# 1 Electrical Impedance Tomography (EIT)

# 2 Implementation of the backprojection algorithm

## Contents

```
clear; close all;
```

## Load in data and Set Parameters

```
load('project2data.mat')

N       = 16;
a       = 10.0;
I       = 1/a;
epislon = 1e-9;

phi = linspace(0,2*pi-(2*pi/N),N);
```

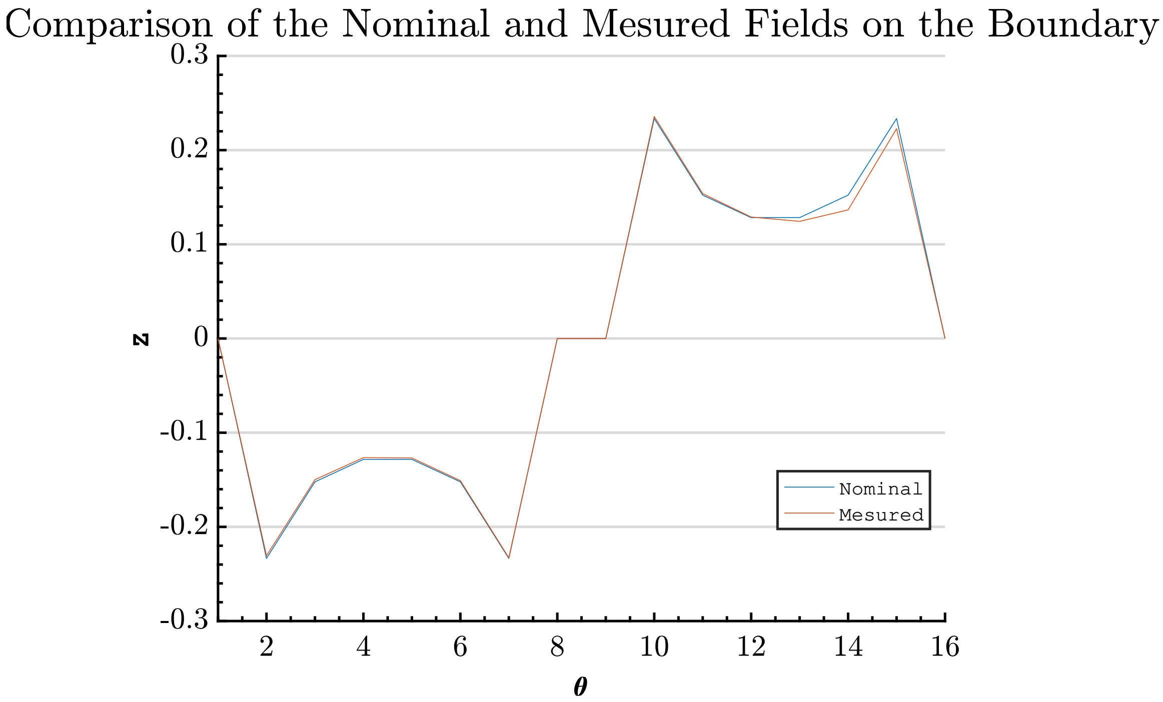
## Build Boundary Conditions

```
uBoundary = 1/(2*pi) * log( ( 1 + cos(phi) ) ./ ( 1 - cos(phi) ) );
uBoundary =  uBoundary - circshift(uBoundary,1) ;

for k = 1:length(uBoundary)
    if ( isinf(uBoundary(k)) )
        uBoundary(k) = epislon;
    end

    if (isnan(uBoundary(k)))
        uBoundary(k) = epislon;
    end
end
```

```
uBoundary = circshift(uBoundary,-1);
```

### Compare Measurement and Nominal Fields

```
plot(uBoundary);
hold on
plot(zMat(:,1));
```

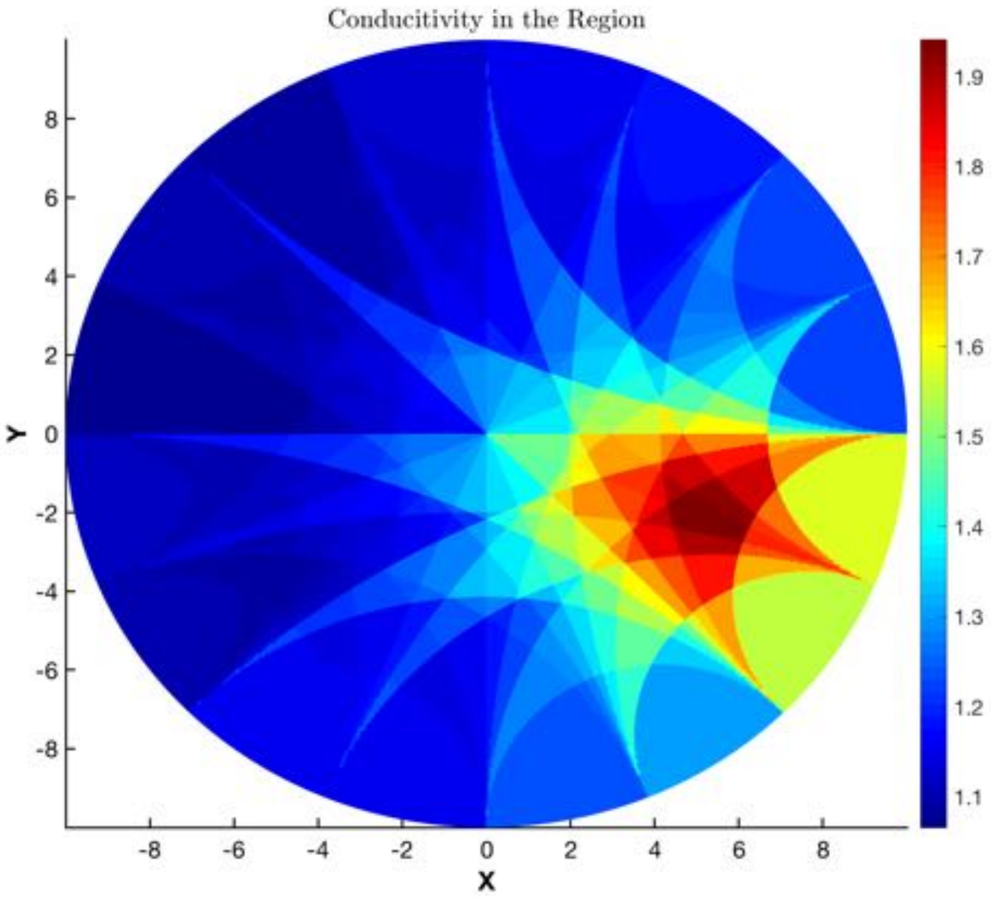


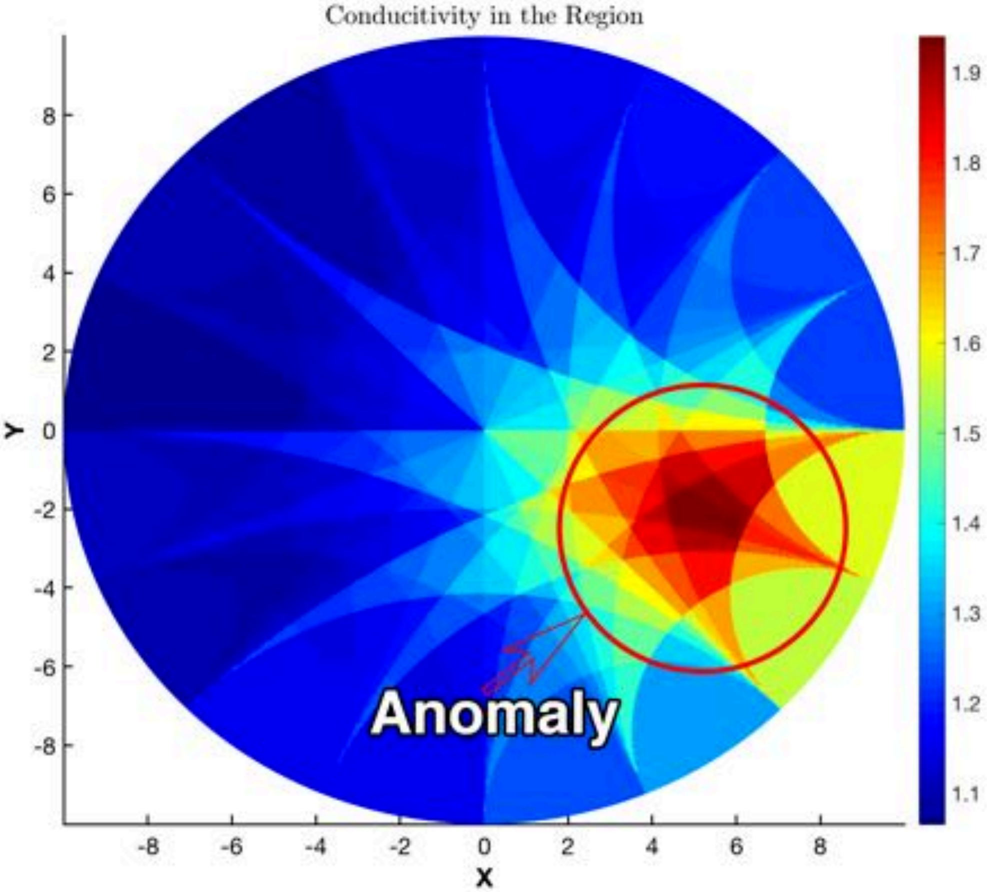
### Implement Level Set Relations

```
steps   = 10^3;
r       = linspace(0, a, steps);
deltaT  = 2*pi/N;
theta   = linspace(deltaT, 2*pi+deltaT, steps);
sigma   = ones(length(r), length(theta));
xplot = r' * cos(theta);
yplot = r' * sin(theta);

f2 = figure(2);
```

## Back Projection Algorithm

```
for k = 1:N/2
    theta = theta - deltaT;
    x = r' * cos(theta);
    y = r' * sin(theta);

    X = 2 * a^2 .* x ./ (x.^2 + y.^2 + a^2);
    Y = sqrt(a^2 - X.^2);

    for i = 1:length(r)
        for j = 1:length(theta)
            alpha = atan2( Y(i,j) , X(i,j) );

            if ( alpha < 0)
                alpha = alpha + 2*pi;
            end

            [c, index] = min( abs( phi-alpha ) );

            if( ( phi(index) - alpha) > 0 )
                indexDelta = index - 1;

                if indexDelta == 0
                    indexDelta = 16;
                end
                indexDelta2 = 17 - indexDelta;
            end
            if (  (phi(index) - alpha) <= 0  )
                indexDelta  = index;
                indexDelta2 = 17 - indexDelta;
            end
            deltaSigTemp = -(zMat(indexDelta, k) / uBoundary(indexDelta) - 1);
            deltaSig2 = -(zMat(indexDelta2, k) / uBoundary(indexDelta2) - 1);

            sigma(i,j) = sigma(i,j) + deltaSigTemp + deltaSig2;

        end

    end
    pcolor(xplot, yplot, sigma);
    shading interp;
    pause(0.3);
end
```




Conducitivity in the Region
Y
8
6
4
2
0
-2
-4
-6
-8
X
-8
-6
-4
-2
0
2
4
6
8
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
Anomaly