I have written a code that will calculate the equation of the best fit-line for a given set of data-points.

The procedure is based on least square approximation, which, in simple words,works by finding a line that is at a minimum distance possible from all the points.

Let's say you have the x-axis points stored in a matrix, 'x' & the y-axis points stored in a matrix 'y'. Then the following code returns the value of the slope, 'm' and intercept 'c' of the best fit-line, and also the fitted points which are basically the points of the y-axis obtained from the equation of the best fit-line.

CODE:

```
//Linear Fit
//To fit a given set of data-points to a line.
//Written By: Manas Sharma(www.bragitoff.com)
funcprot(0);
function [f,m,c]=linefit(x,y)
    n=size(x);
    if n(2)<n(1) then
        n=n(2)
    else
        n=n(1);
    end
    xsum=0;
    ysum=0;
    xysum=0;
    x2sum=0;
    for i=1:n
        xsum=x(i)+xsum;
        ysum=y(i)+ysum;
        x2sum=x(i)*x(i)+x2sum;
        xysum=x(i)*y(i)+xysum;
    end
    m=(n*xysum-xsum*ysum)/(n*x2sum-xsum*xsum);
    c=(x2sum*ysum-xsum*xysum)/(x2sum*n-xsum*xsum);
    for i=1:n
        f(i)=m*x(i)+c;
    end
endfunction
```

Sample Demo:

```
x=[1,2,3,4,5];
y=[2,4,6,8,10];
[yfit,m,c]=linefit(x,y)
```

Output:
c =

0.
m =

2.
yfit =

2.
4.
6.
8.
10.

In this example the y-axis points were simply twice of the value of x-axis points. Hence the equation of line, we get is $y=mx+c=>y=2x$.

Now let's try something not so straight forward.

Sample Demo:

```
x=[1,2,3,4,5];
y=[0.5,2.5,3.3,4.1,5.5];
[yfit,m,c]=linefit(x,y)
```

Output:

c =

- 0.3
  m =

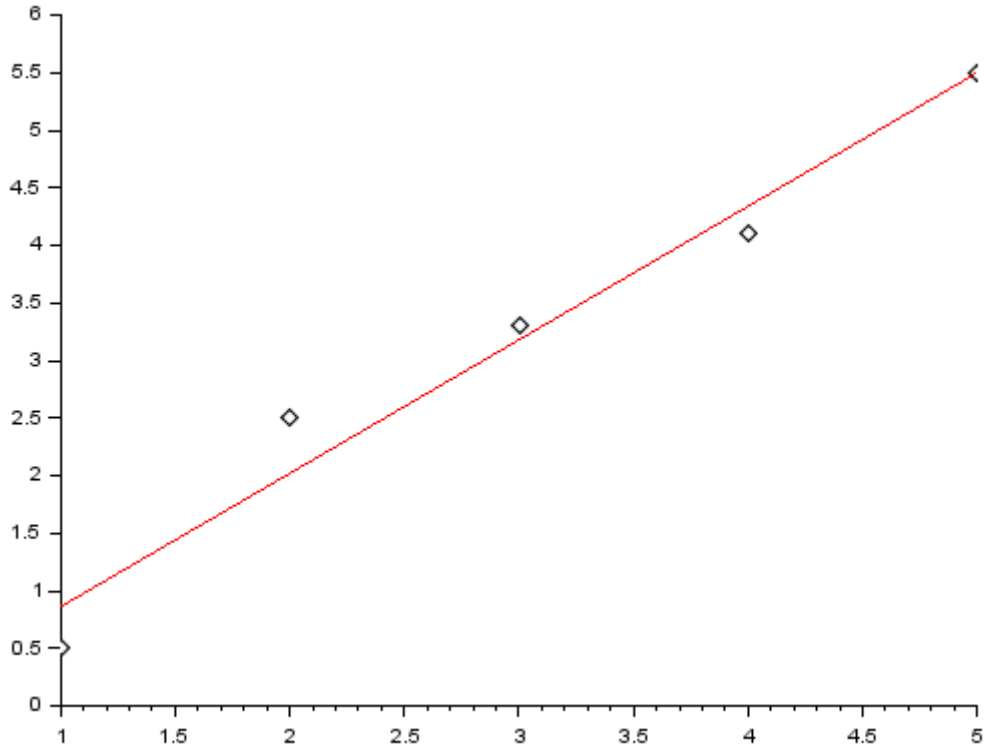
1.16
yfit =

0.86
2.02
3.18
4.34
5.5

We can now even plot the observed data-points and the fitted points(yfit) and compare the two.

```
plot2d(x,y,-5)  //dotted points for the original/observed data-points
plot2d(x,yfit,5) //red line for the fitted points
```

Output:



I have created a module in SCILAB which contains the above macro, and once installed can be used as an in-built function. You can download it from here: https://atoms.scilab.org/toolboxes/curvefit

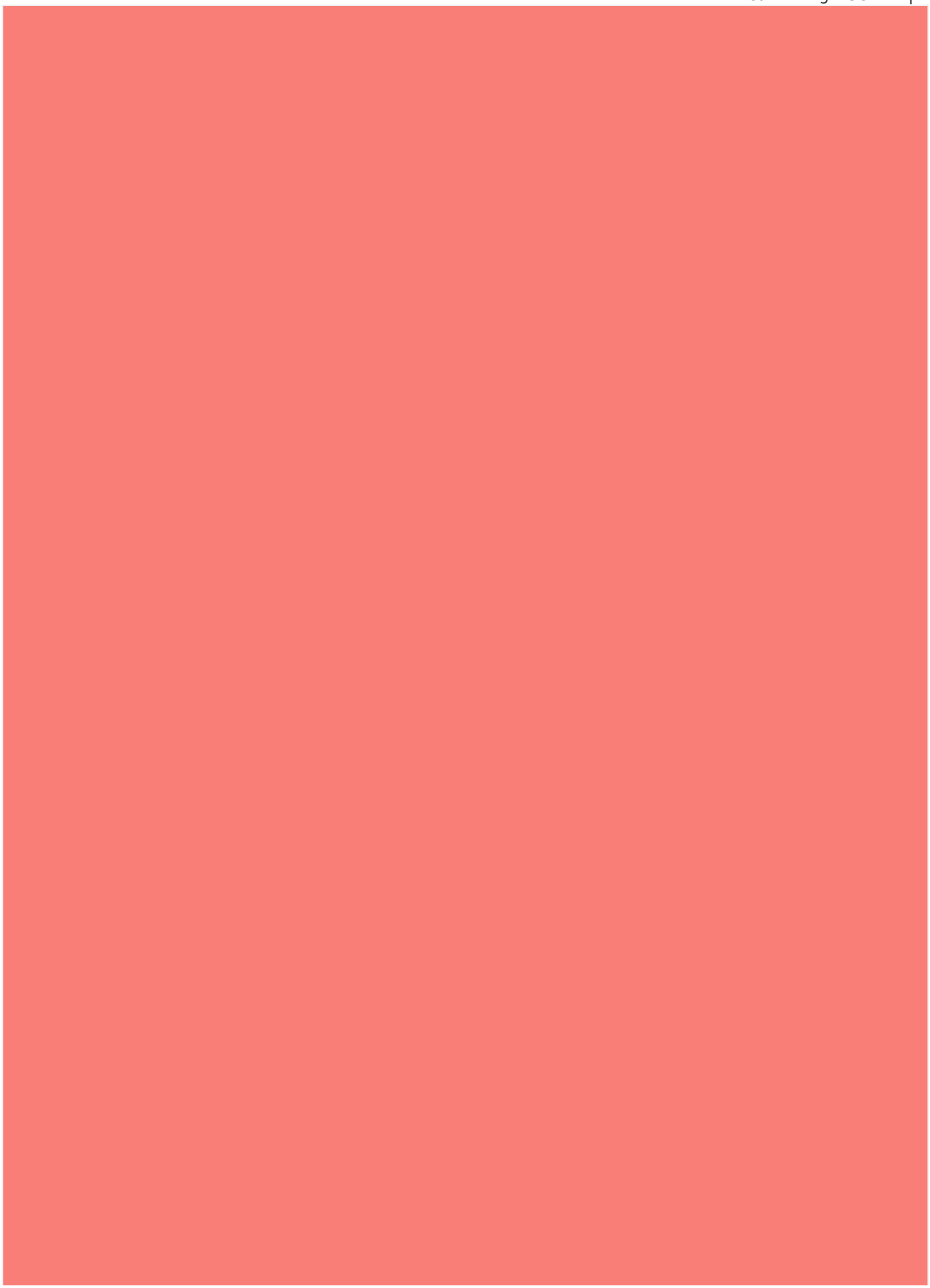
Leave your questions/suggestion/corrections in the comments section down below and I'll get back to you soon.

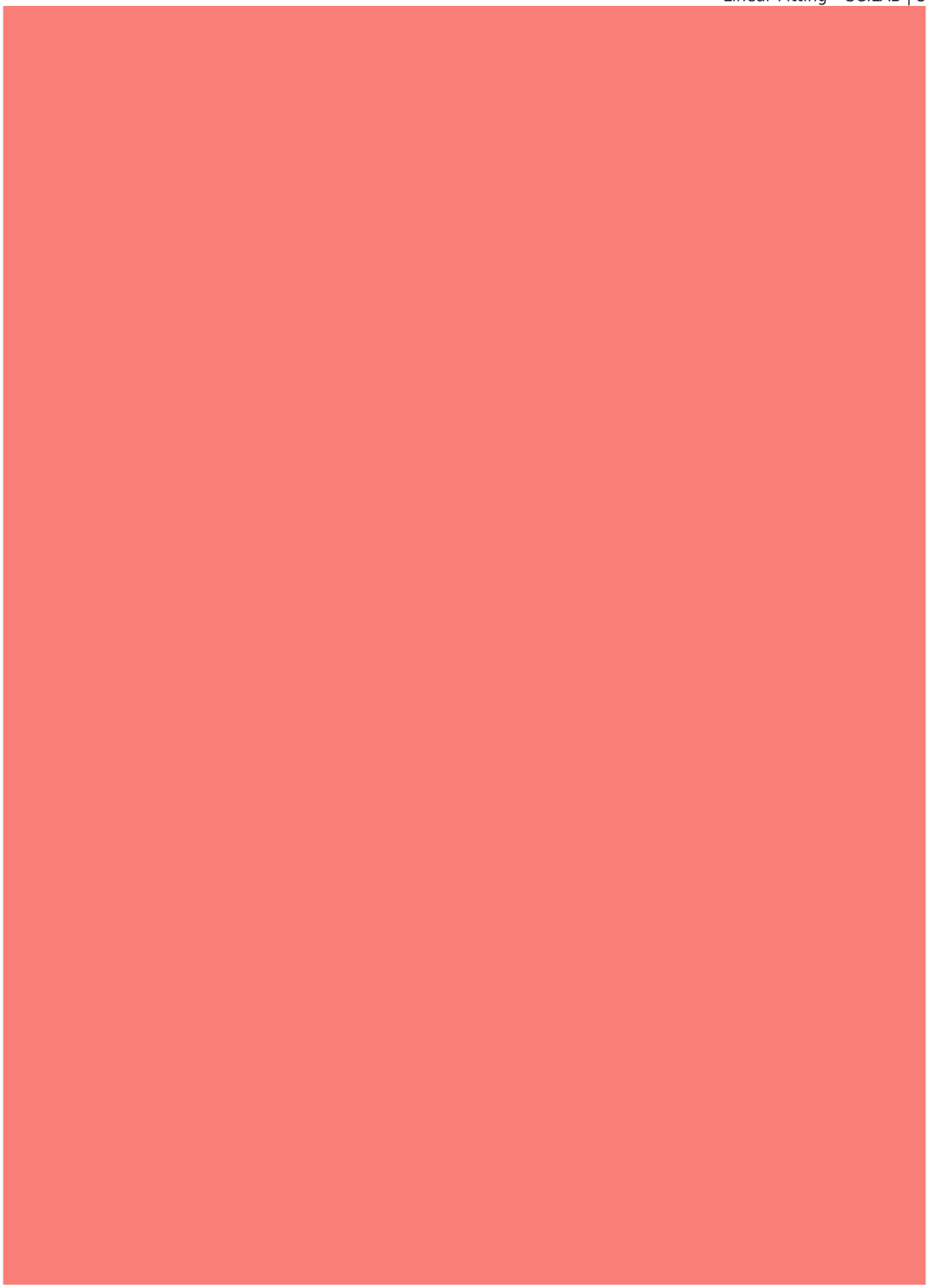


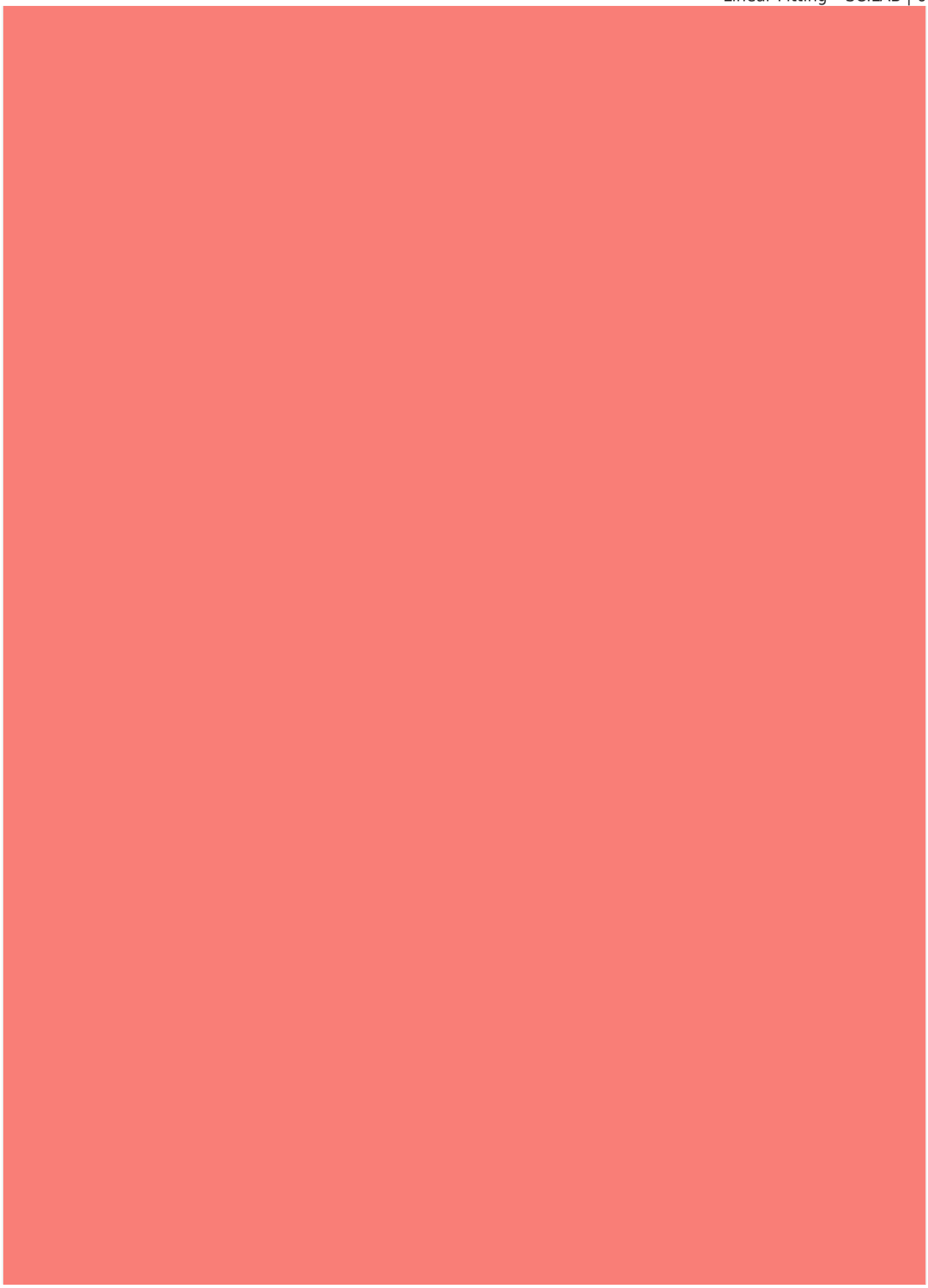
Manas Sharma

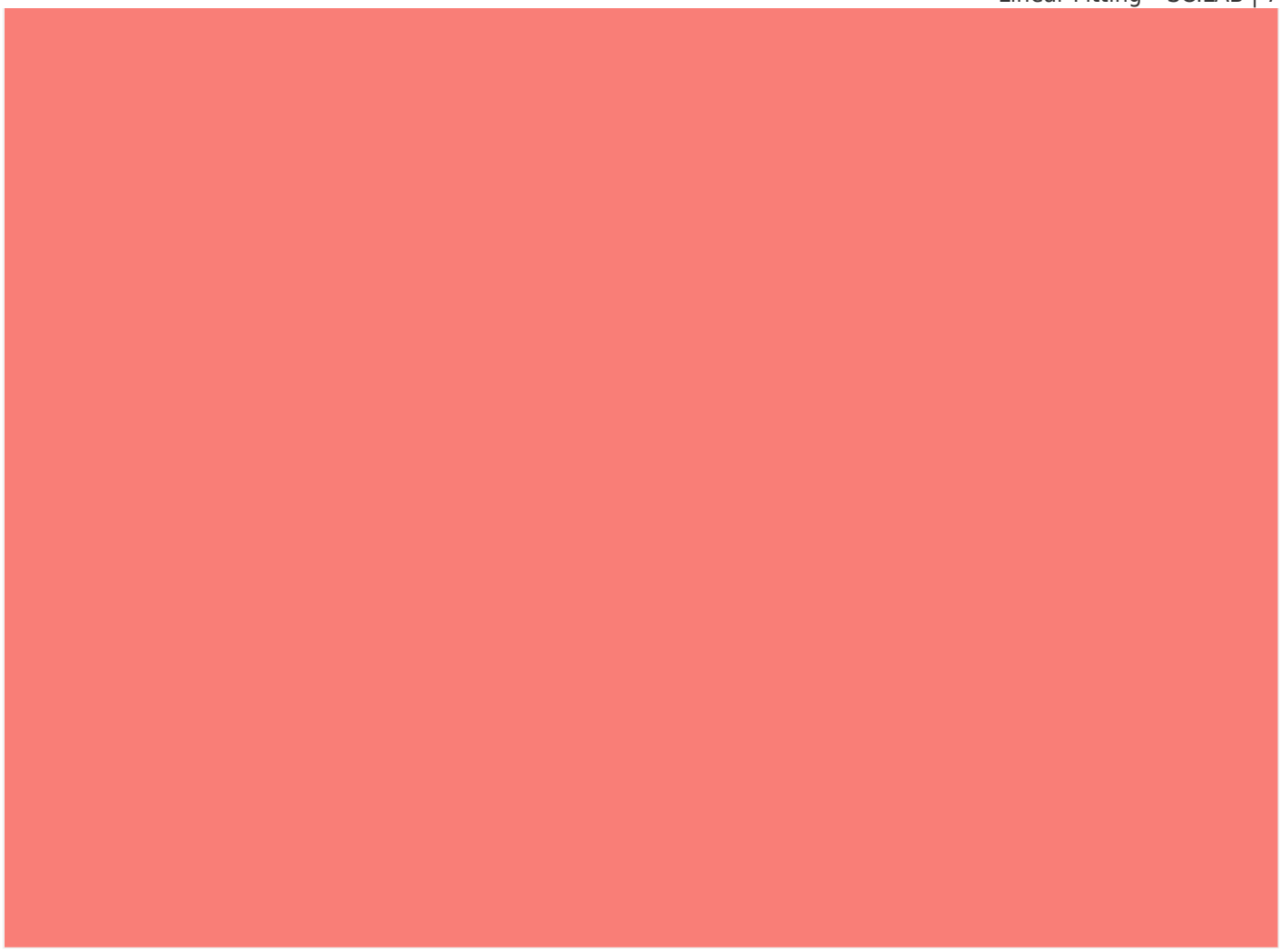
I'm a physicist specializing in computational material science with a PhD in Physics from Friedrich-Schiller University Jena, Germany. I write efficient codes for simulating light-matter interactions at atomic scales. I like to develop Physics, DFT, and Machine Learning related apps and software from time to time. Can code in most of the popular languages. I like to share my knowledge in Physics and applications using this Blog and a YouTube channel.

manas.bragitoff.com/









Share this:

Click to share on Facebook (Opens in new window)

Click to share on Twitter (Opens in new window)

Click to share on WhatsApp (Opens in new window)

Click to share on Pinterest (Opens in new window)

Click to share on Reddit (Opens in new window)

Click to share on LinkedIn (Opens in new window)

Click to email a link to a friend (Opens in new window)

Click to print (Opens in new window)

Click to share on Tumblr (Opens in new window)

Click to share on Pocket (Opens in new window)

Click to share on Telegram (Opens in new window)

[wpedon id="7041" align="center"]