In the last post I showed you guys how to calculate Fourier Coefficients for a given function defined in the range, [-l,l].

We can also use the Fourier Coefficients to calculate the Fourier Series and then Plot the FS Approximation and compare it to the original function.

I have used the same code as before and just added a few more lines of code.

If you haven't read the last post which is being continued here then I recommend you read that first here.

Code:

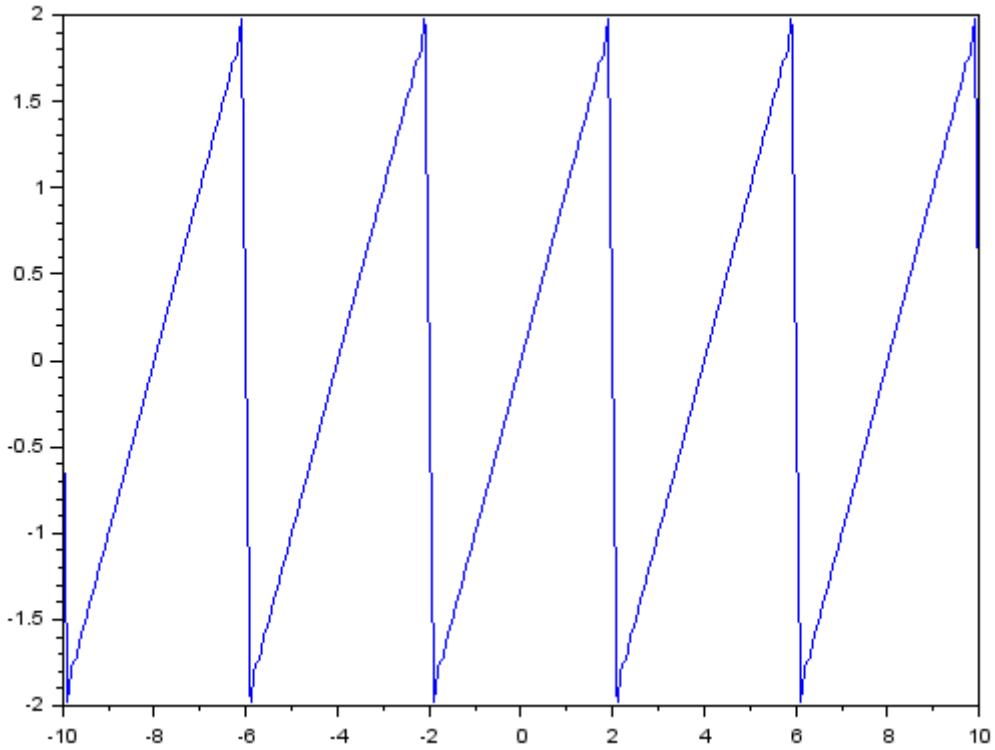
```
funcprot(0);
function [a0,A,B]=fourierplot(l,n,f)
    clf();
    a0=1/l*intg(-l,l,f,.000000001);
    for i=1:n
        function b=f1(x,f)
            b=f(x)*cos(i*%pi*x/l);
        endfunction
        function c=f2(x,f)
            c=f(x)*sin(i*%pi*x/l);
        endfunction
        A(i)=1/l*intg(-l,l,f1,.000000001);
        B(i)=1/l*intg(-l,l,f2,.000000001);
    end
    function series=solution(x)
        series=a0/2;
        for i=1:n
            series=series+A(i)*cos(i*%pi*x/l)+B(i)*sin(i*%pi*x/l);
        end
    endfunction
    x=-5*l:0.1:5*l;
    plot(x,solution(x));
endfunction
```

As you can notice, the code is almost the same as before, except that I have changed the name of the function from *'fourier'* to *'fourierplot'*. And in the previous post I had only calculated the Fourier Coefficients, a0, A, B, but this time I am using those to calculate the Fourier Series, through the function *'solution'*. And finally I have plotted the Fourier Series.

Sample Demo:

```
deff('a=f(x)','a=x');
[A0,A,B]=fourierplot(2,100,f)
```

Output:



In the above sample code I have defined a function f(x)=x, and then I have called the function fourierplot with arguments l=2, n=100 & the function f. l is the half of the period so the function is periodic with a period of 2l=4 from [-2,2], which is evident from the graph. If the value of n is not large enough then you might not get a very good approximation.

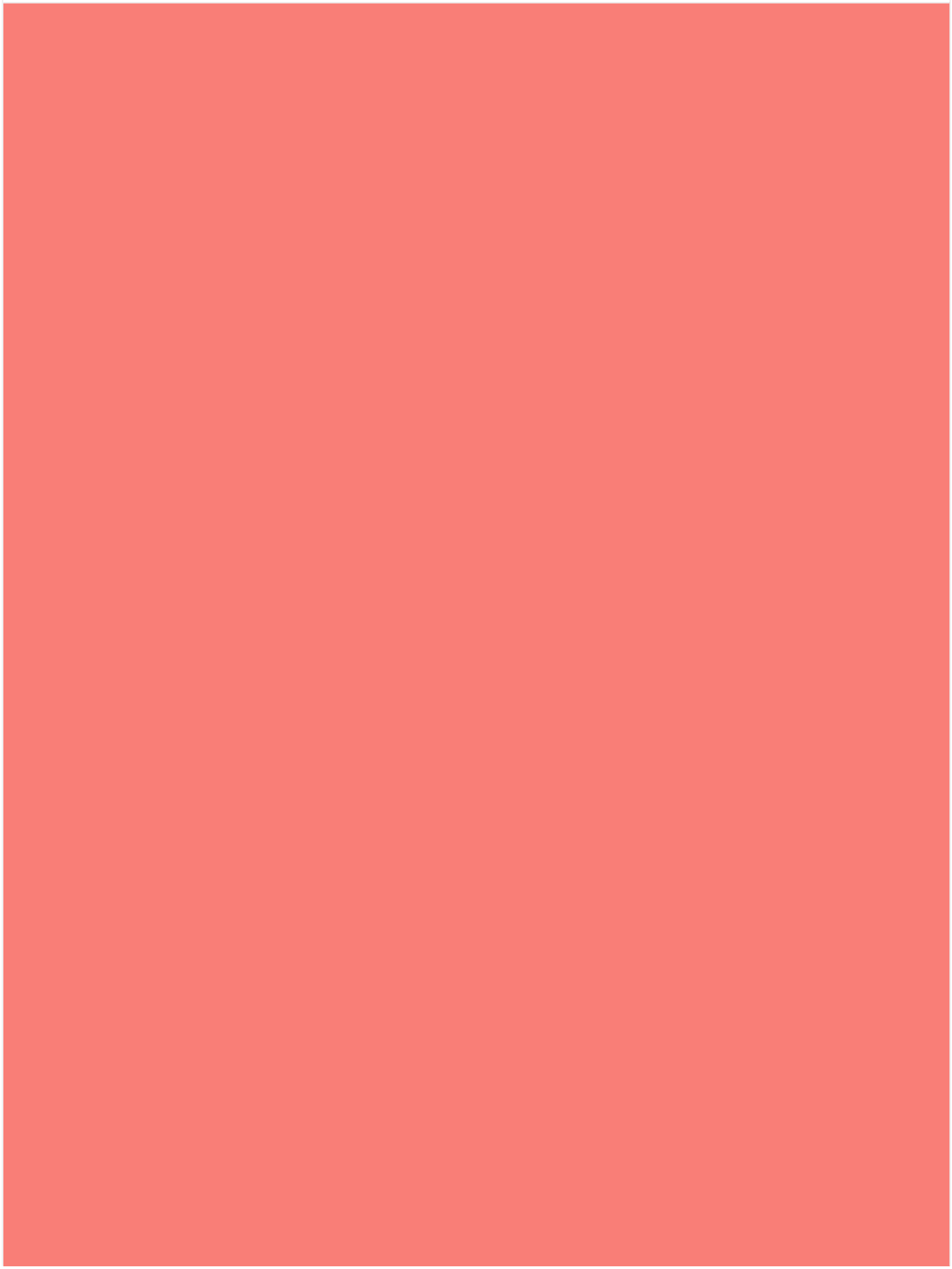
I have created a module in scilab which contains both the above macro, and once installed can be used as an in-built function. You can download it from here: https://atoms.scilab.org/toolboxes/fstools

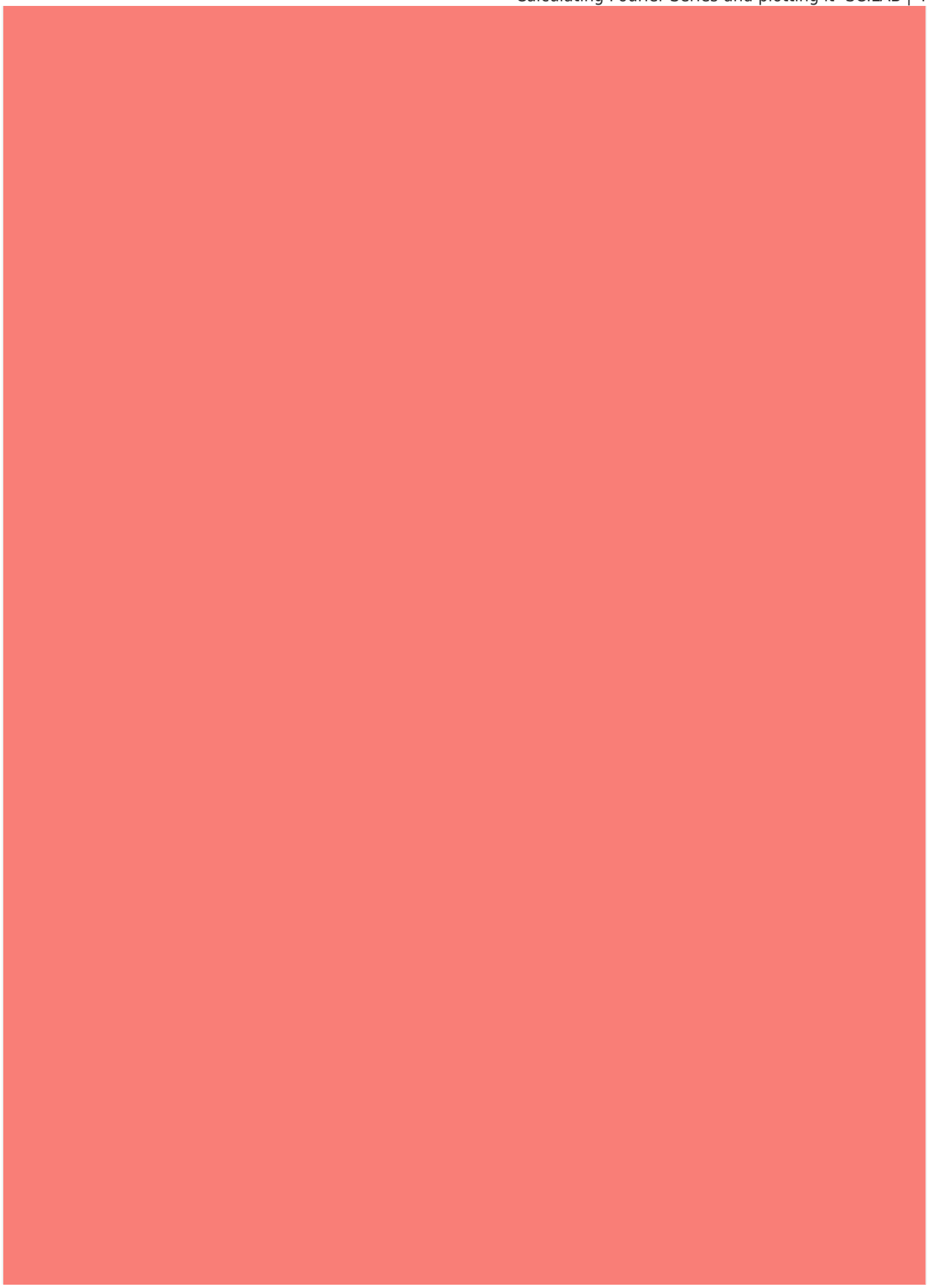
Tutorial:

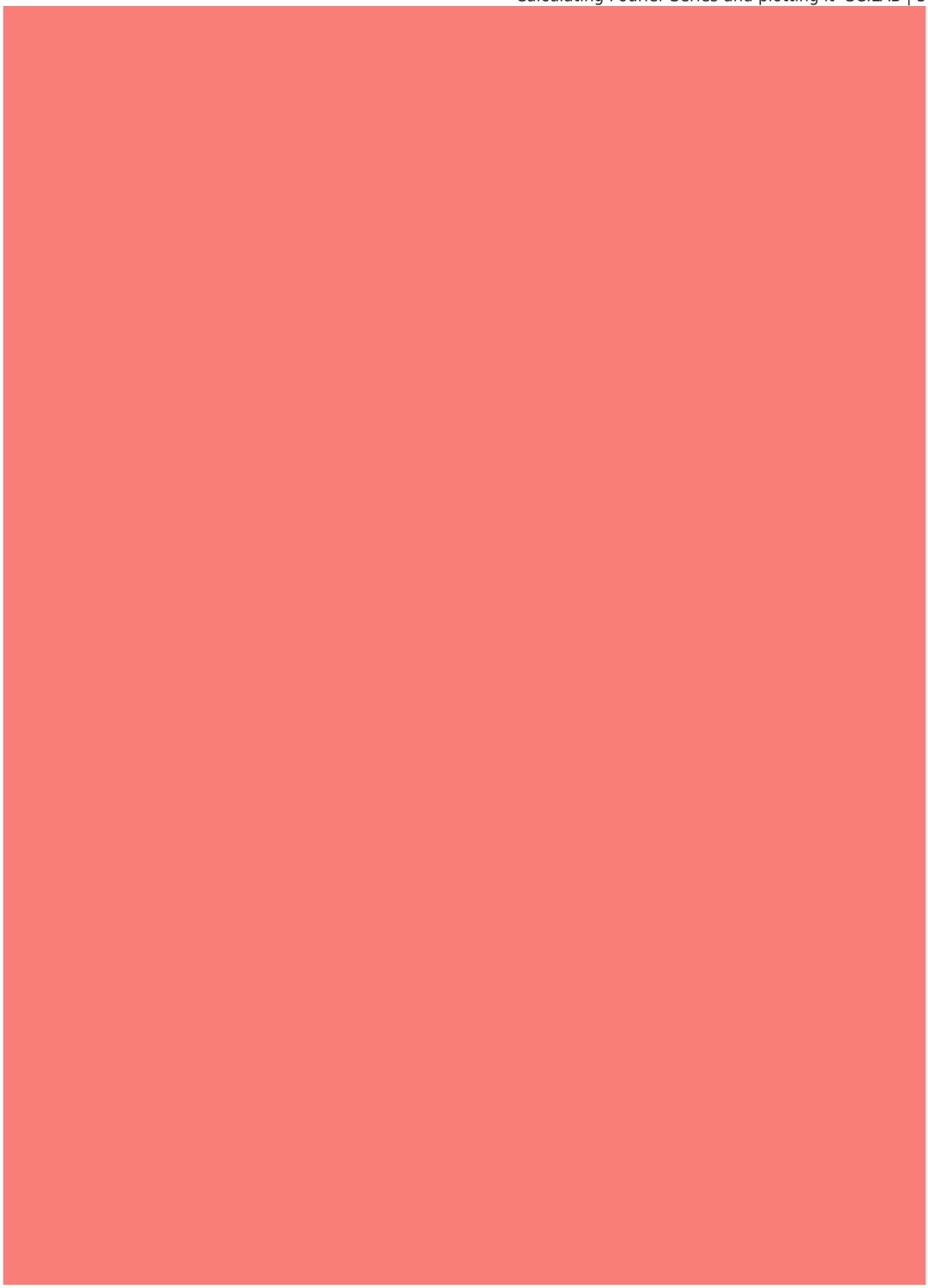
Leave your questions/suggestion/corrections in the comments section down below and I'll get back to you soon.

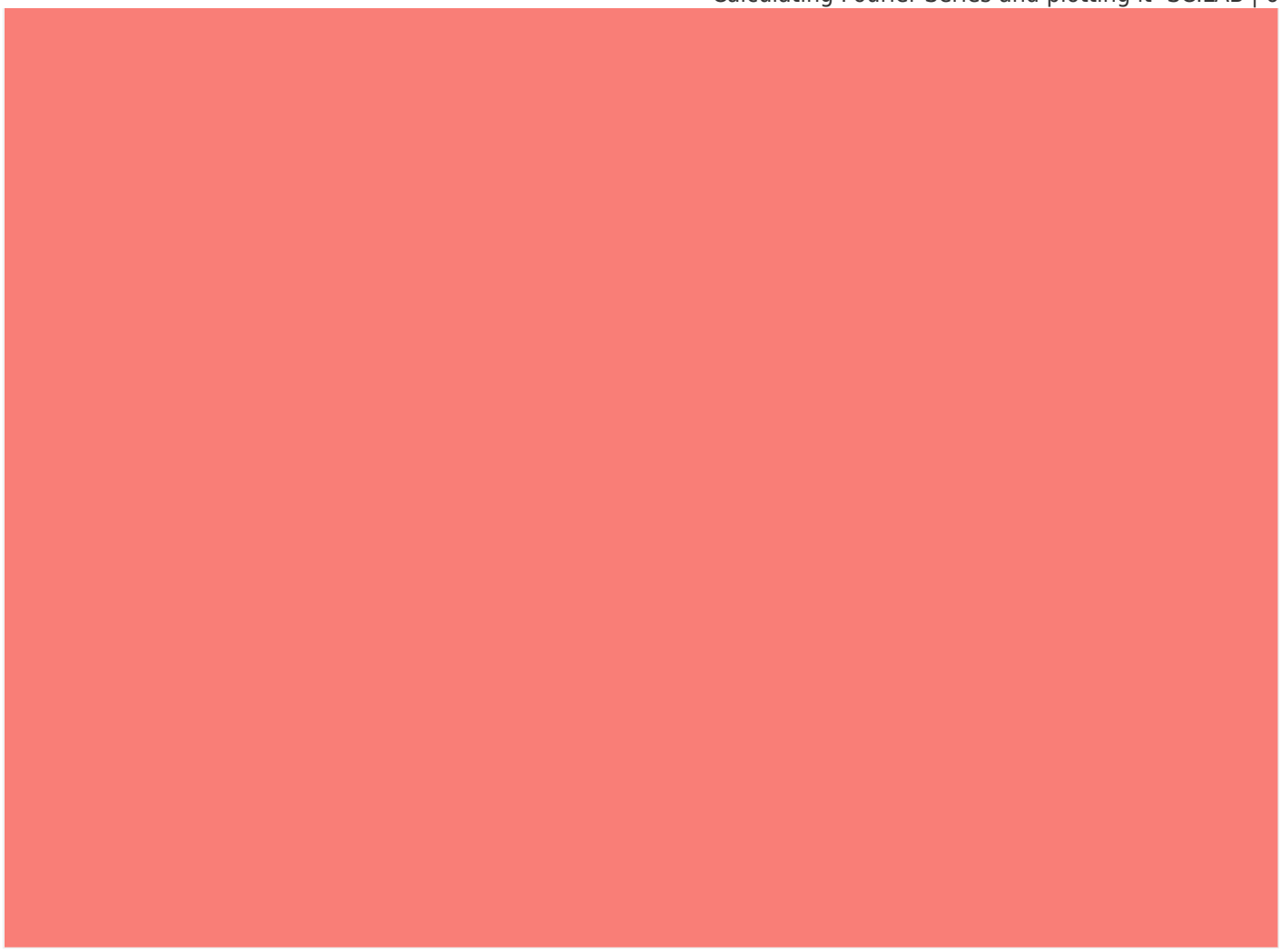
Manas Sharma

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.









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"]