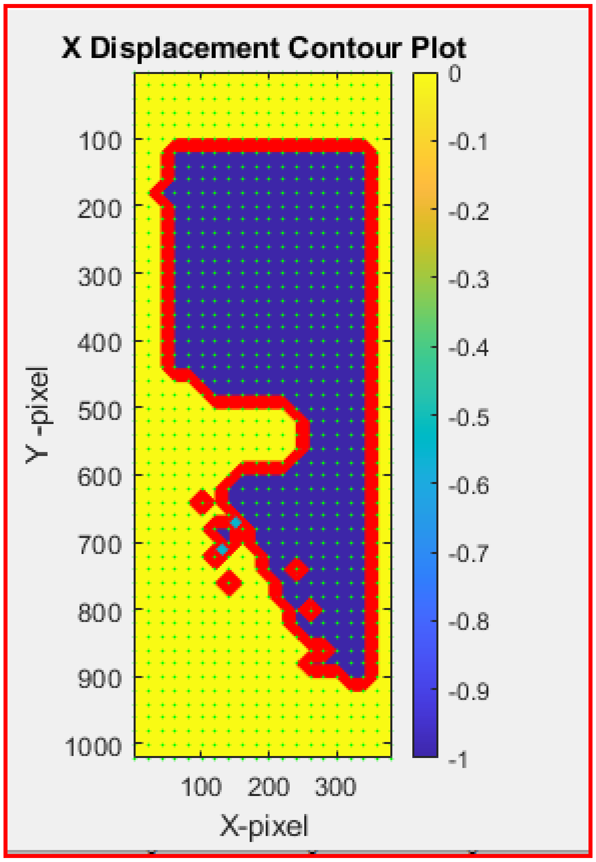
Name	Value
a	52
ax	1x1 Axes
ay	1x1 Axes
b	20
c	1049x409 double
C	1040x400 uint8
columns	400
dx_dx	52x20 double
dx_dy	52x20 double
dy_dx	52x20 double
dy_dy	52x20 double
Exx	52x20 double
Exy	52x20 double
Eyy	52x20 double
F	4-D double
final_displacement	52x20 double
Final_Displacement	52x20 double
grid_spaceing	20
h	10
i	52
j	20
J	52x20 double
Jacobian_J	52x20 double
Lagrange_Exx	52x20 double
Lagrange_Exy	52x20 double
Lagrange_Eyy	52x20 double

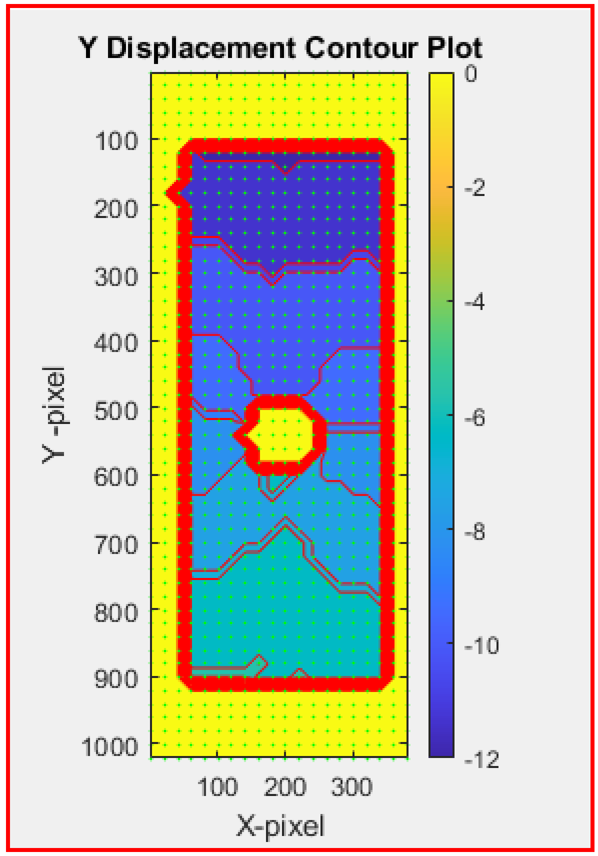


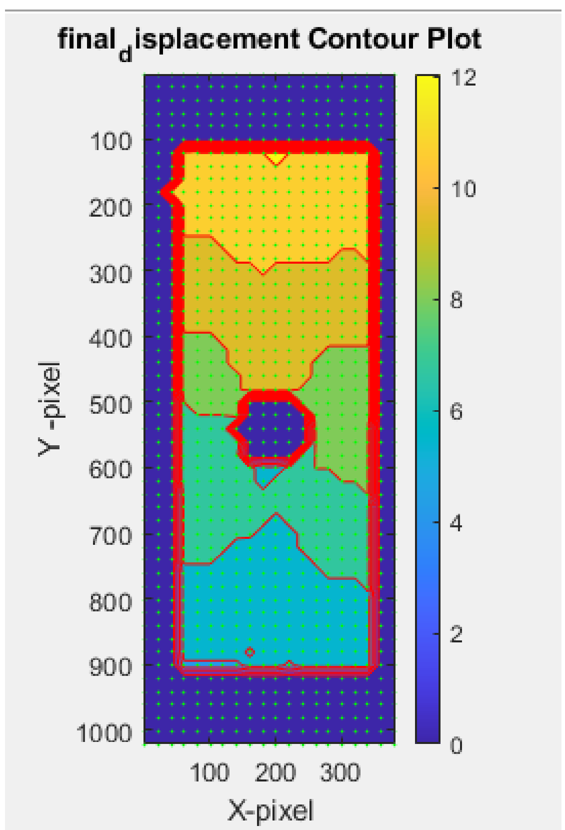
Workspace

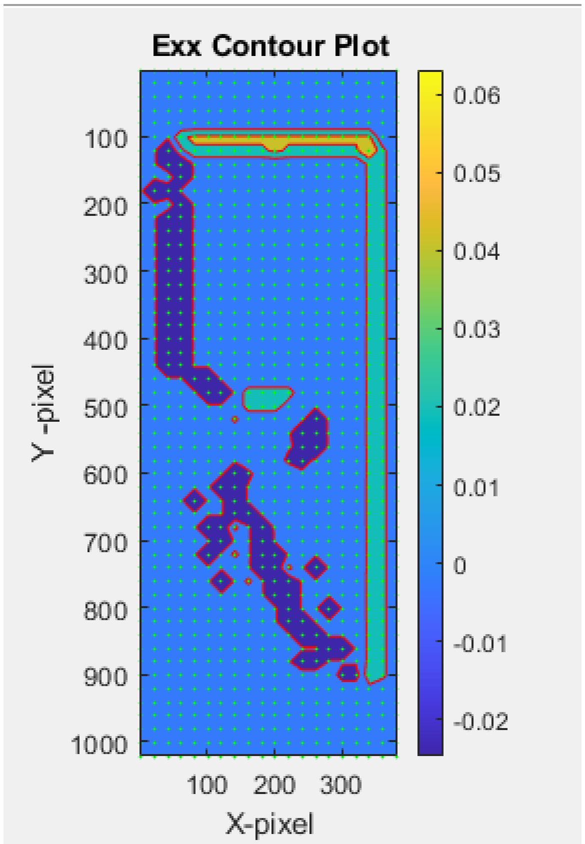
Name	Value
Lagrange_Eyy	52x20 double
levels	[-0.1631,-0.0841,-0.00...
m	52
mask	1040x400 logical
mask_for_exclude	1040x400 logical
n	20
R	1040x400 uint8
R_template	10x10 uint8
R_template_size	[10,10]
ROI	1040x400 logical
rows	1040
Tensor_F	4-D double
w	10
x	381
x_displacement	52x20 double
X_Displacement	52x20 double
xGrid	52x20 double
xMax	385
xMin	376
xoffSet	340
xpeak	344
y	1021
y_displacement	52x20 double
Y_Displacement	52x20 double
yGrid	52x20 double
yMax	1025

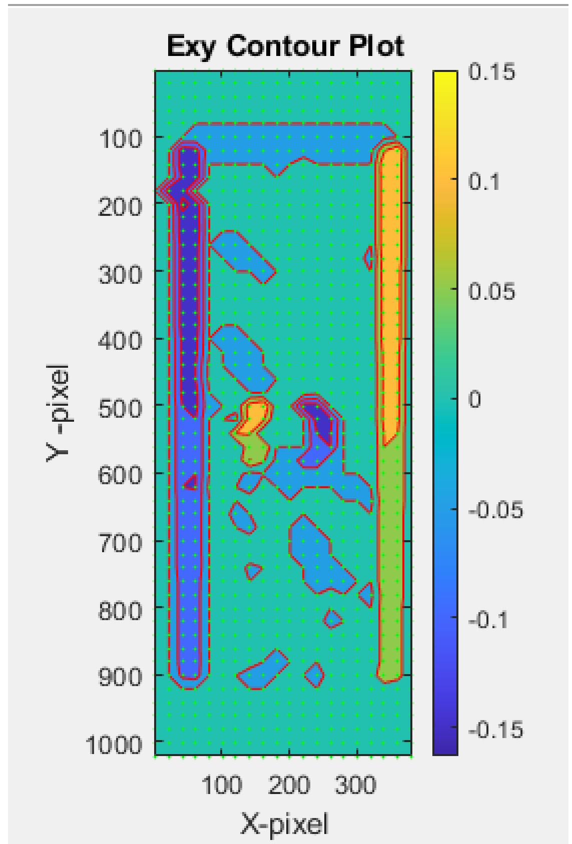
Figure: All workspace variable

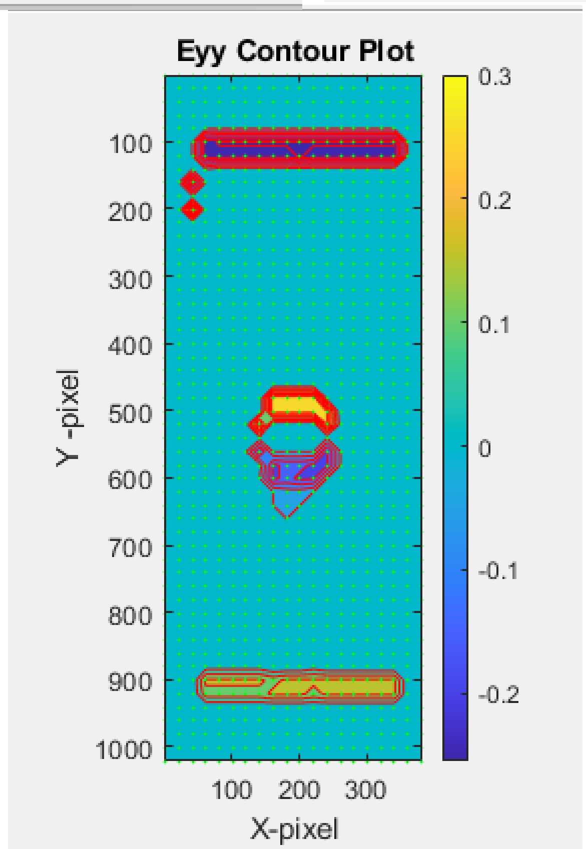

X Displacement Contour Plot
Y -pixel
X-pixel


Y Displacement Contour Plot
Y -pixel
X-pixel


final_displacement Contour Plot
Y -pixel
X-pixel


Exx Contour Plot
Y -pixel
100
200
300
400
500
600
700
800
900
1000
X-pixel
100 200 300
0.06
0.05
0.04
0.03
0.02
0.01
0
-0.01
-0.02


Exy Contour Plot
Y -pixel
100
200
300
400
500
600
700
800
900
1000
X-pixel
100 200 300
0.15
0.1
0.05
0
-0.05
-0.1
-0.15


Eyy Contour Plot
Y -pixel
100
200
300
400
500
600
700
800
900
1000
X-pixel
100 200 300
0.3
0.2
0.1
0
-0.1
-0.2

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	-1	-1	-1	-1	-1	-1	-1
8	0	0	0	-1	-1	-1	-1	-1	-1	-1
9	0	0	0	-1	-1	-1	-1	-1	-1	-1
10	0	0	-1	-1	-1	-1	-1	-1	-1	-1
11	0	0	0	-1	-1	-1	-1	-1	-1	-1
12	0	0	0	-1	-1	-1	-1	-1	-1	-1
13	0	0	0	-1	-1	-1	-1	-1	-1	-1
14	0	0	0	-1	-1	-1	-1	-1	-1	-1
15	0	0	0	-1	-1	-1	-1	-1	-1	-1
16	0	0	0	-1	-1	-1	-1	-1	-1	-1
17	0	0	0	-1	-1	-1	-1	-1	-1	-1
18	0	0	0	-1	-1	-1	-1	-1	-1	-1
19	0	0	0	-1	-1	-1	-1	-1	-1	-1
20	0	0	0	-1	-1	-1	-1	-1	-1	-1
21	0	0	0	-1	-1	-1	-1	-1	-1	-1
22	0	0	0	-1	-1	-1	-1	-1	-1	-1
23	0	0	0	-1	-1	-1	-1	-1	-1	-1
24	0	0	0	0	0	-1	-1	-1	-1	-1
25	0	0	0	0	0	0	-1	-1	-1	-1
26	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	-1	-1
32	0	0	0	0	0	0	0	-1	-1	-1
33	0	0	0	0	0	-1	0	-1	-1	-1
34	0	0	0	0	0	0	0	0	-1	-1
35	0	0	0	0	0	0	-1	-1	0	-1
36	0	0	0	0	0	0	0	-1	0	-1
37	0	0	0	0	0	0	-1	0	0	0
38	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	-1	0	0
40	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0

	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	-1	-1	-1	-1	-1	-1	-1	-1	0	0
8	-1	-1	-1	-1	-1	-1	-1	-1	0	0
9	-1	-1	-1	-1	-1	-1	-1	-1	0	0
10	-1	-1	-1	-1	-1	-1	-1	-1	0	0
11	-1	-1	-1	-1	-1	-1	-1	-1	0	0
12	-1	-1	-1	-1	-1	-1	-1	-1	0	0
13	-1	-1	-1	-1	-1	-1	-1	-1	0	0
14	-1	-1	-1	-1	-1	-1	-1	-1	0	0
15	-1	-1	-1	-1	-1	-1	-1	-1	0	0
16	-1	-1	-1	-1	-1	-1	-1	-1	0	0
17	-1	-1	-1	-1	-1	-1	-1	-1	0	0
18	-1	-1	-1	-1	-1	-1	-1	-1	0	0
19	-1	-1	-1	-1	-1	-1	-1	-1	0	0
20	-1	-1	-1	-1	-1	-1	-1	-1	0	0
21	-1	-1	-1	-1	-1	-1	-1	-1	0	0
22	-1	-1	-1	-1	-1	-1	-1	-1	0	0
23	-1	-1	-1	-1	-1	-1	-1	-1	0	0
24	-1	-1	-1	-1	-1	-1	-1	-1	0	0
25	-1	-1	-1	-1	-1	-1	-1	-1	0	0
26	0	0	-1	-1	-1	-1	-1	-1	0	0
27	0	0	0	-1	-1	-1	-1	-1	0	0
28	0	0	0	-1	-1	-1	-1	-1	0	0
29	0	0	0	-1	-1	-1	-1	-1	0	0
30	0	0	-1	-1	-1	-1	-1	-1	0	0
31	-1	-1	-1	-1	-1	-1	-1	-1	0	0
32	-1	-1	-1	-1	-1	-1	-1	-1	0	0
33	-1	-1	-1	-1	-1	-1	-1	-1	0	0
34	-1	-1	-1	-1	-1	-1	-1	-1	0	0
35	-1	-1	-1	-1	-1	-1	-1	-1	0	0
36	-1	-1	-1	-1	-1	-1	-1	-1	0	0
37	-1	-1	-1	-1	-1	-1	-1	-1	0	0
38	-1	-1	0	-1	-1	-1	-1	-1	0	0
39	0	-1	-1	-1	-1	-1	-1	-1	0	0
40	0	-1	-1	-1	-1	-1	-1	-1	0	0
41	0	0	-1	0	-1	-1	-1	-1	0	0
42	0	0	-1	-1	-1	-1	-1	-1	0	0
43	0	0	0	-1	-1	-1	-1	-1	0	0
44	0	0	0	0	0	-1	-1	-1	0	0
45	0	0	0	-1	-1	-1	-1	-1	0	0
46	0	0	0	0	0	0	-1	-1	0	0
47	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	-11	-12	-12	-12	-12	-12	-12
8	0	0	0	-11	-11	-11	-11	-11	-11	-11
9	0	0	0	-11	-11	-11	-11	-11	-11	-11
10	0	0	-11	-11	-11	-11	-11	-11	-11	-11
11	0	0	0	-11	-11	-11	-11	-11	-11	-11
12	0	0	0	-11	-11	-11	-11	-11	-11	-11
13	0	0	0	-11	-11	-11	-11	-11	-11	-11
14	0	0	0	-10	-10	-10	-11	-11	-11	-11
15	0	0	0	-10	-10	-10	-10	-11	-11	-11
16	0	0	0	-10	-10	-10	-10	-10	-10	-11
17	0	0	0	-10	-10	-10	-10	-10	-10	-10
18	0	0	0	-10	-10	-10	-10	-10	-10	-10
19	0	0	0	-10	-10	-10	-10	-10	-10	-10
20	0	0	0	-10	-10	-10	-10	-10	-10	-10
21	0	0	0	-9	-9	-9	-10	-10	-10	-10
22	0	0	0	-9	-9	-9	-9	-10	-10	-10
23	0	0	0	-9	-9	-9	-9	-10	-10	-10
24	0	0	0	-9	-9	-9	-9	-9	-10	-10
25	0	0	0	-9	-9	-9	-9	-9	-10	-10
26	0	0	0	-8	-9	-9	-9	-9	0	0
27	0	0	0	-8	-8	-8	-8	-9	0	0
28	0	0	0	-8	-8	-8	-8	0	0	0
29	0	0	0	-8	-8	-8	-8	-8	0	0
30	0	0	0	-8	-8	-8	-8	-7	0	0
31	0	0	0	-8	-8	-8	-7	-7	-7	-6
32	0	0	0	-8	-8	-7	-7	-7	-7	-6
33	0	0	0	-7	-7	-7	-7	-7	-7	-7
34	0	0	0	-7	-7	-7	-7	-7	-7	-7
35	0	0	0	-7	-7	-7	-7	-7	-7	-7
36	0	0	0	-7	-7	-7	-7	-7	-7	-6
37	0	0	0	-7	-7	-7	-7	-6	-6	-6
38	0	0	0	-7	-7	-7	-6	-6	-6	-6
39	0	0	0	-6	-6	-6	-6	-6	-6	-6
40	0	0	0	-6	-6	-6	-6	-6	-6	-6
41	0	0	0	-6	-6	-6	-6	-6	-6	-6
42	0	0	0	-6	-6	-6	-6	-6	-6	-6
43	0	0	0	-6	-6	-6	-6	-6	-6	-6
44	0	0	0	-6	-6	-6	-6	-6	-6	-6
45	0	0	0	-6	-6	-6	-6	-6	-5	-6
46	0	0	0	-5	-5	-5	-5	-5	-6	-6
47	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0

	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	-12	-12	-12	-12	-12	-12	-12	-12	0	0
8	-12	-11	-11	-11	-11	-11	-11	-11	0	0
9	-11	-11	-11	-11	-11	-11	-11	-11	0	0
10	-11	-11	-11	-11	-11	-11	-11	-11	0	0
11	-11	-11	-11	-11	-11	-11	-11	-11	0	0
12	-11	-11	-11	-11	-11	-11	-11	-11	0	0
13	-11	-11	-11	-11	-11	-11	-11	-11	0	0
14	-11	-11	-11	-11	-11	-11	-11	-11	0	0
15	-11	-11	-11	-11	-11	-10	-10	-11	0	0
16	-10	-10	-10	-10	-10	-10	-10	-10	0	0
17	-10	-10	-10	-10	-10	-10	-10	-10	0	0
18	-10	-10	-10	-10	-10	-10	-10	-10	0	0
19	-10	-10	-10	-10	-10	-10	-10	-10	0	0
20	-10	-10	-10	-10	-10	-10	-10	-10	0	0
21	-10	-10	-10	-10	-10	-10	-10	-10	0	0
22	-10	-10	-10	-10	-9	-9	-9	-9	0	0
23	-10	-10	-10	-9	-9	-9	-9	-9	0	0
24	-10	-10	-10	-9	-9	-9	-9	-9	0	0
25	-10	-10	-9	-9	-9	-9	-9	-9	0	0
26	0	0	-9	-9	-9	-9	-9	-9	0	0
27	0	0	0	-9	-9	-9	-9	-9	0	0
28	0	0	0	-8	-8	-8	-8	-8	0	0
29	0	0	0	-8	-8	-8	-8	-8	0	0
30	0	0	-7	-8	-8	-8	-8	-8	0	0
31	-6	-7	-7	-8	-8	-8	-8	-8	0	0
32	-7	-7	-7	-7	-7	-8	-8	-8	0	0
33	-7	-7	-7	-7	-7	-7	-7	-8	0	0
34	-7	-7	-7	-7	-7	-7	-7	-7	0	0
35	-6	-7	-7	-7	-7	-7	-7	-7	0	0
36	-6	-6	-7	-7	-7	-7	-7	-7	0	0
37	-6	-6	-7	-7	-7	-7	-7	-7	0	0
38	-6	-6	-6	-7	-7	-7	-7	-7	0	0
39	-6	-6	-6	-6	-7	-7	-7	-7	0	0
40	-6	-6	-6	-6	-6	-6	-6	-7	0	0
41	-6	-6	-6	-6	-6	-6	-6	-6	0	0
42	-6	-6	-6	-6	-6	-6	-6	-6	0	0
43	-6	-6	-6	-6	-6	-6	-6	-6	0	0
44	-6	-6	-6	-6	-6	-6	-6	-6	0	0
45	-6	-6	-6	-6	-6	-6	-6	-6	0	0
46	-6	-5	-6	-6	-6	-6	-6	-6	0	0
47	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0

Results Interpretations:

Most of the x displacement values were 0 or -1 pixel. So where the value is 0; the x_displacement plot becomes plain. For all displacement and strain plots, where there is standard deviation, plots were drawn otherwise plain plots were drawn.

Actual Results form Ncorr website

For the same image, we can get the actual results from Ncorr website which is a much more delicate code to get the displacement and strain in more accuracy. That code also use consecutive image for beter measurements. That code output for the same image pair are given below.

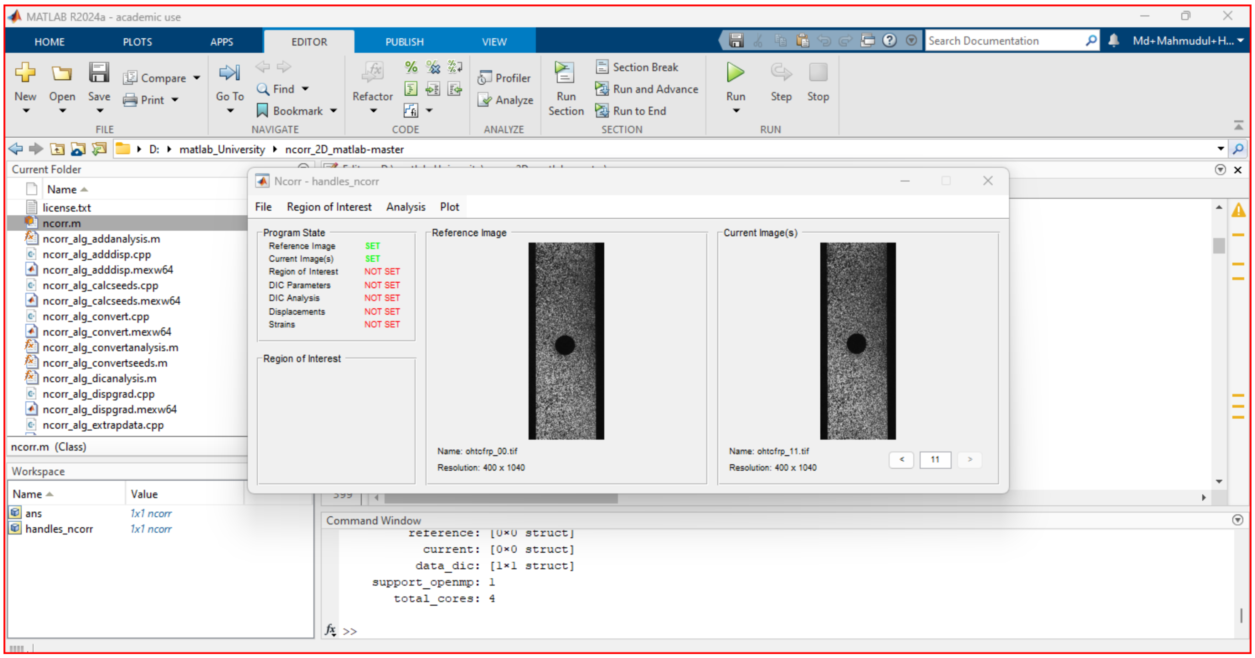


Figure: setting image at ncorr

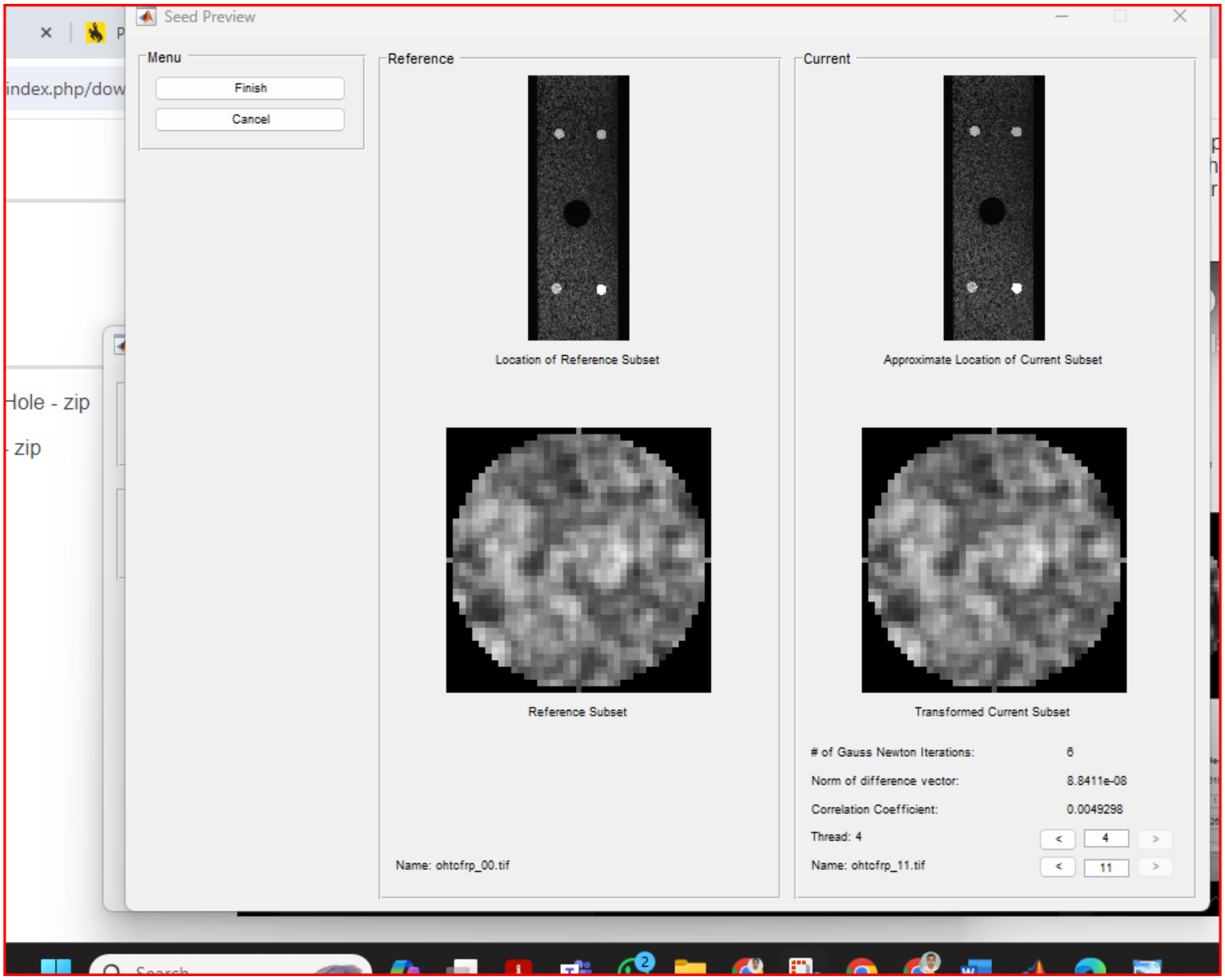


Figure: seed preview

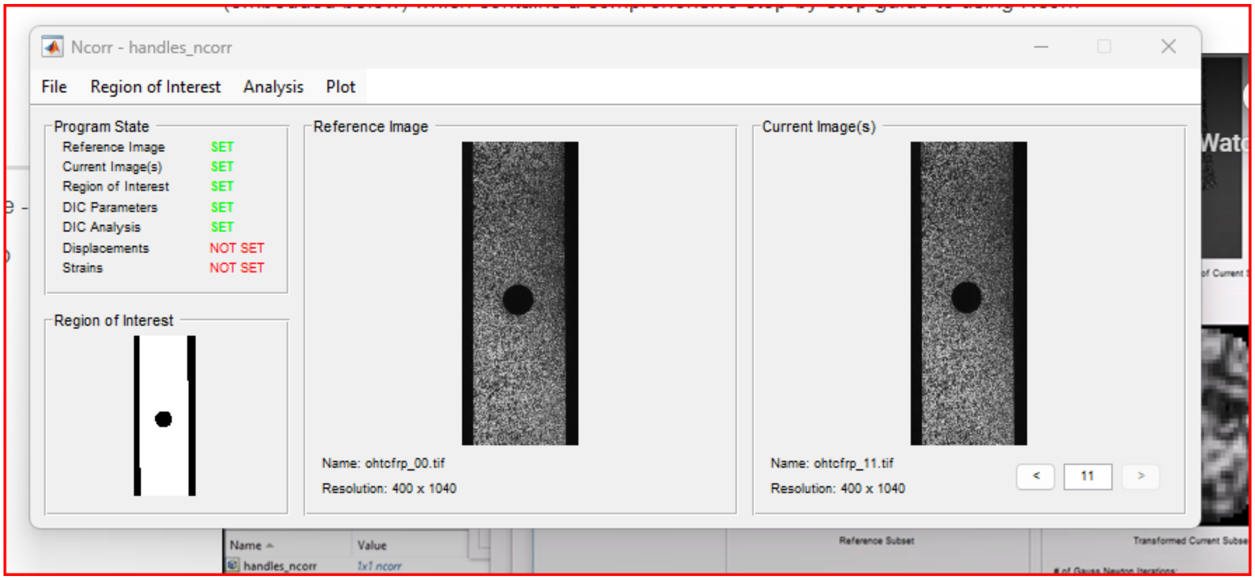


Figure: different steps in Ncorr

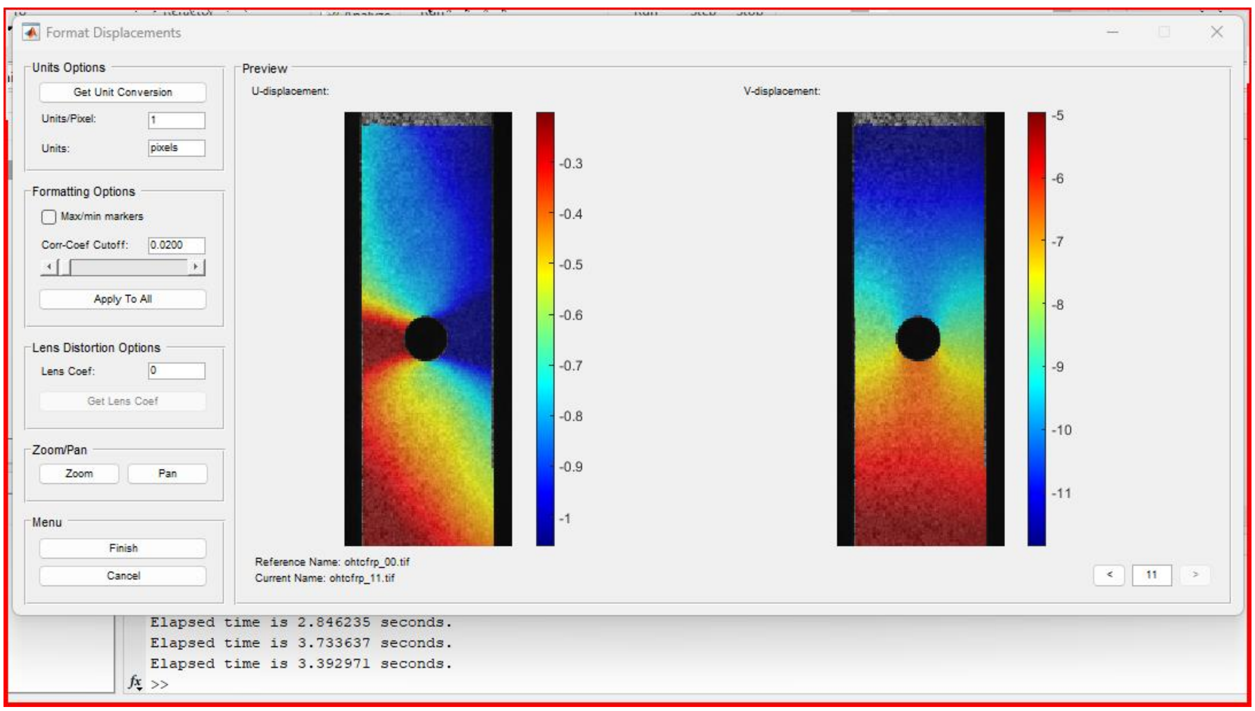


Figure: formet of displacement

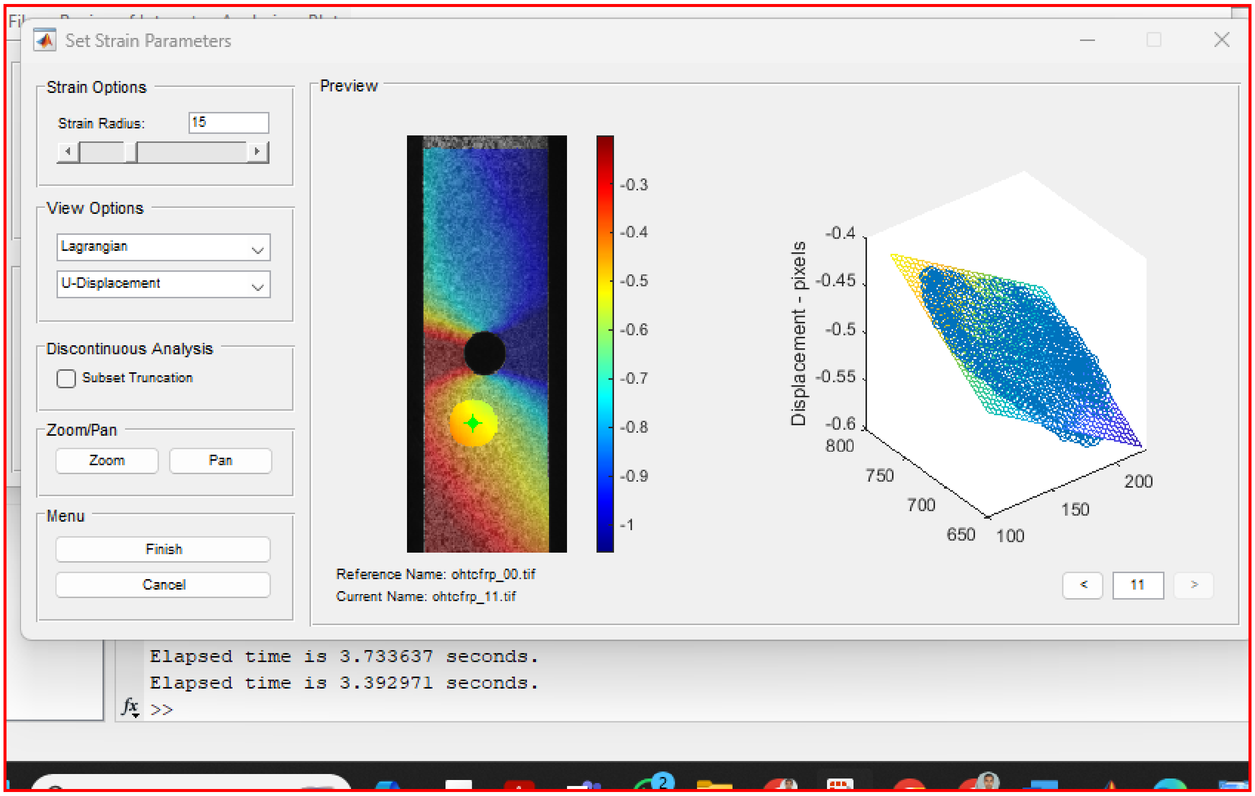


Figure: Formet of Lagrange Strain

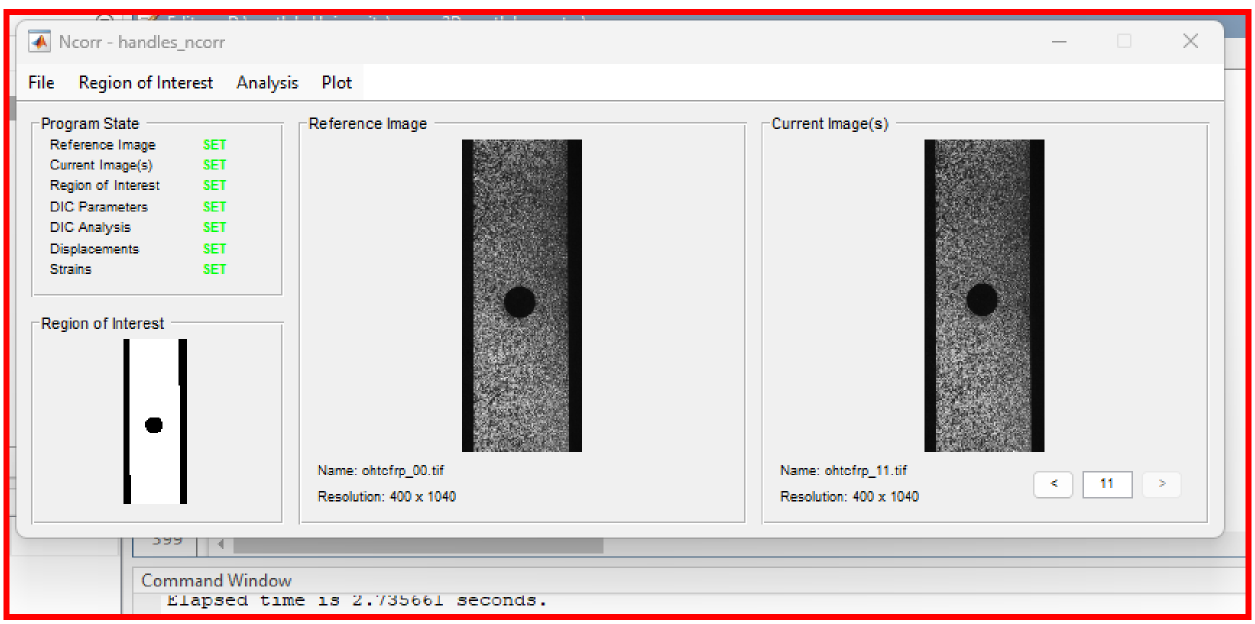


Figure: Completing all steps

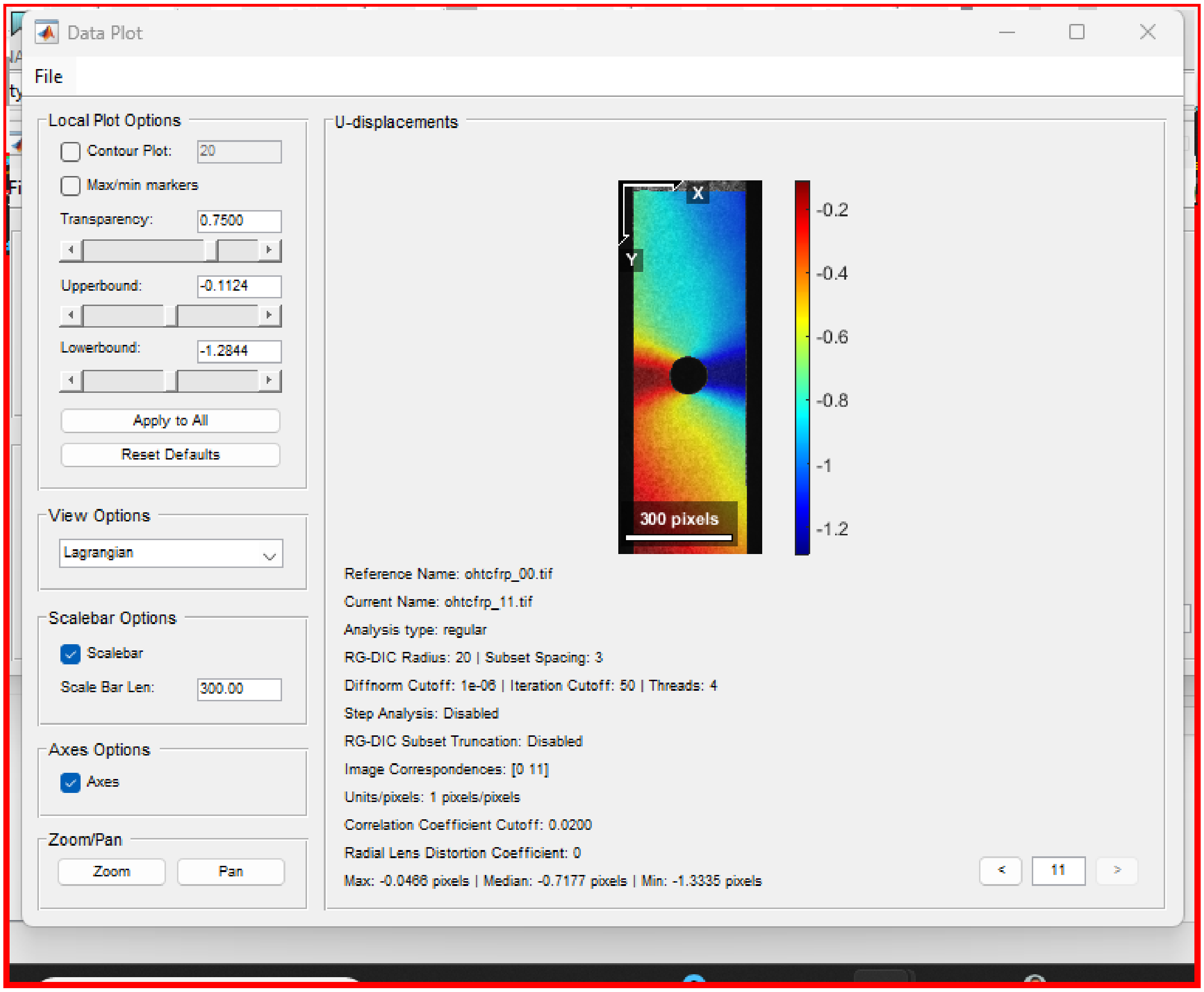


Figure: x_displacement plot configuration

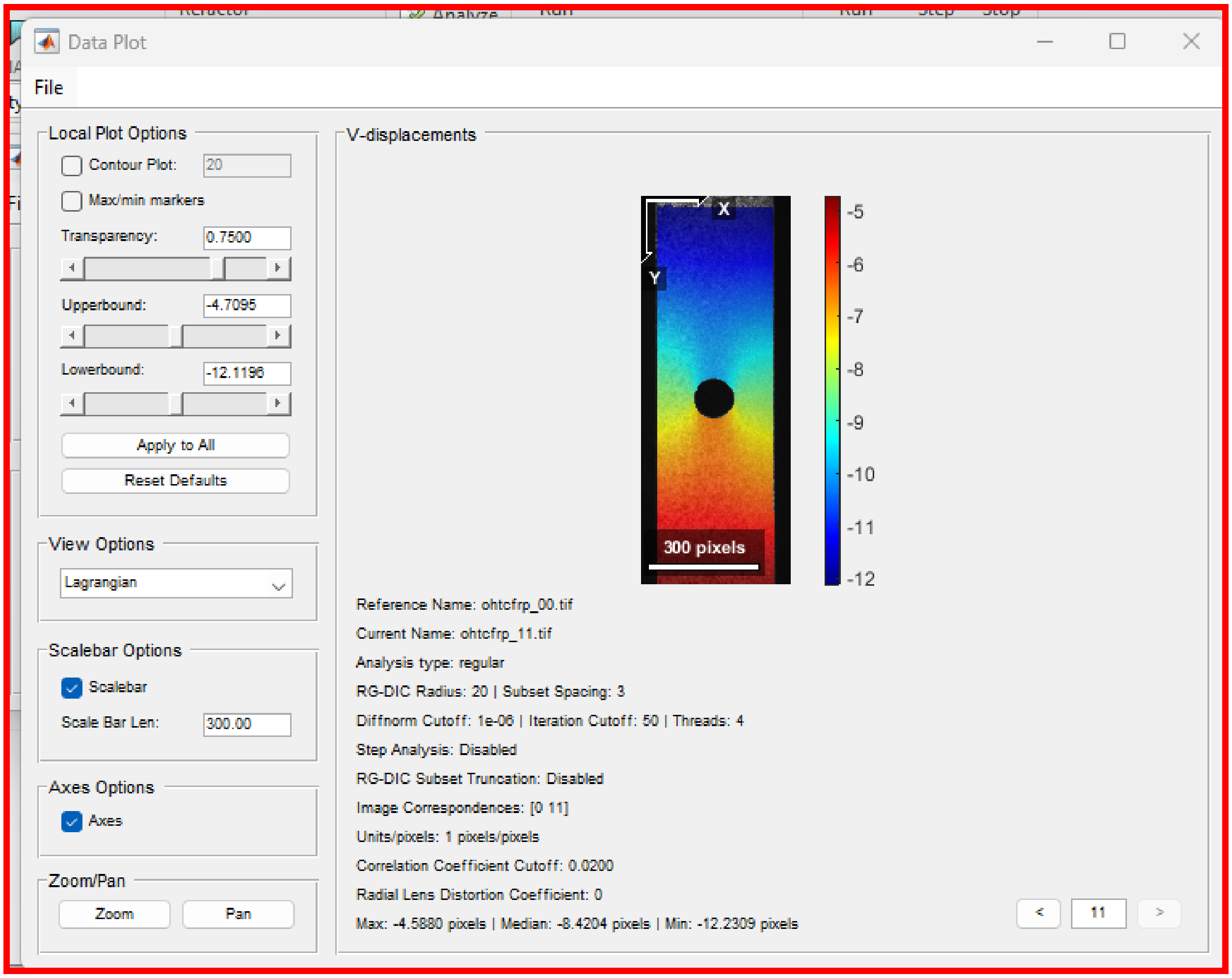


Figure: Y displacement plot configuration

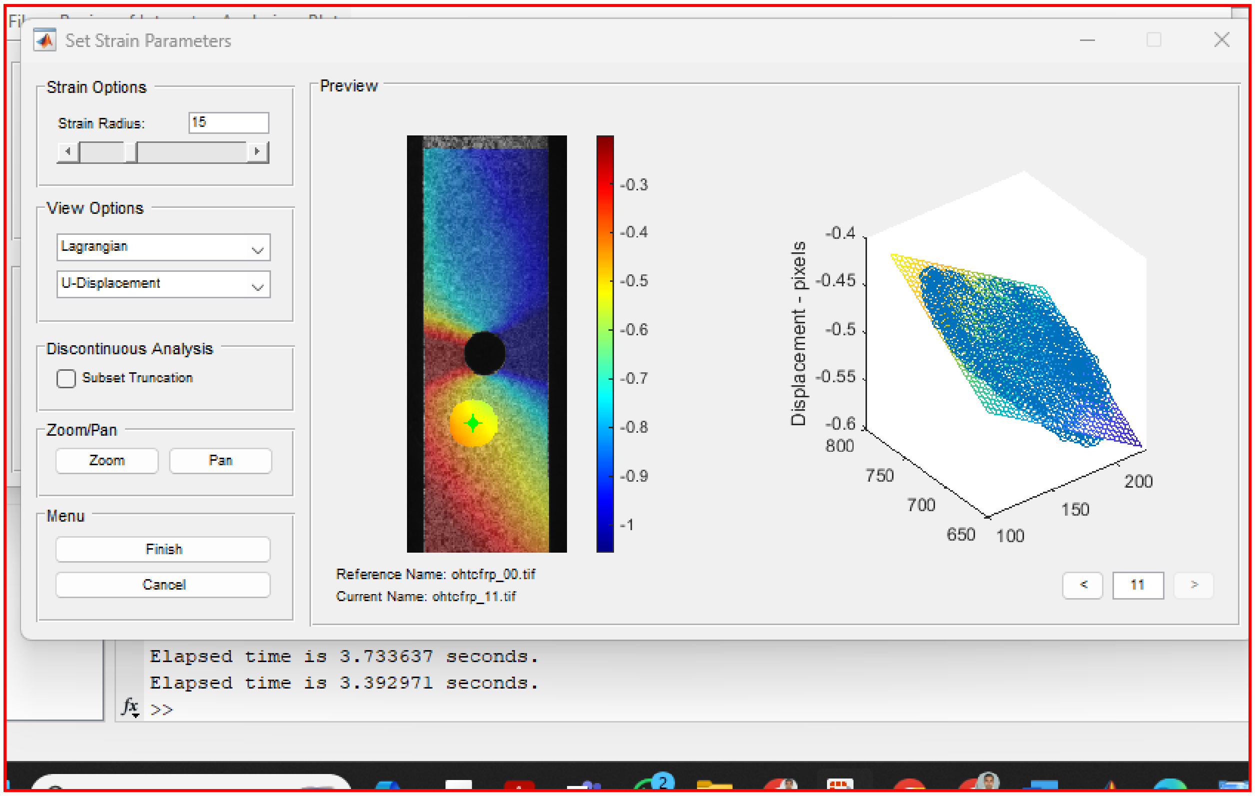


Figure: Lagrange strain configuration

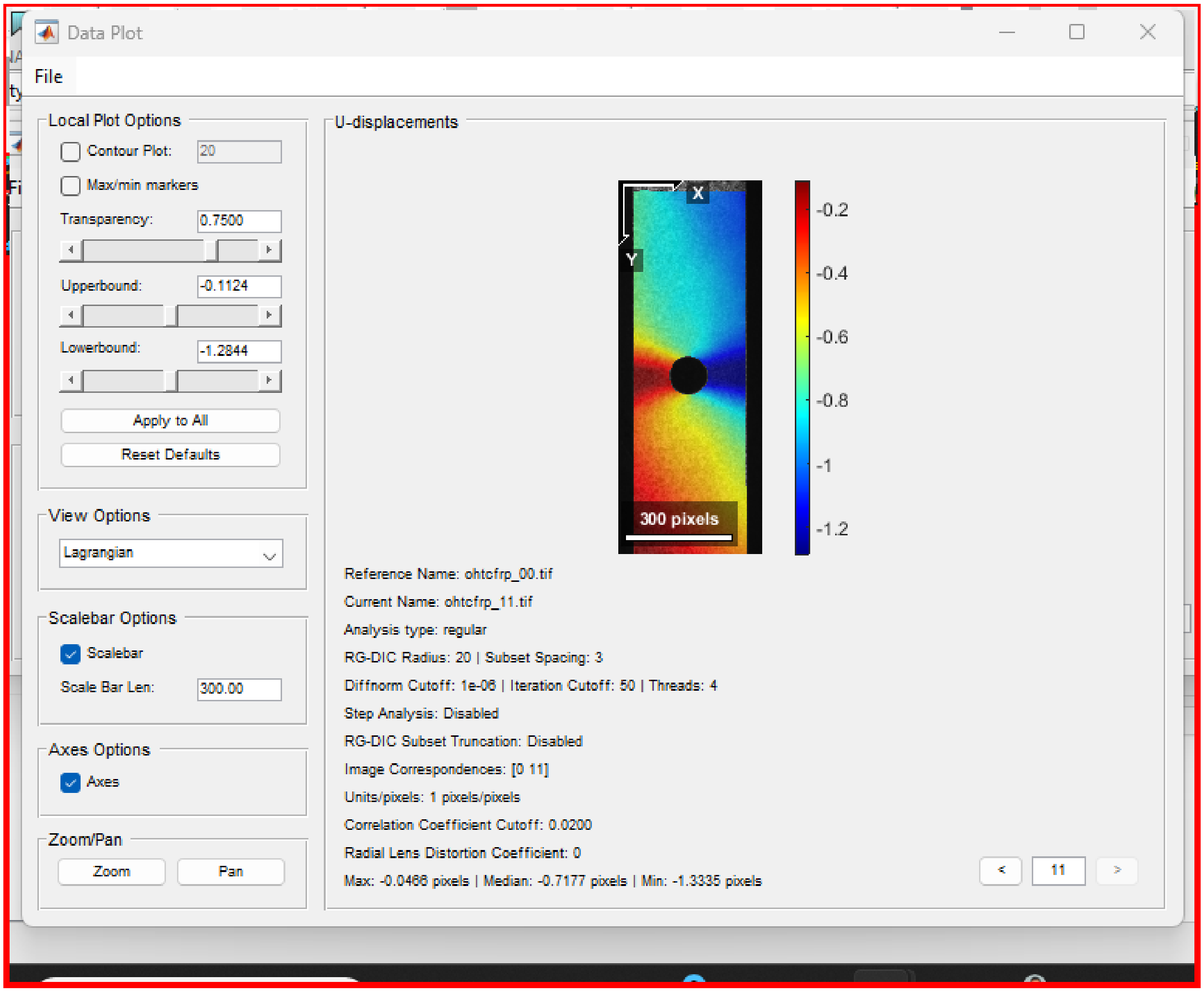


Figure: X displacement

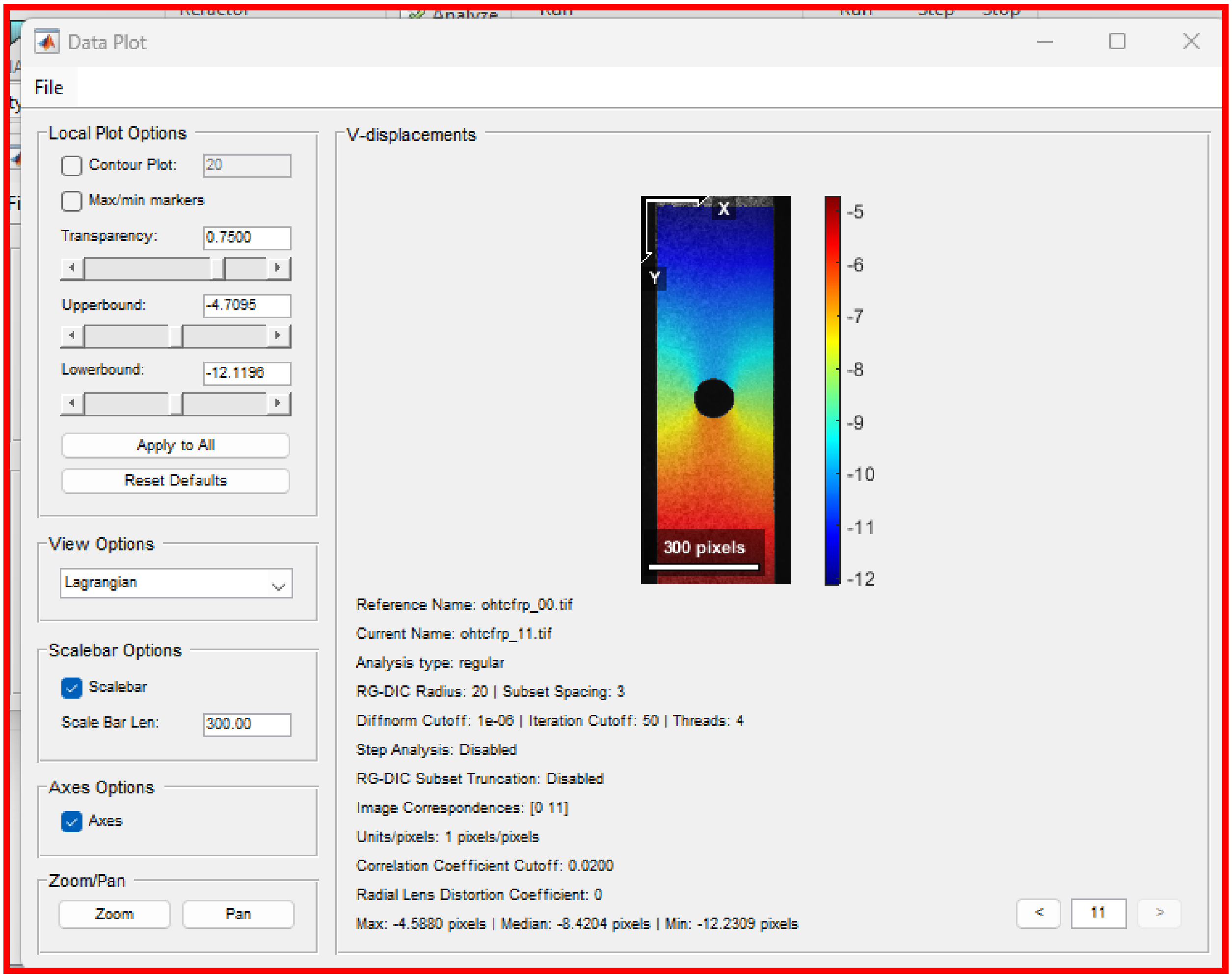


Figure: Y displacement plot

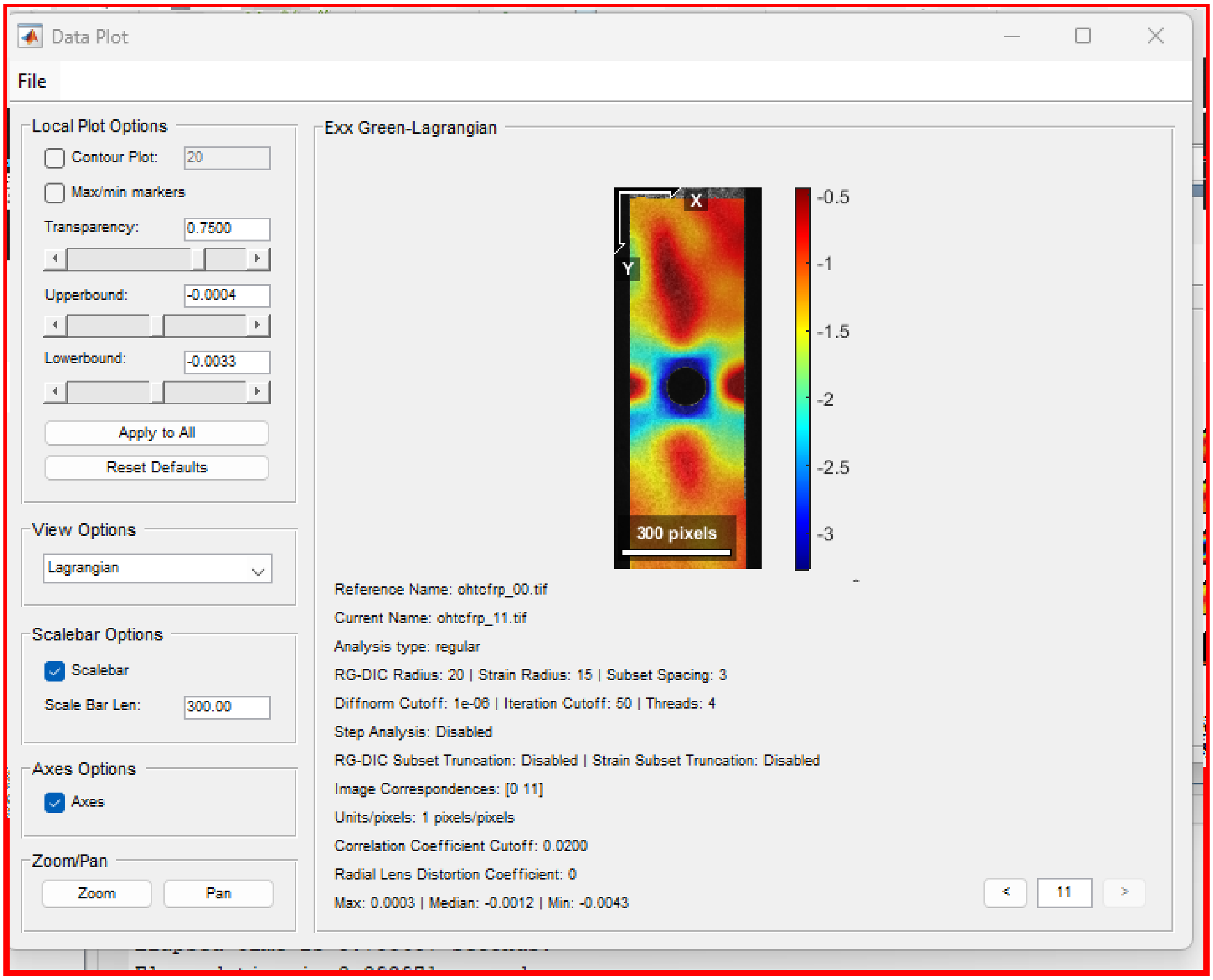


Figure: Exx Lagrange Strain

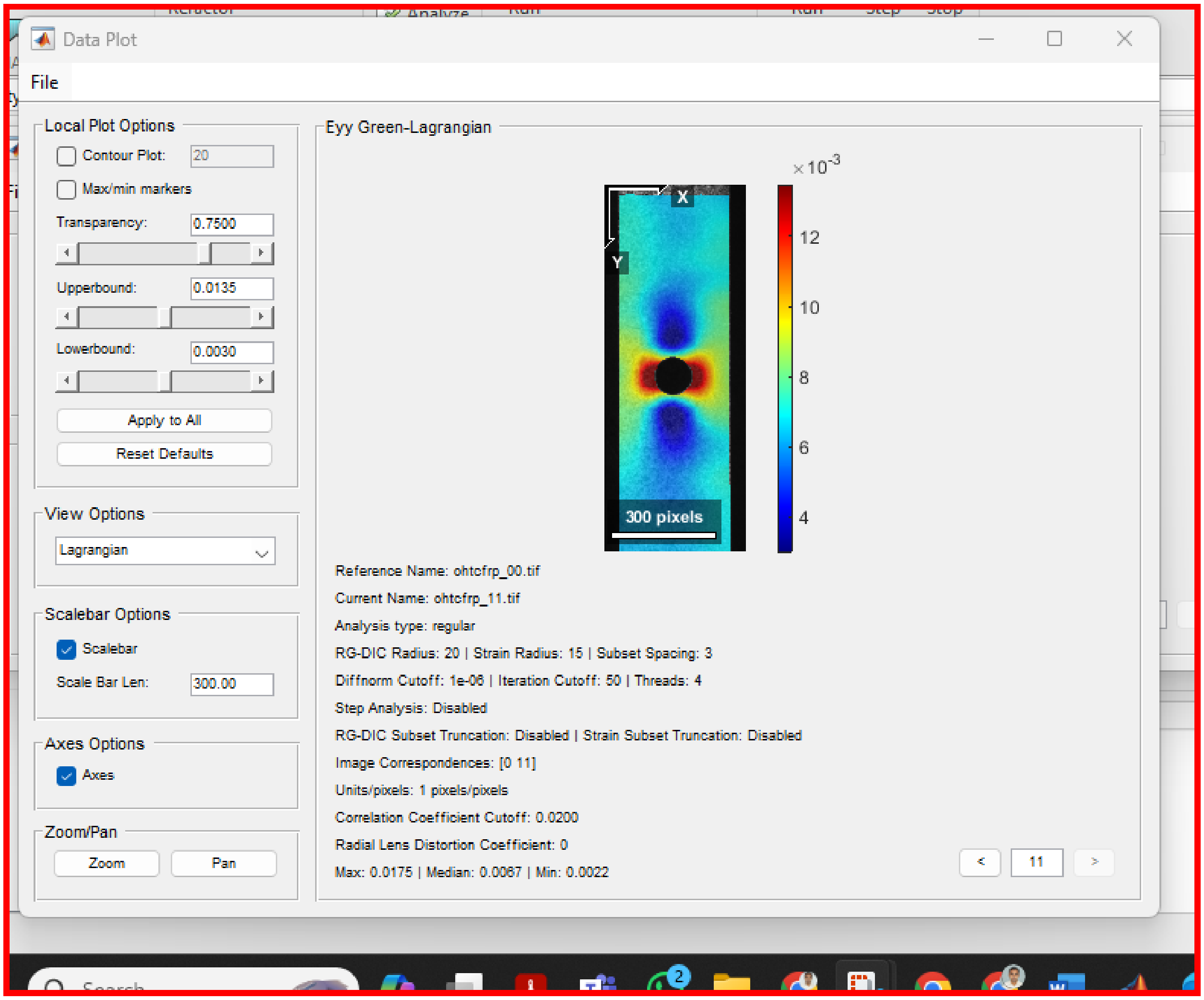


Figure: Eyy Lagrange Strain

This results uses different types of refinement and interpolation to compute the accurate results. So, there are some discrepancies between Ncorr results to this project's results.

3. Validation of Code

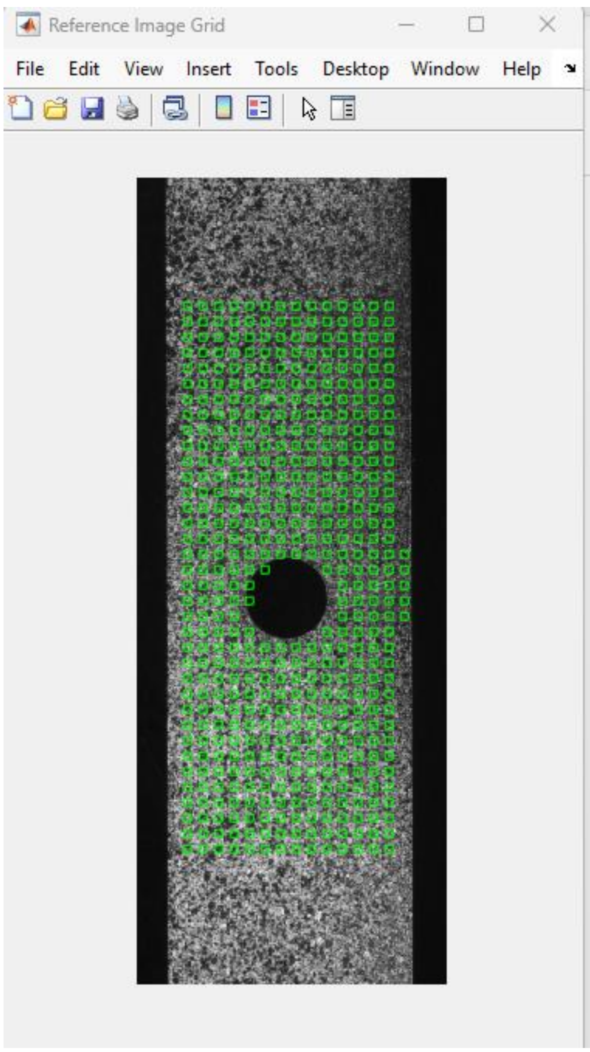
There are several ways to check the code accuracy. Some examples are given below:

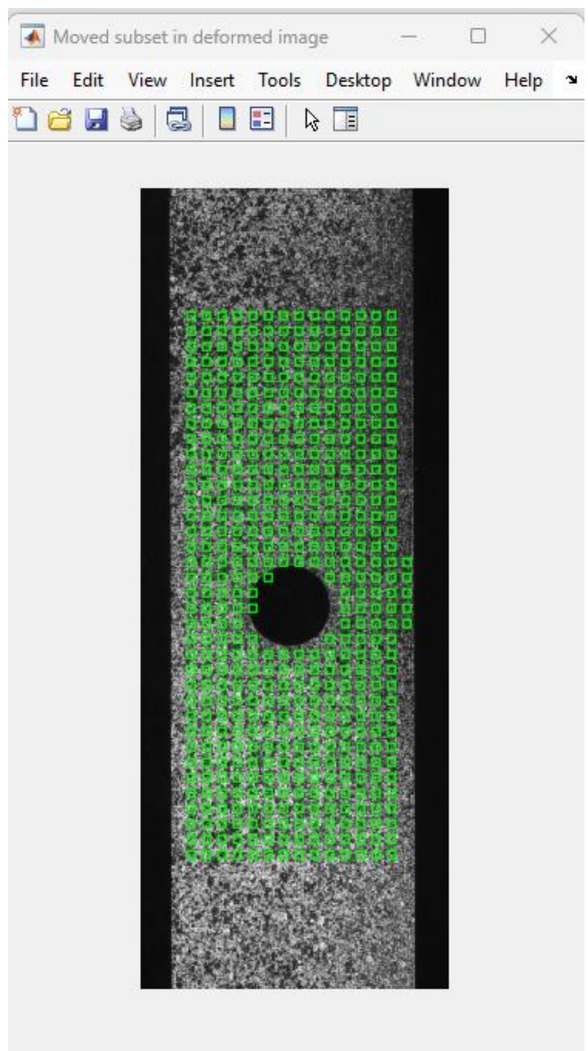
3.1.. Using reference image as the current image

Using the same image gives us the upper hand to check whether the code is right or wrong. It is clear that in this situation all types of displacement are zero. So, if we modify the first step of the code then it will be easily verified that the code is correct. All other steps remain the same as the main code provided in appendix.

```
R = imread('referenceimage.tif');
C = imread(' referenceimage.tif');
```

Now if the displacement at each point is zero (no subset movement), then it is inferred that the code is correct for the same images used in deformed and undeformed images. The output is given below-





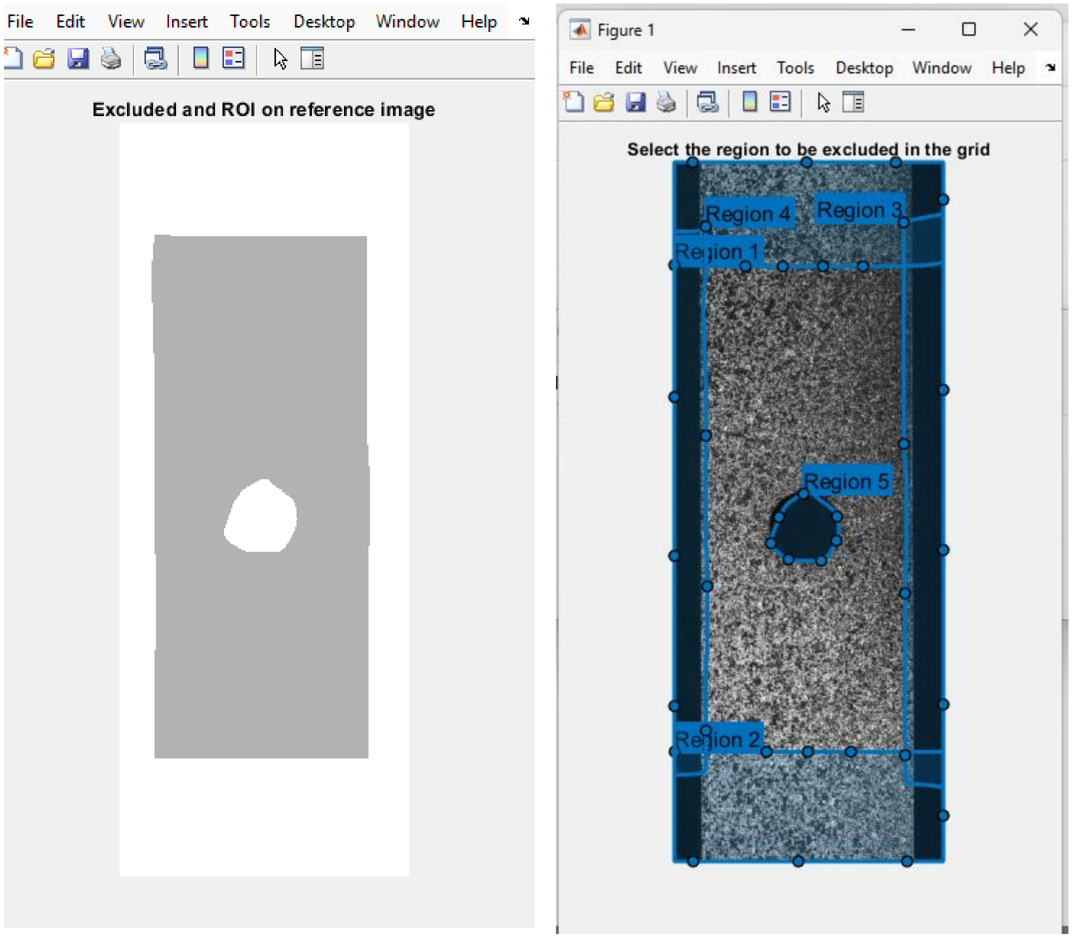


Figure: selection ROI

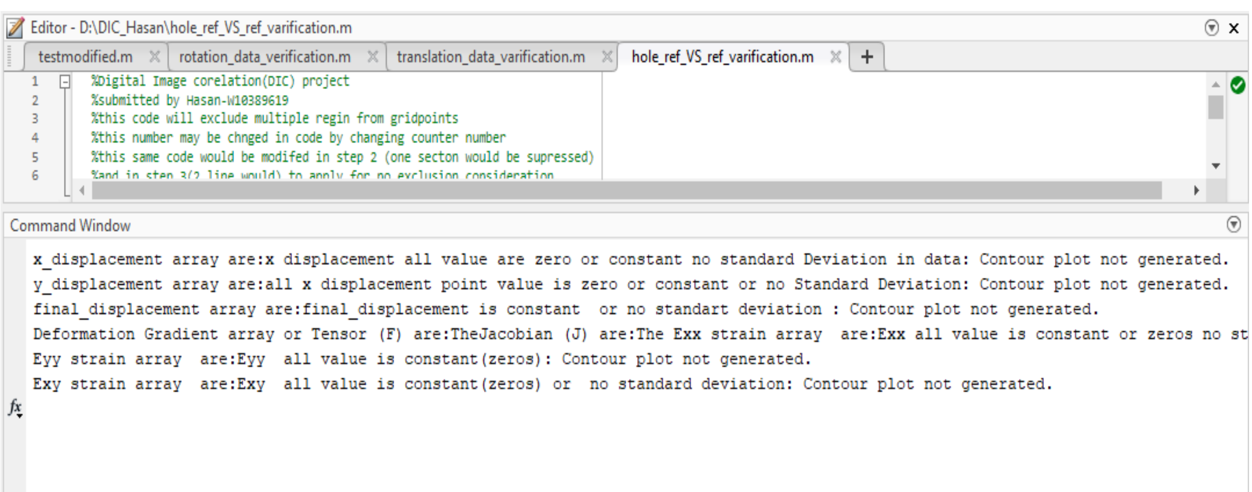


Figure: output terminal results instead of displacement and stain plot

X displacement:

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0

	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0

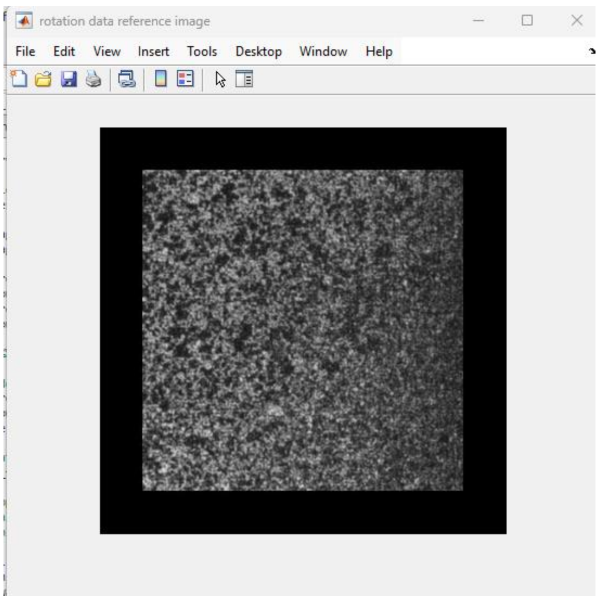
Y displacement:

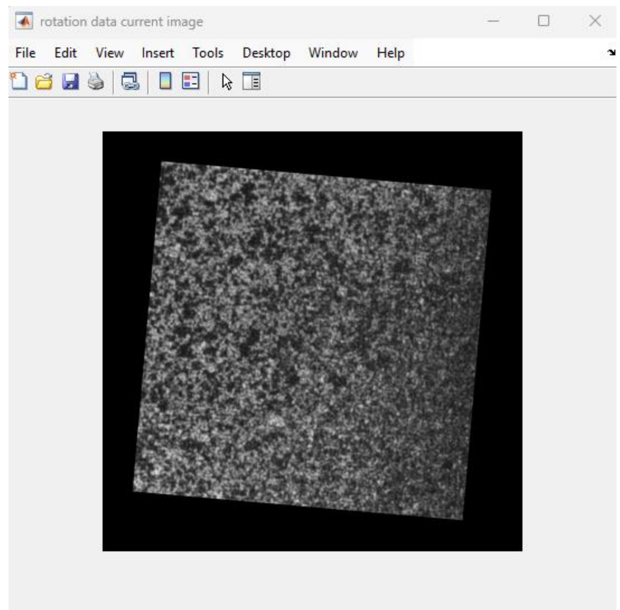
	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0

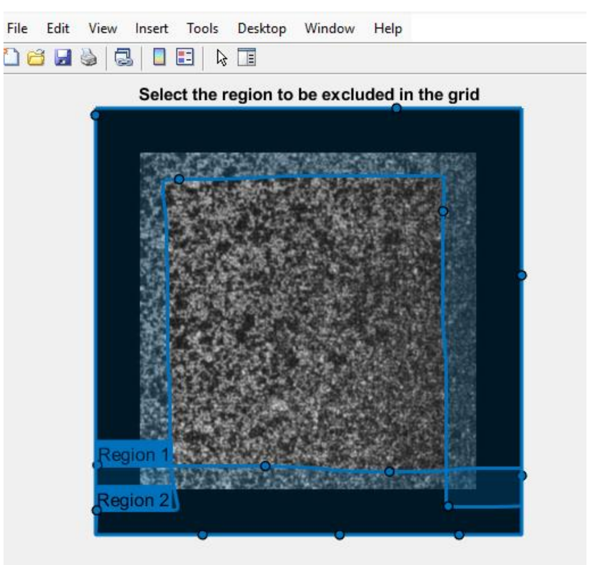
	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0

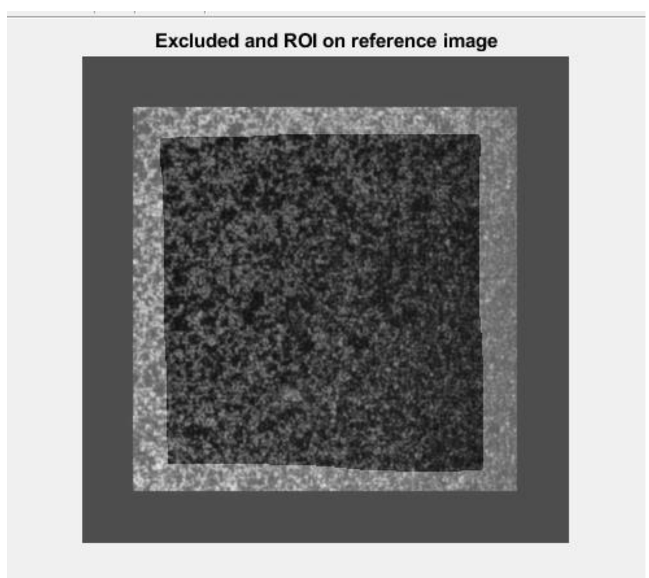
3.2 Known Value for displacement (Roataion_data fo Ncorr)

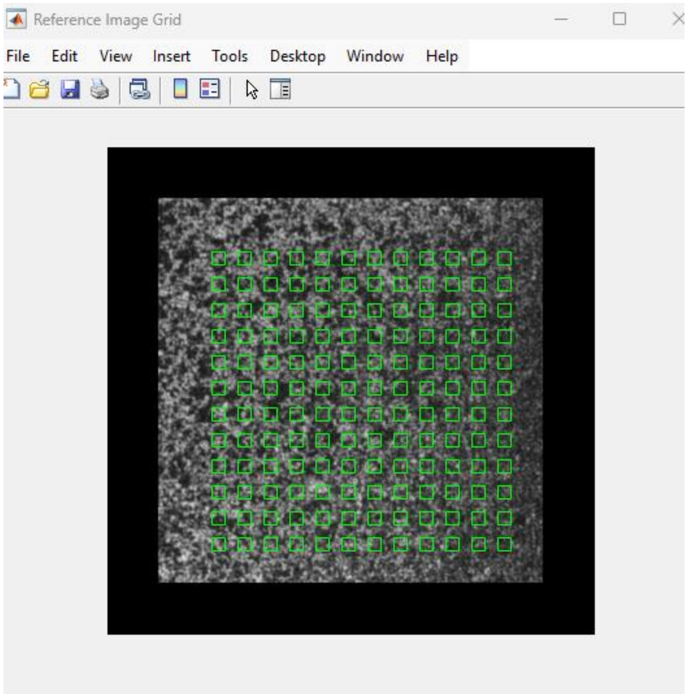
The images and their displacement values are given in Nocorr website. This project code results is different from that website for some reasons. Ncorr website uses some extra functions like cpnorm, and smoothing functions to refine the results. So, my results is not the same as those mentioned on that website. If this project code uses that further refining operation my results might be approach to accurate results. Output of these is given below:

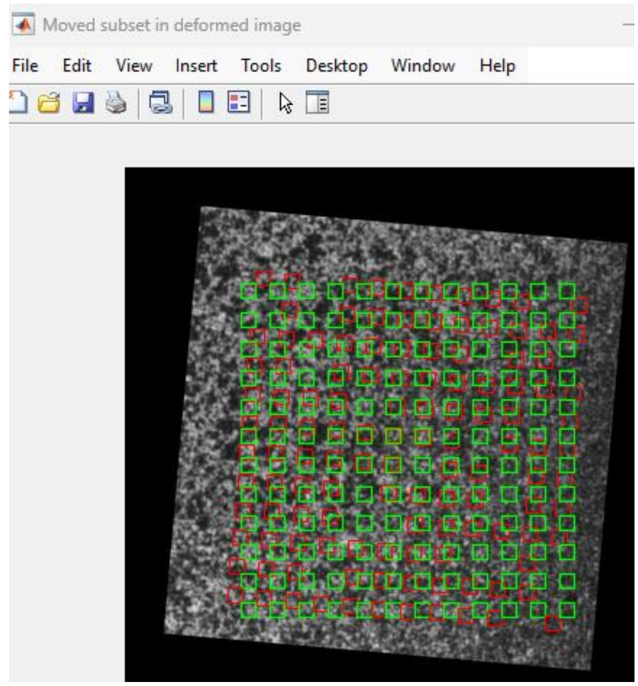


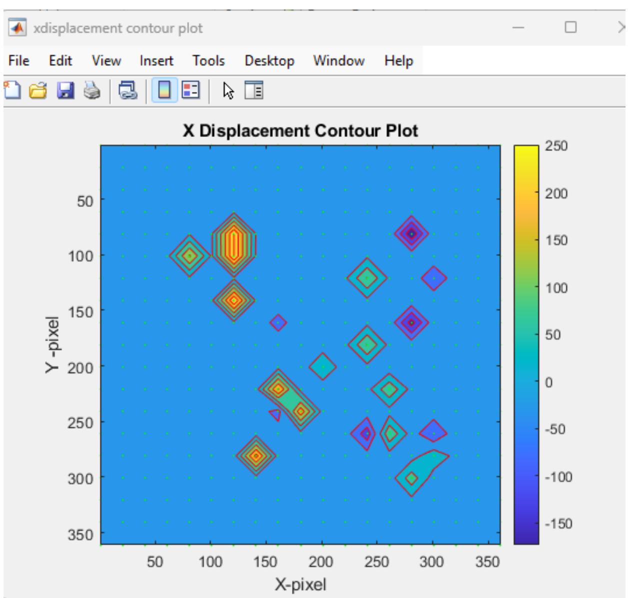


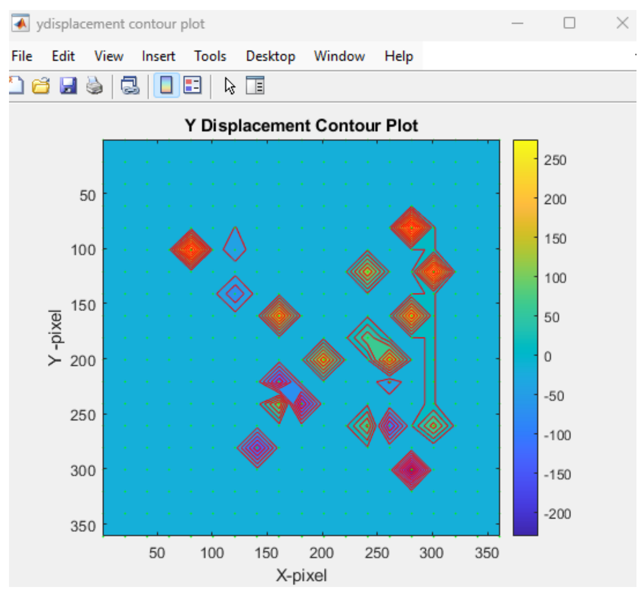


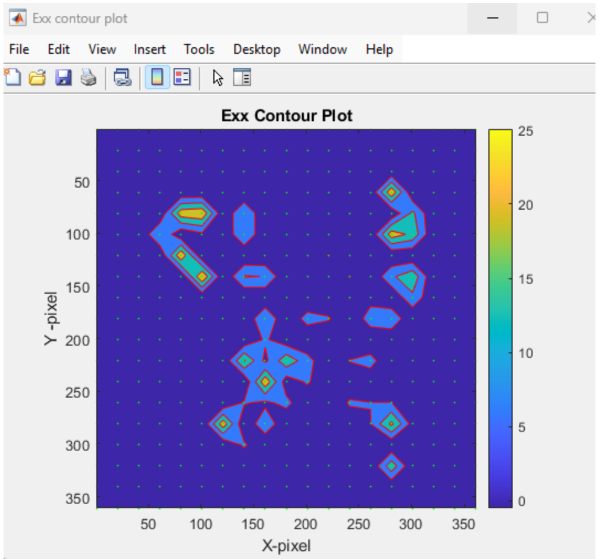



Reference Image Grid
File Edit View Insert Tools Desktop Window Help

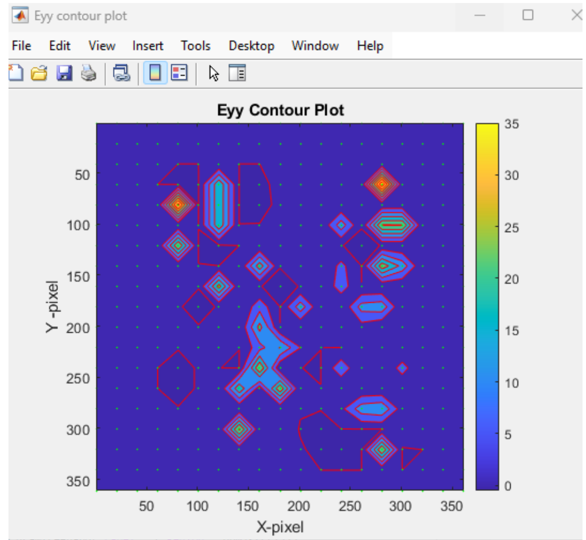

Moved subset in deformed image
File Edit View Insert Tools Desktop Window Help


xdisplacement contour plot
File Edit View Insert Tools Desktop Window Help
X Displacement Contour Plot
Y -pixel
X-pixel
50 100 150 200 250 300 350
250 200 150 100 50 0 -50 -100 -150

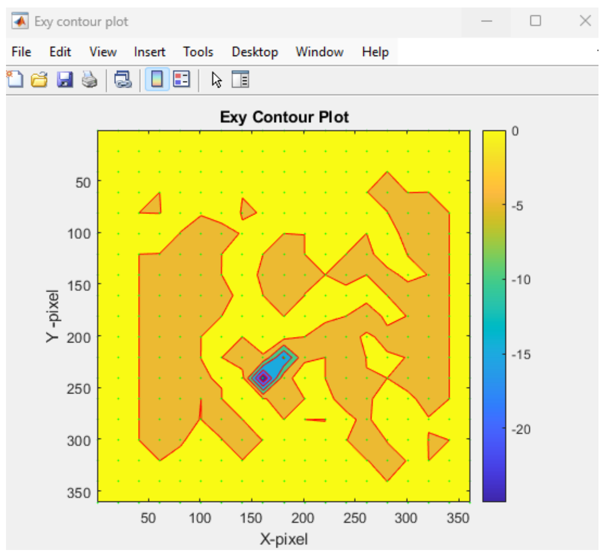

ydisplacement contour plot
File Edit View Insert Tools Desktop Window Help
Y Displacement Contour Plot
Y -pixel
X-pixel
50 100 150 200 250 300 350
250 200 150 100 50 0 -50 -100 -150 -200



Exx contour plot
File Edit View Insert Tools Desktop Window Help
Exx Contour Plot
Y -pixel
X-pixel



Eyy contour plot
File Edit View Insert Tools Desktop Window Help
Eyy Contour Plot
Y -pixel
X-pixel



Exy contour plot
File Edit View Insert Tools Desktop Window Help
Exy Contour Plot
Y -pixel
X-pixel

MATLAB Variable: Lagrange_Exx Page 1
Apr 7, 2024 2:40:52 AM

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0.0253	0.0153	0.1013	0.0050	0.0013	0
5	0	0	0	0.2813	22.5591	24.0153	0.2813	12.1303	-0.0234	0
6	0	0	0	12.5050	0.2200	4.4613	0.0450	12.2512	0	0
7	0	0	0	0.1613	23.9663	3.1250e-04	0.5000	0	0	0
8	0	0	0	0.1050	0.1050	25.0613	0	12.7613	12.1278	0
9	0	0	0	0.0778	0.0778	0	1.2553	-0.4200	0.0253	2.3206
10	0	0	0	0.0253	0.0253	3.1250e-04	0	0	12.1278	3.1250e-04
11	0	0	0	-0.0247	-0.0247	0	0	0	6.8450	1.6013
12	0	0	0	-0.0488	-0.0488	-0.0247	-0.0247	19.5028	4.2778	16.8528
13	0	0	0	-0.0950	-0.0947	0	0	-0.4744	23.3641	1.9200
14	0	0	0	-0.1387	-0.1384	0	0	6.5703	4.3513	8
15	0	0	0	-0.1800	-0.1800	0	22.6200	3.1250e-04	11.0200	0
16	0	0	0	-0.1997	-0.1684	0.0200	-0.0134	6.9131	0.0028	3.1250e-04
17	0	0	0	0	0.0313	0.0200	0.0113	0.0050	0.0028	3.1250e-04
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0

MATLAB Variable: Lagrange_Exx Page 2
Apr 7, 2024 2:40:52 AM

	11	12	13	14	15	16	17	18	19
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	3.1250e-04	0.0028	0.0078	0.0153	23.6328	0.0313	0	0	0
5	3.1250e-04	0.0028	0.0078	5.8166	0.0200	13.7091	-0.1997	0	0
6	-0.0247	-0.0247	6.1250	0	22.2778	16.9516	-0.1597	0	0
7	0	6.1613	-0.0247	0.9031	0.1050	-0.1172	3.9253	0	0
8	0	0	6.1250	3.1250e-04	11.0450	17.0162	-0.0950	0	0
9	-0.0247	0	2	3.4903	0	10.9003	-0.0488	0	0
10	8.4831	6.6253	0	9.1200	11.0450	0	0	0	0
11	0	-0.4988	1.7016	0.9113	0.5153	0.0253	0.0513	0	0
12	8.5078	0	6.0706	8	0.8203	0.0778	0.0778	0	0
13	6.9112	3.1250e-04	2.6203	2.0503	0	2.8941	0.1328	0	0
14	0	0.8203	6.5313	5.9800	6.0703	0.3856	4.0753	0	0
15	0	0	2.6450	5.6953	20.3263	3.0403	-0.4550	0	0
16	3.1250e-04	0.0028	0.0050	5.6363	0.0200	0.3516	0.2813	0	0
17	3.1250e-04	0.0013	0.0050	0.0113	16.5313	0.0253	0	0	0
18	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0

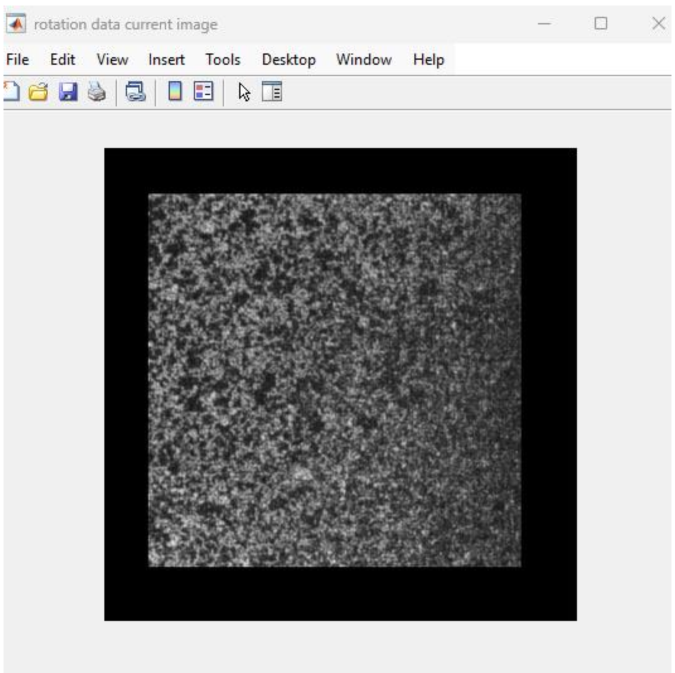
Here some of the displacement value as well as the Lagrange strain value is out of range due refinement issue of this project.

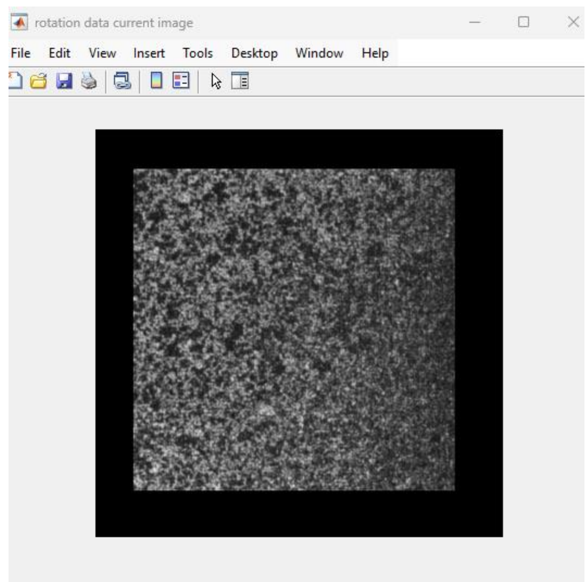
3.3 Known Value for displacement (translation_data fo Ncorr)

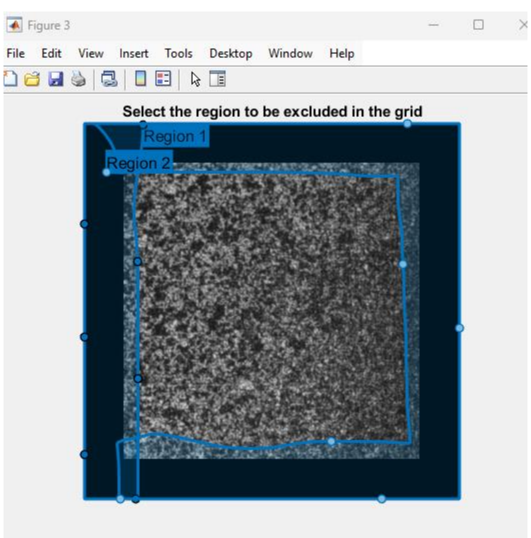
For running these data in this project, step 1 would be modified by the following code, and 'rotation_data.mat' file must be in the working directory of the Matlab

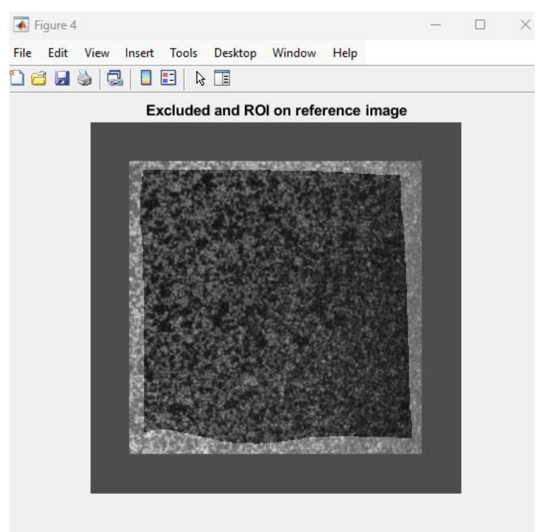
```
images= load('rotation_data.mat')
R=images.ref;
C=images.cur;
```

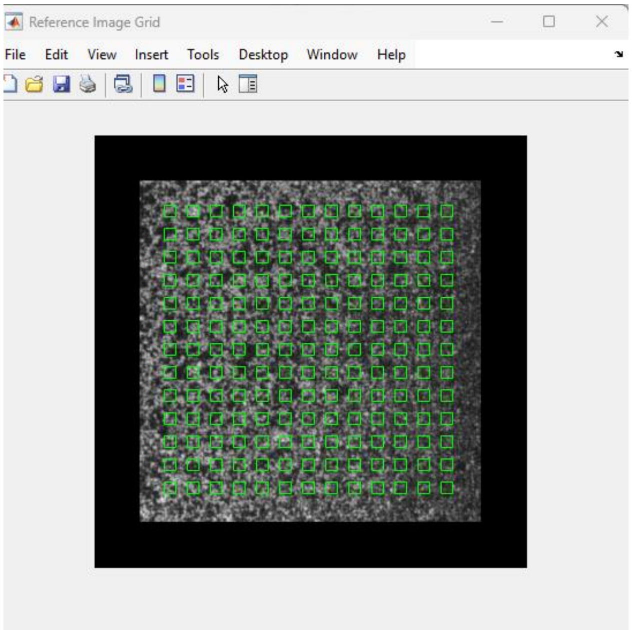
Here same issue happens as 3.1. Due to refinement and data smoothing shortages some of the values is out of range prescribed in Ncorr website. The project results for this file are given below:

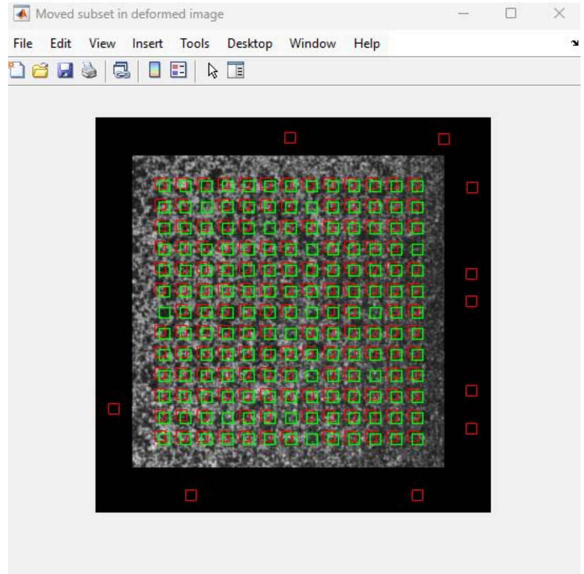


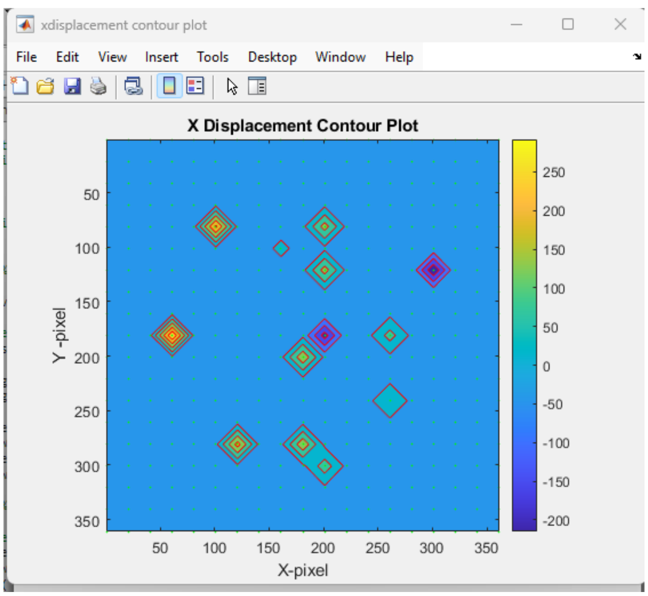


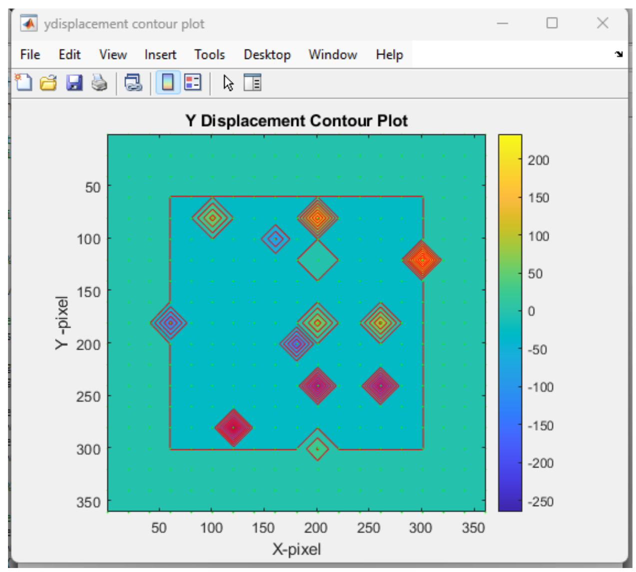




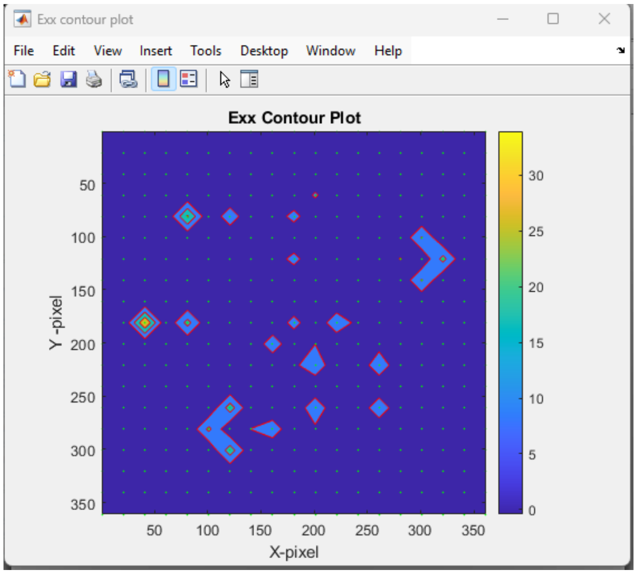


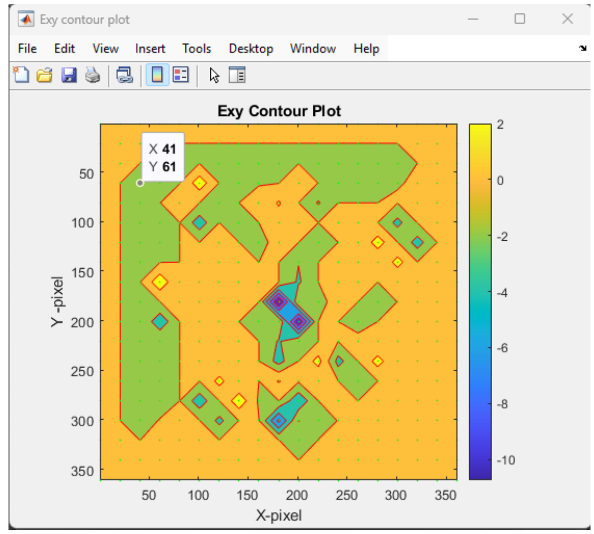


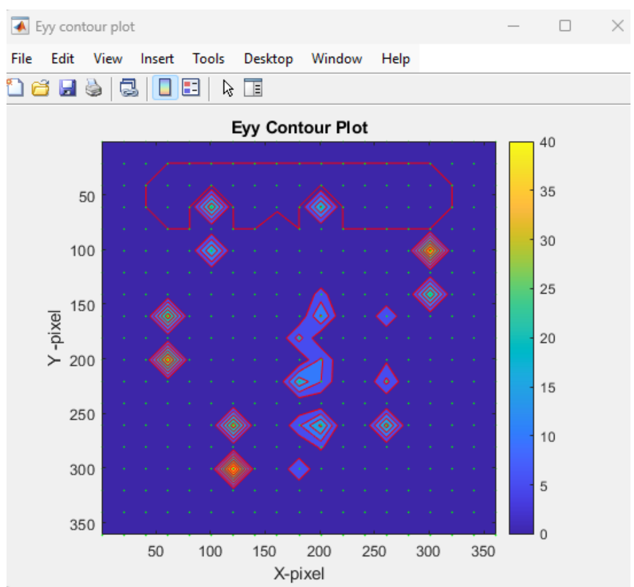




Here some subset during normxcorr2 moved to the black undeformed region. This is a point of improvement of this code.


Exx contour plot
File Edit View Insert Tools Desktop Window Help
Exx Contour Plot
Y -pixel
X-pixel
50 100 150 200 250 300 350
50 100 150 200 250 300 350
0 5 10 15 20 25 30


Exy contour plot
File Edit View Insert Tools Desktop Window Help
Exy Contour Plot
X 41
Y 61
Y -pixel
X-pixel
50 100 150 200 250 300 350
50 100 150 200 250 300 350
2 0 -2 -4 -6 -8 -10


Eyy contour plot
File Edit View Insert Tools Desktop Window Help
Eyy Contour Plot
Y -pixel
X-pixel
50 100 150 200 250 300 350
50 100 150 200 250 300 350
40 35 30 25 20 15 10 5 0

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	-4	-4	-4	-4	-4	-4	-4
5	0	0	0	-4	-4	251	-4	-4	-4	-4
6	0	0	0	-4	-4	-4	-4	-4	19	-4
7	0	0	0	-4	-4	-4	-4	-4	-4	-4
8	0	0	0	-4	-4	-4	-4	-4	-4	-4
9	0	0	0	-4	-4	-4	-4	-4	-4	-4
10	0	0	0	292	-4	-4	-4	-4	-4	-4
11	0	0	0	-4	-4	-4	-4	-4	-4	172
12	0	0	0	-4	-4	-4	-4	-4	-4	-4
13	0	0	0	-4	-4	-4	-4	-4	-4	-4
14	0	0	0	-4	-4	-4	-4	-4	-4	-4
15	0	0	0	-4	-4	-4	205	-4	-4	171
16	0	0	0	-4	-4	-4	-4	-4	-4	-4
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0

	11	12	13	14	15	16	17	18	19
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	-4	-4	-4	-4	-4	-4	0	0	0
5	151	-4	-4	-4	-4	-4	0	0	0
6	-4	-4	-4	-4	-4	-4	0	0	0
7	151	-4	-4	-4	-4	-215	0	0	0
8	-4	-4	-4	-4	-4	-4	0	0	0
9	-4	-4	-4	-4	-4	-4	0	0	0
10	-188	-4	-4	91	-4	-4	0	0	0
11	-4	-4	-4	-4	-4	-4	0	0	0
12	-4	-4	-4	-4	-4	-4	0	0	0
13	-21	-4	-4	65	-4	-4	0	0	0
14	-4	-4	-4	-4	-4	-4	0	0	0
15	-4	-4	-4	-4	-4	-4	0	0	0
16	100	-4	-4	-4	-4	-4	0	0	0
17	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0

Many values go out of range abruptly and wrongfully find their(like row10, column 4) matched position on the blacked region. This is a point of improvement. Same issue happen in y displacement and subsequent Lagrange strain. Improvement in this code will ensure the results just like Ncorr website results.

3.3 Known Value for displacement (Crack file of Ncorr website)

Results from Ncorr website code which almost produced as accurate results are given below:

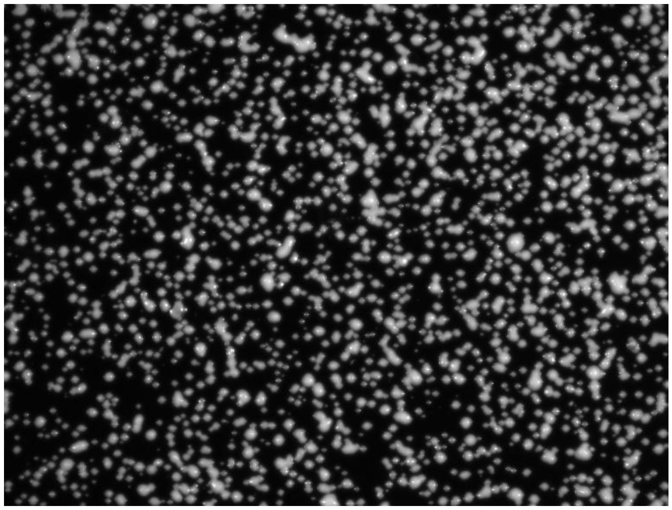


Figure: Reference

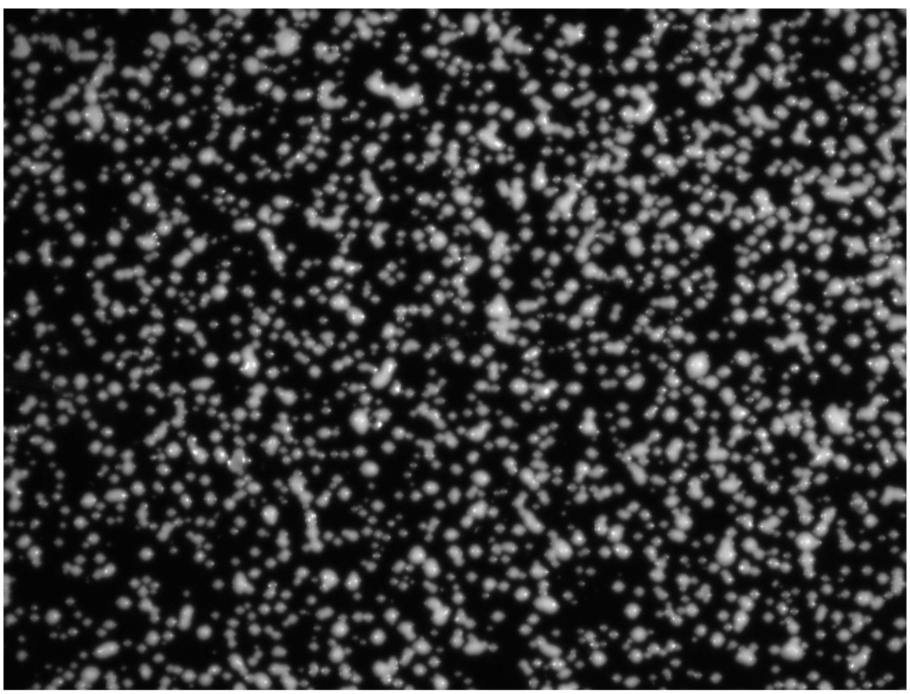


Figure: current image

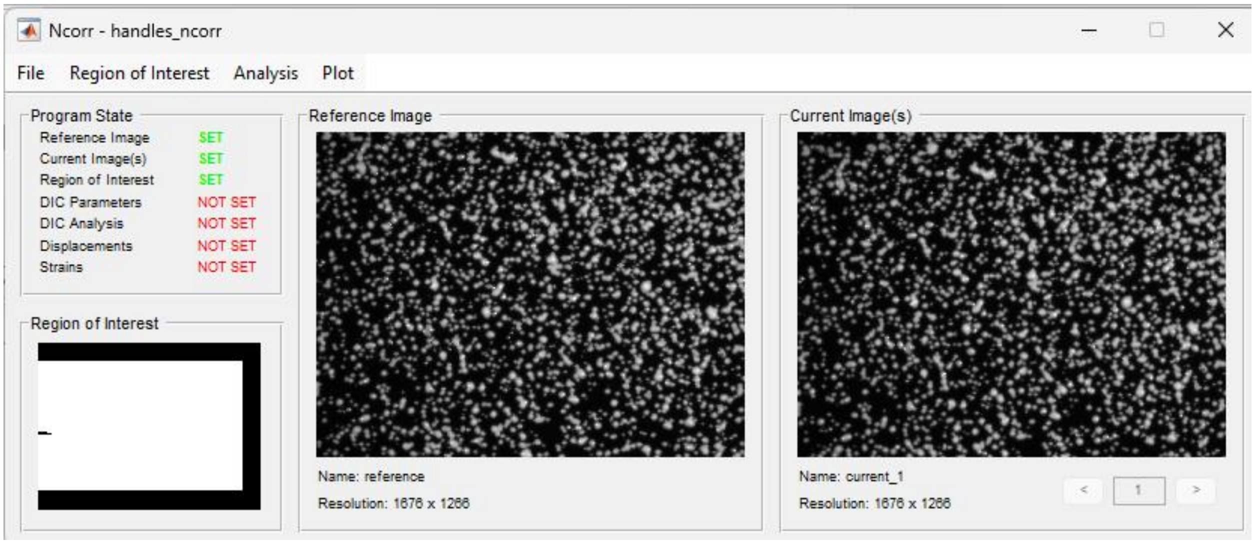
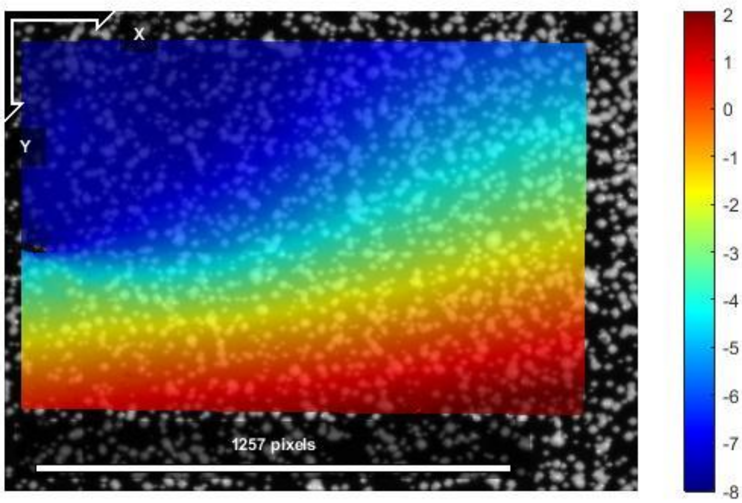
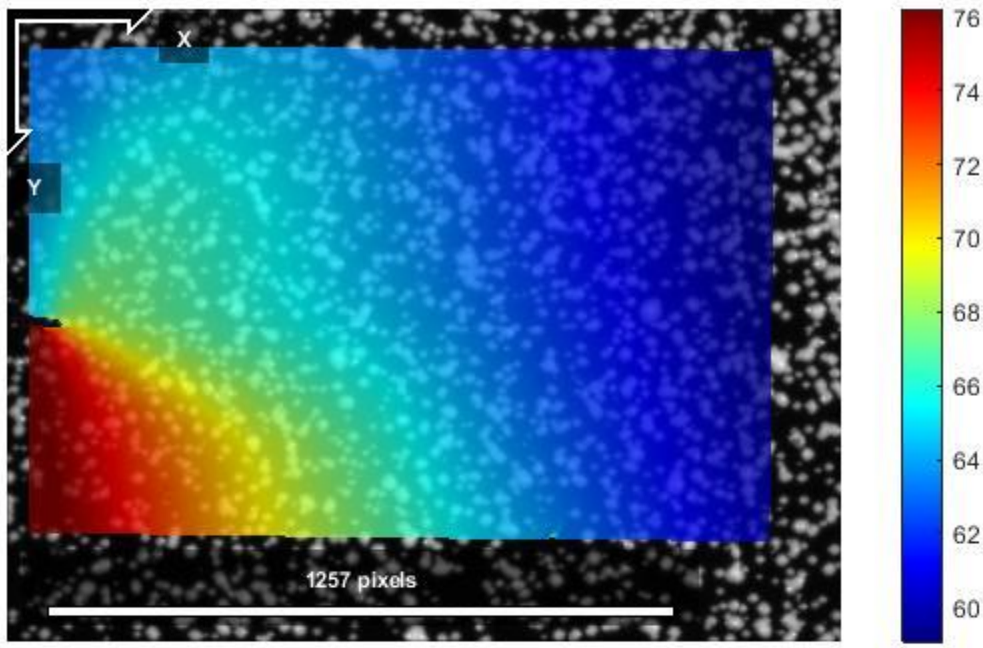


Figure: preparing all data file



Type: u-plot
Reference Name: reference
Current Name: current_1
Analysis type: backward
RG-DIC Radius: 42 | Subset Spacing: 3
Diffnorm Cutoff: 1e-06 | Iteration Cutoff: 50 | Threads: 4
Step Analysis: Disabled
RG-DIC Subset Truncation: Disabled
Image Correspondences: [1 0]
Units/pixels: 1 pixels/pixels
Correlation Coefficient Cutoff: 0.0536
Radial Lens Distortion Coefficient: 0
Max: 2.6513 pixels | Median: -4.1259 pixels | Min: -8.5806 pixels

Figure: x displacement



Type: v-plot

Reference Name: reference

Current Name: current_1

Analysis type: backward

RG-DIC Radius: 42 | Subset Spacing: 3

Diffnorm Cutoff: 1e-06 | Iteration Cutoff: 50 | Threads: 4

Step Analysis: Disabled

RG-DIC Subset Truncation: Disabled

Image Correspondences: [1 0]

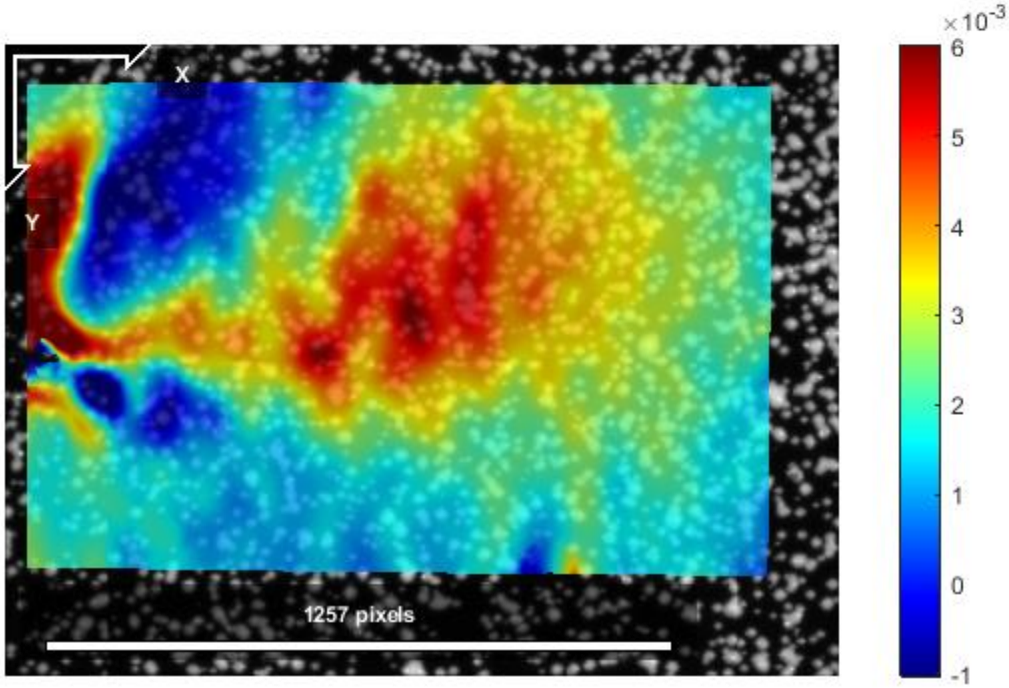
Units/pixels: 1 pixels/pixels

Correlation Coefficient Cutoff: 0.0536

Radial Lens Distortion Coefficient: 0

Max: 77.3967 pixels | Median: 64.0631 pixels | Min: 58.7354 pixels

Figure: Y displacement



Type: exx-plot

Reference Name: reference

Current Name: current_1

Analysis type: backward

RG-DIC Radius: 42 | Strain Radius: 15 | Subset Spacing: 3

Diffnorm Cutoff: 1e-06 | Iteration Cutoff: 50 | Threads: 4

Step Analysis: Disabled

RG-DIC Subset Truncation: Disabled | Strain Subset Truncation: Disabled

Image Correspondences: [1 0]

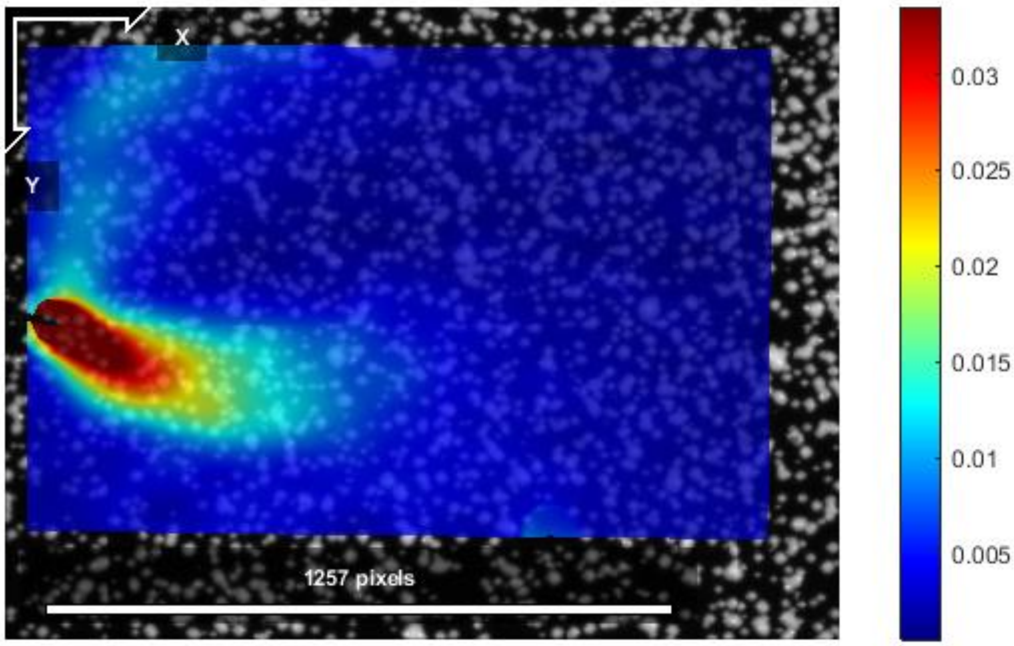
Units/pixels: 1 pixels/pixels

Correlation Coefficient Cutoff: 0.0536

Radial Lens Distortion Coefficient: 0

Max: 0.0117 | Median: 0.0023 | Min: -0.0069

Figure: Exx lagrange strain



Type: eyy-plot

Reference Name: reference

Current Name: current_1

Analysis type: backward

RG-DIC Radius: 42 | Strain Radius: 15 | Subset Spacing: 3

Diffnorm Cutoff: 1e-06 | Iteration Cutoff: 50 | Threads: 4

Step Analysis: Disabled

RG-DIC Subset Truncation: Disabled | Strain Subset Truncation: Disabled

Image Correspondences: [1 0]

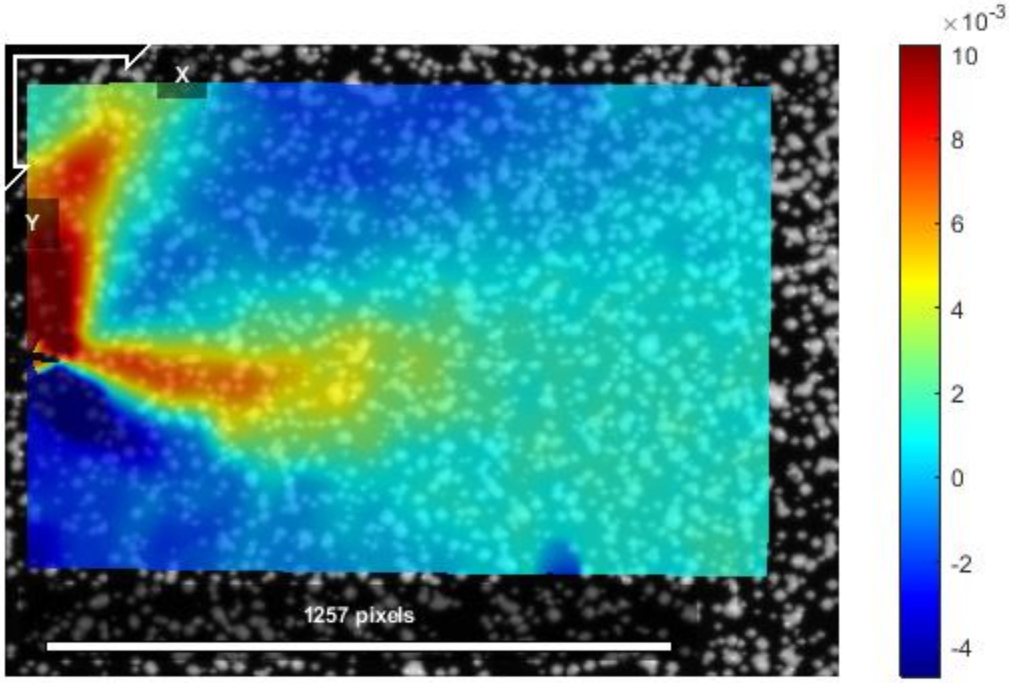
Units/pixels: 1 pixels/pixels

Correlation Coefficient Cutoff: 0.0536

Radial Lens Distortion Coefficient: 0

Max: 0.1338 | Median: 0.0031 | Min: -0.0000

Figure: Eyy Lagrange strain



Type: exy-plot

Reference Name: reference

Current Name: current_1

Analysis type: backward

RG-DIC Radius: 42 | Strain Radius: 15 | Subset Spacing: 3

Diffnorm Cutoff: 1e-06 | Iteration Cutoff: 50 | Threads: 4

Step Analysis: Disabled

RG-DIC Subset Truncation: Disabled | Strain Subset Truncation: Disabled

Image Correspondences: [1 0]

Units/pixels: 1 pixels/pixels

Correlation Coefficient Cutoff: 0.0536

Radial Lens Distortion Coefficient: 0

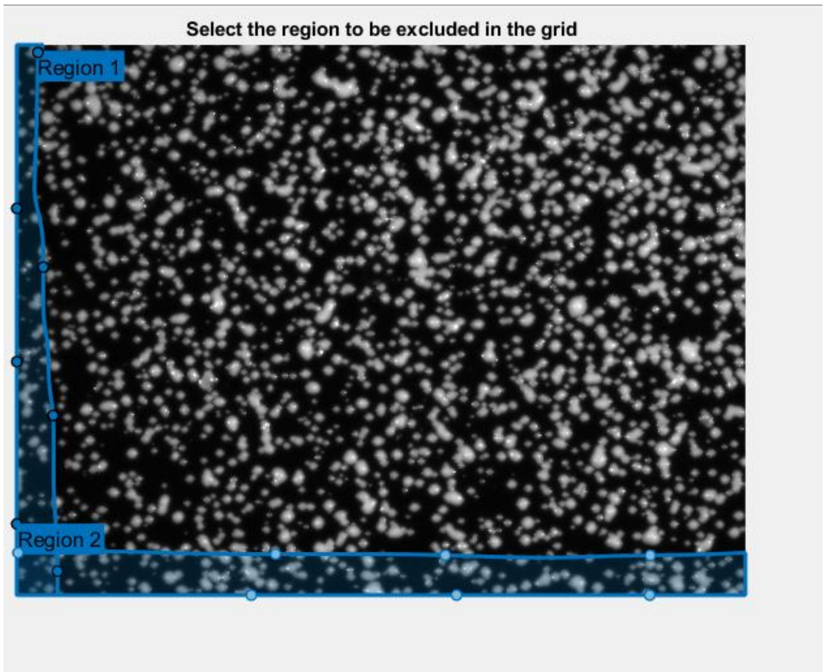
Max: 0.0177 | Median: 0.0009 | Min: -0.0184

Figure: Exy Lagrange strain

To ran the same file in this project code step 1 need to be modified just like this:

```
R = imread('ref.TIF');
C = imread('cur_920.TIF');
```

This is a big size image (almost 2500*35000) pixel. So based on the computational capability the excluded region number, subset size and grid spacing was modified. This project output for the crack sample images are given below:



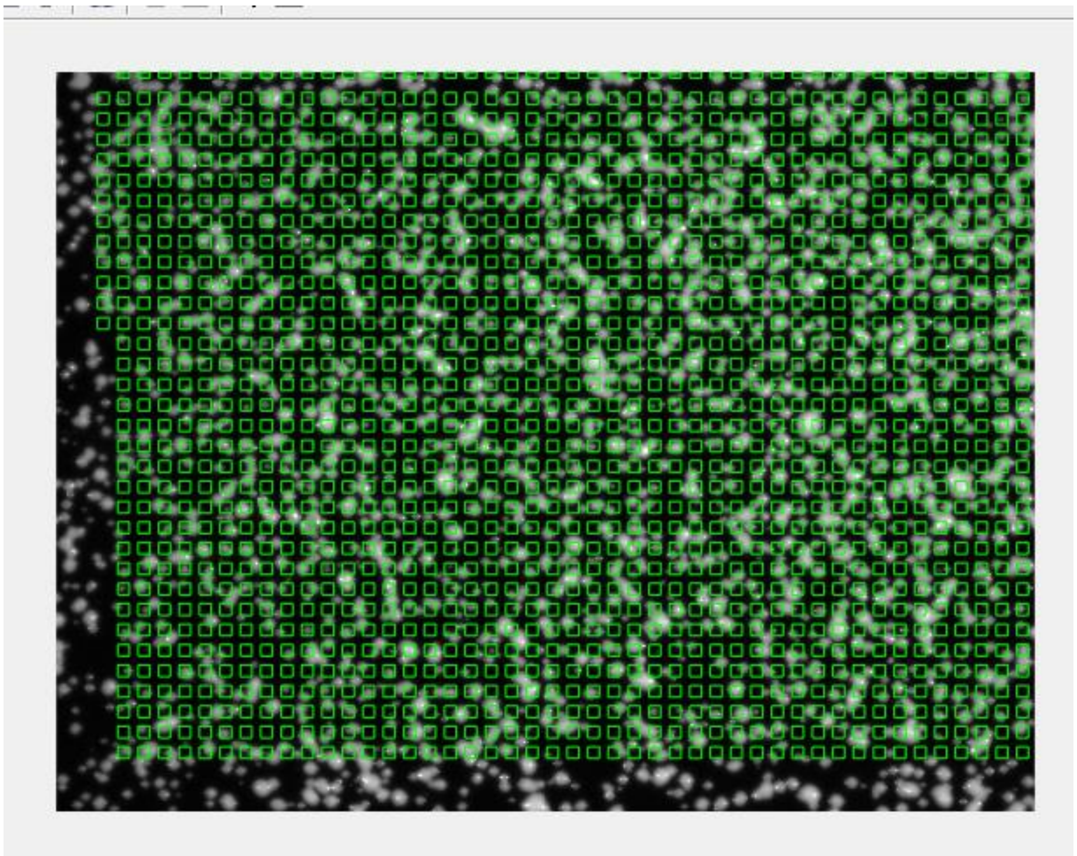


Figure: Reference image subset

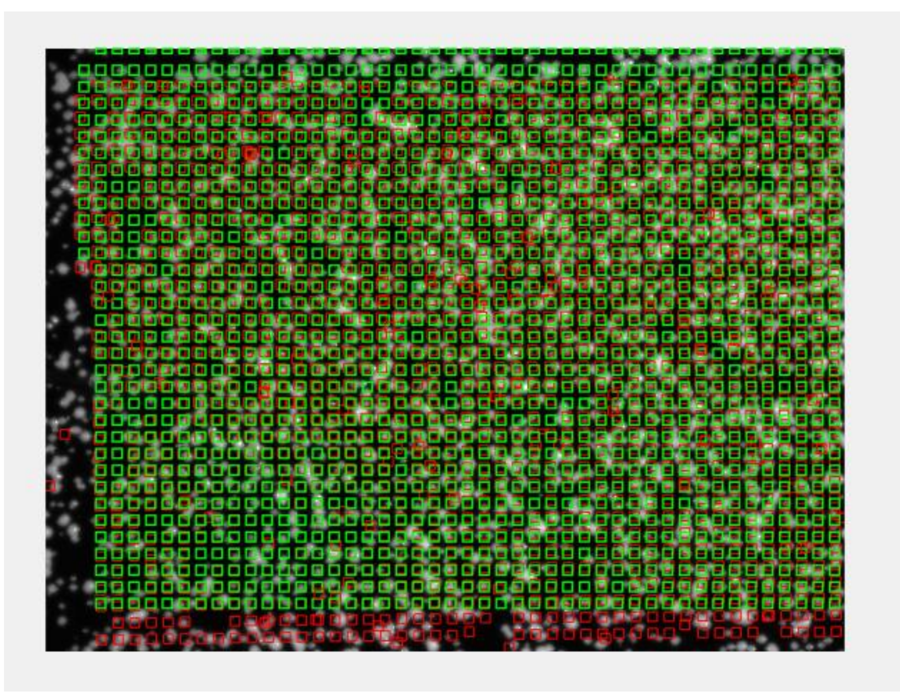
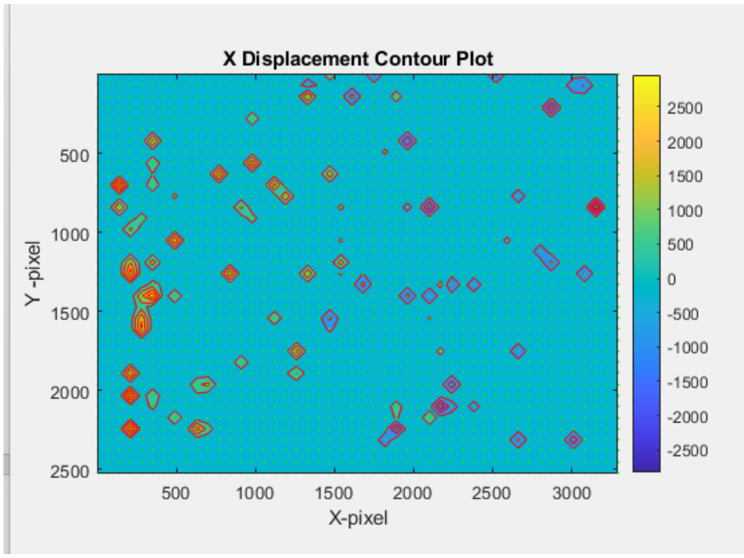
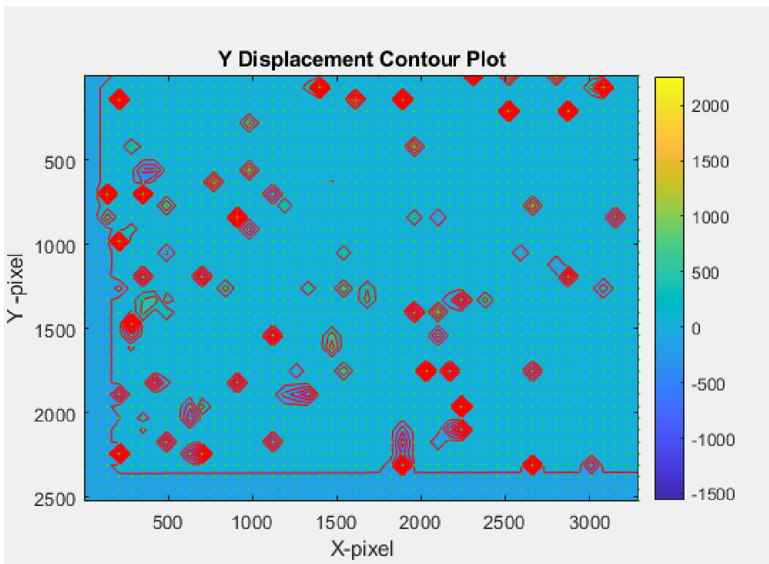
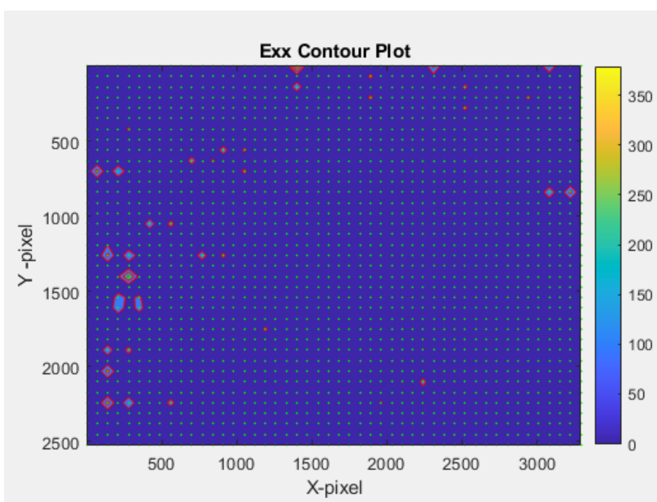
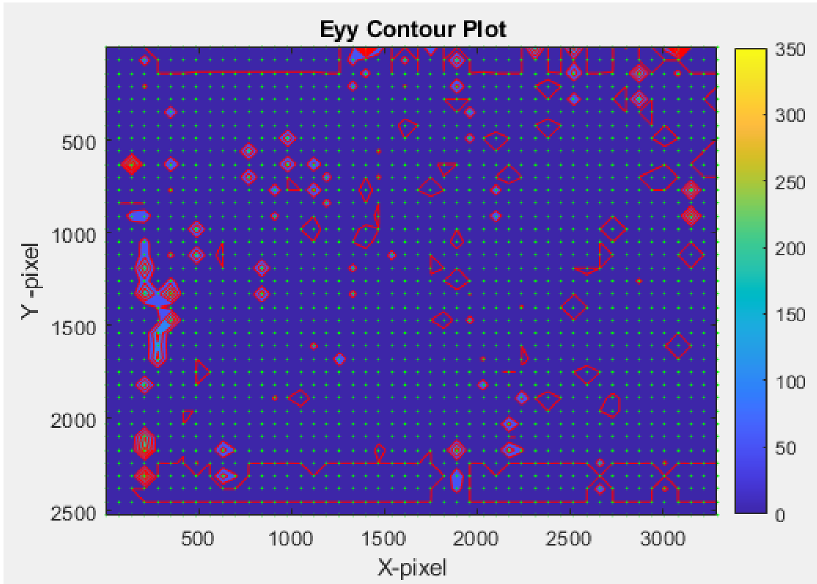


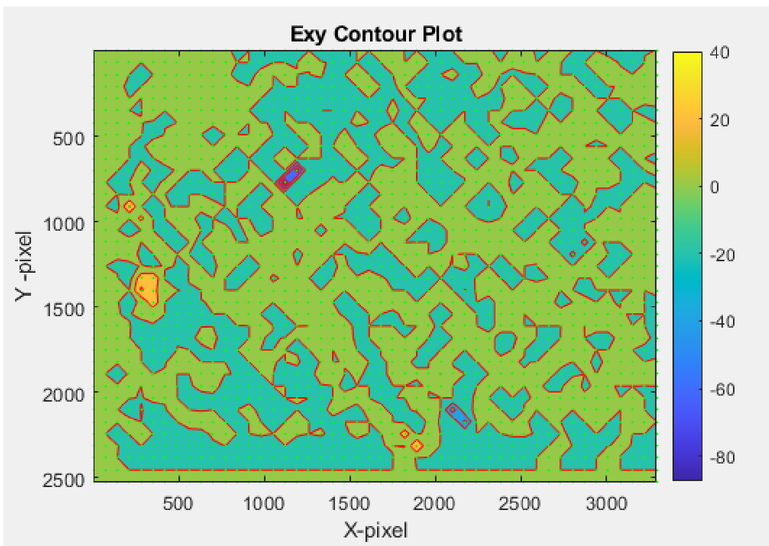
Figure: moved subset in current image


X Displacement Contour Plot
Y -pixel
X-pixel
500
1000
1500
2000
2500
3000
2500
2000
1500
1000
500
0
-500
-1000
-1500
-2000
-2500


Y Displacement Contour Plot
Y -pixel
X-pixel
500
1000
1500
2000
2500
3000
2000
1500
1000
500
0
-500
-1000
-1500


Exx Contour Plot
Y -pixel
X-pixel
500
1000
1500
2000
2500
3000
350
300
250
200
150
100
50
0





This is a big image. So big subset would be taken for quick results (this code runtime was more than one hour using 4 core). Based on the spackling pattern it might get better results if 100*100 subset size. Due to the small subset, some subsets were completely black which find its best matching position outside the ROI and give some abrupt results compared to the Ncorr code.

4. Summery and Recommendations

This project was an endeavor to understand the structure, and function of DIC as well as get used to the Matlab image processing toolbox. Sometimes results were accurate, sometimes results created a lot of questions. For better results, it is recommended to use Ncorr website handler Matlab code or other commercial DIC software. However, the improvement of this project code would be-

1. Creating an option for both taking exclusion or not in the code for ROI. Sometimes there is no need for excluding any region from mesh-grid.
2. Adding an option to check whether the input images are in RGB or not. If it is not RGB it will take the necessary steps to convert it to RGB else it will proceed with the input.
3. Instead of manual drawing for exclusion, it might be automated based on grayscale value Which is a better option.
4. During Normxcorr2 operation, a subset might be taken from the current image for searching peak which will reduce run time. The center of both subsets must be the same. This will reduce the problem that sometimes a subset moves abruptly and goes to an undeformed black region.
5. Adding an option so that the program will ask to give the input in the terminal so that no manual edit is necessary for steps one and two.
6. Adding cpcorr function to refine the displacement in pixel range.
7. Adding different smoothing functions for better plots of different output.
8. Add debug output so that the program stops and resumes at a point of error and after fixing the error problem runs from that point not from the beginning.
9. Discretize the program into different functions which is always preferred.
10. Fixing the issue during the validation process and getting the results in the same range described on the Ncorr website.

5. Conclusion

If the improvement points are ensured then there will be no output results error during validation steps and the code will give a more accurate result compared to commercial DIC software.

Appendix

Main code

```
%Digital Image corelation(DIC) project
%submitted by Hasan-W10389619
%this code will exclude multiple regin from gridpoints
%this number may be chnged in code by changing counter number
%this same code would be modifed in step 2 (one secton would be supressed)
%and in step 3(2 line would) to apply for no exclusion consideration
%whole code is in 2d, No data flattening

clc
clearvars

%%%%%%%%%%Step 1: Collecting file%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%try to read the reference and last current image to matlab

R = imread('referenceimage.tif');%R=reference image=referenceimage.tif
C = imread('deformedimage.tif');%C=current image=deformedimage.tif

%%%%%%%%%%Step 2: Exclude Region and define ROI %%%%%%%%%

%%%%delete/supress this for no exclusion-start, follow line where the stop
indication%%%%%%%%%
figure;
imshow(R);
title('Select the region to be excluded in the grid');

%create a mask to exclude the undeformed region
mask_for_exclude = false(size(R, 1), size(R, 2));

% Loop for interactive selection, replace N with the number of regions to exclude
%total 3 or 4 or 5 regeion will be excluded
%double click after completing and enclosed region

for i = 1:5% the second value might be equal to number of excluded area
    h = drawfreehand('Label', ['Region ' num2str(i)]);
    wait(h); % hold until all selection is completed
    mask = createMask(h); % Create a mask from the selection
    mask_for_exclude = mask_for_exclude | mask; % Combine the current mask with the
previous masks
end

%ROI=region of interest
ROI= ~mask_for_exclude;

%showing region of interest
figure;
imshow(R);
hold on;
h = imshow(~ROI);  % Assuming false in roiMask means exclusion
set(h, 'AlphaData', 0.3);  % Making the mask semi-transparent
```

```
title('Excluded and ROI on reference image');
%%%%%%%%%delete/supress this for no exclusion-end%%%%%%%


%%%%%%%%%%%%%%%%%%%Step 3: Meshgrid & initial variable for Iteration-
Plot%%%%%%%%%%%%%%%%

%low grid_spacing give more accurate results but increase computational cost
grid_spaceing=20; %two grid points distance
[rows,columns] = size(R);
[xGrid, yGrid] = meshgrid(1:grid_spaceing:columns, 1:grid_spaceing:rows);


figure('Name','Reference Image Grid','NumberTitle','off');
ax = axes; % Create axes in the figure
imshow(R, 'Parent', ax); % Display the image in these axes

figure('Name','Moved subset in deformed image','NumberTitle','off');
ay = axes; % Create axes in the figure
imshow(C, 'Parent', ay); % Display the image in these axes


w=10; % Width-horizontal length-cartesian x
h=10; %height-vertical lenght-cartesian y

R_template_size=[w,h];

x_displacement = zeros(size(xGrid));
y_displacement = zeros(size(yGrid));
final_displacement =zeros(size(xGrid));

%%%%%%%%%%%%%%%%%Step 4: normxcorr2: displacement%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Iterate through the grid points
%xGrid and y%Grid have the same dimension (2D)
for i = 1:size(xGrid, 1)
    for j = 1:size(xGrid, 2)
        %if ROI(i, j)  % If the grid point is within the ROI
        %if ~mask_for_exclude(round(yGrid(i)), round(xGrid(i)))
        %if ROI(round(yGrid(i)), round(xGrid(j)))-ok
        %if ROI((yGrid(i)), (xGrid(j)))
            x=xGrid(i,j);%
            y=yGrid(i,j);%

            % Define the template from the undeformed image
            xMin = max(1, x - R_template_size(1)/2);
            xMax = min(columns, x + R_template_size(1)/2 -1);
            yMin = max(1, y - R_template_size(2)/2);
            yMax = min(rows, y + R_template_size(2)/2-1);

            % Ensure the entire template is within the ROI
         if all(ROI(yMin:yMax, xMin:xMax), 'all')  % This line would be
deleted/supress during no exclusion
            R_template = R(yMin:yMax, xMin:xMax);
```

```
            % Perform cross-correlation and find the peak(c) in each iteration
            c = normxcorr2(R_template, C);
            [ypeak, xpeak] = find(c==max(c(:)));

            %subset were defined from middle position so as the yoffset
            yoffSet = ypeak-(size(R_template,1)/2)+1;
            xoffSet = xpeak-(size(R_template,2)/2)+1;

            % Calculate the displacement
            x_displacement(i, j) = xoffSet - x;
            y_displacement(i, j) = yoffSet - y;

            final_displacement(i,j) =sqrt(x_displacement(i,j).^2
+y_displacement(i,j).^2);

            %plotting subset in the terminal

            %plot of grid of R
            rectangle(ax, 'Position', [x, y, size(R_template,2), size(R_template,1)],
'EdgeColor', 'g','LineWidth', .5);

            %plot of deformed grid position on c
            rectangle(ay, 'Position', [xoffSet, yoffSet, size(R_template,2),
size(R_template,1)], 'EdgeColor', 'r','LineWidth', .5);

            %plot of grid and deformed in current coordinate
            rectangle(ay, 'Position', [x, y, size(R_template,2), size(R_template,1)],
'EdgeColor', 'g','LineWidth', 1);


        end%% This line would be deleted/supress during no exclusion
    end
end

%%%%%%%%%%%%%%%%%Step 5: Displacement Plot %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%all displacement plot as well as the next strain plot can hanldle the zero
displaement or same dsiplacement in each point plot without showing any error.
%so if this code searh standard deviation of the data before activate countourf
function than it can handle almost all validation process without error
%using countourf function of matlab, one of the requisite is that all input array
must be in same dimension
% no variation on array value will not defined in countourf

%xdisplacement plot
fprintf('The x_displacement array are:');
X_Displacement= x_displacement

if std(x_displacement(:) ~= 0)

     %standard Deviation(std) must not be zero for countourf plot
     %carefully select ROI so that no undeformed region is chosen.
     %check workspace varible of the corresponding date whether the deviation is zero
or not
```

```
     % at least one value of x_displacement must not be zero for plotting active
countourf
     % i.e no displacement - no graph.No error during valiadadation process with same
image, translation, rotation of the image
     % Manually define contour levels around the range of interest for better
visualization

     figure('Name','xdisplacement contour plot','NumberTitle','off');
     levels = linspace(min(x_displacement(:)), max(x_displacement(:)), 10); % see thw
workspace variable for x displacement value range. if it is small use low number
     contourf(xGrid, yGrid, x_displacement, levels, 'LineColor', 'r');
     colorbar; % Adds a color bar to indicate the displacement values
     title('X Displacement Contour Plot');
     %colormap('parula'); % This can make small differences more visible colormap
such as jet, hot, parula.
     xlabel('X-pixel');
     ylabel('Y -pixel');
     axis equal; % original size
     set(gca, 'YDir', 'reverse'); %reversing y direction
     hold on; % Retain the contour plot
     plot(xGrid, yGrid, 'g+', 'MarkerSize', .5, 'LineWidth', .5); % green plus is the
R_sub centre in C
     hold off; % Release the hold to prevent further additions
else
    % x_displacement is constant,
    disp('x displacement all value are zero or constant no standard Deviation in
data: Contour plot not generated.');
end

%ydisplacement plot

fprintf('The y_displacement array are:');
Y_Displacement= y_displacement

if std(y_displacement(:) ~= 0)
     %see xdisplacement same line for understaing different line of this loop
     figure('Name','ydisplacement contour plot','NumberTitle','off');
     levels = linspace(min(y_displacement(:)), max(y_displacement(:)), 20);
     contourf(xGrid, yGrid, y_displacement, levels, 'LineColor', 'r');
     colorbar; % Adds a color bar to indicate the displacement values
     title('Y Displacement Contour Plot');
     %colormap('hot'); % This can make small differences more visible colormap such
as jet, parula, hot,
     xlabel('X-pixel');
     ylabel('Y -pixel');
     axis equal; % original size
     set(gca, 'YDir', 'reverse'); %reversing y direction
     hold on; % Retain the contour plot
     plot(xGrid, yGrid, 'g+', 'MarkerSize', .5, 'LineWidth', .5); % green plus is the
R_sub centre in C
     hold off; % Release the hold to prevent further additions
else
    % y_displacement is constant, help eliminate error during validation
    disp('all x displacement point value is zero or constant or no Standard
Deviation: Contour plot not generated.');
```

```
end

%final displacement plot

fprintf('The final_displacement array are:');
Final_Displacement= final_displacement

if std(final_displacement(:) ~= 0)
     %  see xdisplacement for loop to  understand different line of this loop
     figure('Name','final_displacement contour plot','NumberTitle','off');
     levels = linspace(min(final_displacement(:)), max(final_displacement(:)), 20);
     contourf(xGrid, yGrid, final_displacement, levels,'LineColor', 'r');
     colorbar; % Adds a color bar to indicate the displacement values
     title('final_displacement Contour Plot');
     %colormap('hot'); % This can make small differences more visible colormap such
as jet, parula, hot,
     xlabel('X-pixel');
     ylabel('Y -pixel');
     axis equal; % original size
     set(gca, 'YDir', 'reverse'); %reversing y direction
     hold on; % Retain the contour plot
     plot(xGrid, yGrid, 'g+', 'MarkerSize', .5, 'LineWidth', .5); % green plus is the
R_sub centre in C
     hold off; % Release the hold to prevent further additions
else
    % y_displacement is constant,
    disp('final_displacement is constant  or no standart deviation : Contour plot not
generated.');
end

%%%%%%%%%%%%%%%%%Step 6: Deformatino Gradient & jacobian %%%%%%%%%%%

%Spatial Gradient of displacement field

[dx_dx, dx_dy]=gradient(x_displacement,grid_spaceing,grid_spaceing);%here
grid_spaceing in x and y direction
[dy_dx, dy_dy]=gradient(y_displacement,grid_spaceing,grid_spaceing);

%initial 2*2 zeros matrix to hold deformation gradient(F)of each point

[a,b]=size(xGrid);
F=zeros(2,2,a,b);

for m=1:size(xGrid,1)
    for n=1:size(xGrid,2)
        F(:,:,a,b)= [dx_dx(i,j), dx_dy(i,j);
                  dy_dx(i,j), dy_dy(i,j)];
    end

end

fprintf('The Deformation Gradient array or Tensor (F) are:');
Tensor_F= F

%Jacobian Matrix
```

```
J=dx_dx.*dy_dy- dx_dy.*dy_dx;

fprintf('TheJacobian (J) are:');
Jacobian_J= J

%%%%%%%%%%%%step9: Lagrange strain & corresponding plot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Lagrange Strain(Exx,Eyy,Exy)

Exx=dx_dx+ 0.5*(dx_dx.^2+dy_dy.^2);
Eyy = dy_dy + 0.5*(dx_dy.^2 + dy_dy.^2);
Exy = 0.5*(dx_dy + dy_dx) + 0.5*(dx_dx.*dx_dy + dy_dx.*dy_dy);

%Exx plot

fprintf('The Exx strain array  are:');
Lagrange_Exx=Exx

if std(Exx(:) ~= 0)
     %see xdisplacement comments to understand each line of this loop
     figure('Name','Exx contour plot','NumberTitle','off');
     levels = linspace(min(Exx(:)), max(Exx(:)), 5);
     contourf(xGrid, yGrid, Exx, levels, 'LineColor', 'r');
     colorbar; % Adds a color bar to indicate the displacement values
     title('Exx Contour Plot');
     %colormap('hot'); % This can make small differences more visible colormap such
as jet, parula, hot,
     xlabel('X-pixel');
     ylabel('Y -pixel');
     axis equal; % original size
     set(gca, 'YDir', 'reverse'); %reversing y direction
     hold on; % Retain the contour plot
     plot(xGrid, yGrid, 'g+', 'MarkerSize', .5, 'LineWidth', .5); % green plus is the
R_sub centre in C
     hold off; % Release the hold to prevent further additions
else
    % Exx is constant,
    disp('Exx all value is constant or zeros no standard deviation: Contour plot not
generated.');
end

%Eyy plot
fprintf('The Eyy strain array  are:');
Lagrange_Eyy=Eyy

if std(Eyy(:) ~= 0)
     %see xdisplacement comments to understand each line of this loop
     figure('Name','Eyy contour plot','NumberTitle','off');
     levels = linspace(min(Eyy(:)), max(Eyy(:)), 5);
     contourf(xGrid, yGrid, Eyy, 'LineColor', 'r');
     colorbar; % Adds a color bar to indicate the displacement values
     title('Eyy Contour Plot');
```

```
     %colormap('hot'); % This can make small differences more visible colormap such
as jet, parula, hot,
     xlabel('X-pixel');
     ylabel('Y -pixel');
     axis equal; % original size
     set(gca, 'YDir', 'reverse'); %reversing y direction
     hold on; % Retain the contour plot
     plot(xGrid, yGrid, 'g+', 'MarkerSize', .5, 'LineWidth', .5); % green plus is the
R_sub centre in C
     hold off; % Release the hold to prevent further additions
else
    % Eyy  is constant,
    disp('Eyy  all value is constant(zeros): Contour plot not generated.');
end

%Exy plot

fprintf('The Exy strain array  are:');
Lagrange_Exy=Exy

if std(Exy(:) ~= 0)
     % if all data of the Exx is like of its first data
     % i.e no strain - no graph.No error during valiadation process with same image,
translation, rotation of the image
     figure('Name','Exy contour plot','NumberTitle','off');
     levels = linspace(min(Exy(:)), max(Exy(:)), 5);
     contourf(xGrid, yGrid, Exy, 'LineColor', 'r');
     colorbar; % Adds a color bar to indicate the displacement values
     title('Exy Contour Plot');
     %colormap('jet'); % This can make small differences more visible colormap such
as jet, parula, hot,
     xlabel('X-pixel');
     ylabel('Y -pixel');
     axis equal; % original size
     set(gca, 'YDir', 'reverse'); %reversing y direction
     hold on; % Retain the contour plot
     plot(xGrid, yGrid, 'g+', 'MarkerSize', .5, 'LineWidth', .5); % green plus is the
R_sub centre in C
     hold off; % Release the hold to prevent further additions
else
    % Exy  is constant,
    disp('Exy  all value is constant(zeros) or  no standard deviation: Contour plot
not generated.');
end
```