Runge-Kutta Method is a numerical technique to find the solution of ordinary differential equations.

The second-order Runge-Kutta method uses the following formula:

$$k_1 = hf(x_i, y_i)$$
$$k_2 = hf(x_i + h/2, y_i + k_1/2)$$
$$y_{i+1} = y_i + k_2 + O(h^3)$$

The fourth-order Runge-Kutta method uses the following formula:

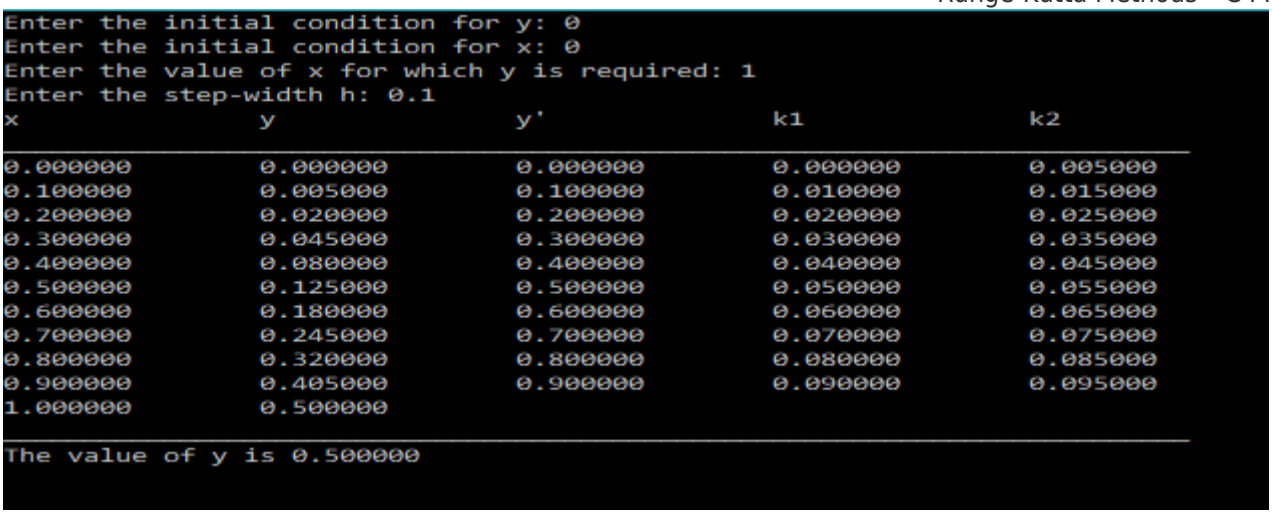
$$k_1 = hf(x_i, y_i)$$
$$k_2 = hf(x_i + h/2,_i + k_1/2)$$
$$k_3 = hf(x_i + h/2, y_i + k_2/2)$$
$$k_4 = hf(x_i + h, y_i + k_3)$$

The program for the second-order Runge-Kutta Method is shown below:

PROGRAM(RK II ORDER:

```
/************************************
*********RUNGE-KUTTA METHOD(1)*********
************************************/
#include<stdio.h>
#include<math.h>
/*Define the RHS of the first order differential equation here(Ex: dy/dx=f(x,y))  */
double f(double x, double y){
        //return 2-exp(-4*x)-2*y;
        //return x+y;
        return x;
}
main(){
        int i;
        double x,y,x0,y0,h,k1,k2;
        printf("Enter the initial condition for y: ");
        scanf("%lf",&y0);
        printf("Enter the initial condition for x: ");
        scanf("%lf",&x0);
        printf("Enter the value of x for which y is required: ");
        scanf("%lf",&x);
        printf("Enter the step-width h: ");
        scanf("%lf",&h);
        printf("x\t\ty\t\ty'\t\tk1\t\tk2\n");
printf("_____________________________________________________________________________\n");
        //Begin Runge-Kutta Routine
        while((x-x0)>0.0000000001){
                k1=h*f(x0,y0);
                k2=h*f(x0+h/2.0,y0+k1/2.0);
                y=y0+k2;
                printf("%lf\t%lf\t%lf\t%lf\t%lf\n",x0,y0,f(x0,y0),k1,k2);
                y0=y;
                x0=x0+h;
        }
        printf("%lf\t%lf\n",x0,y0);
printf("_____________________________________________________________________________\n");
        printf("The value of y is %lf\n\n",y);
}
```

OUTPUT:



The program for the fourth-order Runge-Kutta Method is shown below:

PROGRAM(RK 4th ODER):

```
/************************************
*********RUNGE-KUTTA METHOD(2)*********
************************************/
#include<stdio.h>
#include<math.h>
/*Define the RHS of the first order differential equation here(Ex: dy/dx=f(x,y))  */
double f(double x, double y){
        //return 2-exp(-4*x)-2*y;
        //return x+y;
        return x;
}
main(){
        int i;
        double x,y,x0,y0,h,k1,k2,k3,k4;
        printf("Enter the initial condition for y: ");
        scanf("%lf",&y0);
        printf("Enter the initial condition for x: ");
        scanf("%lf",&x0);
        printf("Enter the value of x for which y is required: ");
        scanf("%lf",&x);
        printf("Enter the step-width h: ");
        scanf("%lf",&h);
        printf("x\t\ty\t\tk1\t\tk2\t\tk3\t\tk4\t\tk_avg\n");
printf("_____________________________________________________________________________________
_____________________________\n");
        //Begin Runge-Kutta Routine
        while((x-x0)>0.0000000001){
                k1=h*f(x0,y0);
                k2=h*f(x0+h/2.0,y0+k1/2.0);
                k3=h*f(x0+h/2.0,y0+k2/2.0);
                k4=h*f(x0+h,y0+k3);
printf("%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\n",x0,y0,k1,k2,k3,k4,1/6.0*(k1+2*k2+2*k3+k4))
;
                y=y0+1/6.0*(k1+2*k2+2*k3+k4);
                y0=y;
                x0=x0+h;
        }
        printf("%lf\t%lf\n",x0,y0);
printf("_____________________________________________________________________________________
_____________________________\n");
        printf("The value of y is %lf\n\n",y);
}
```

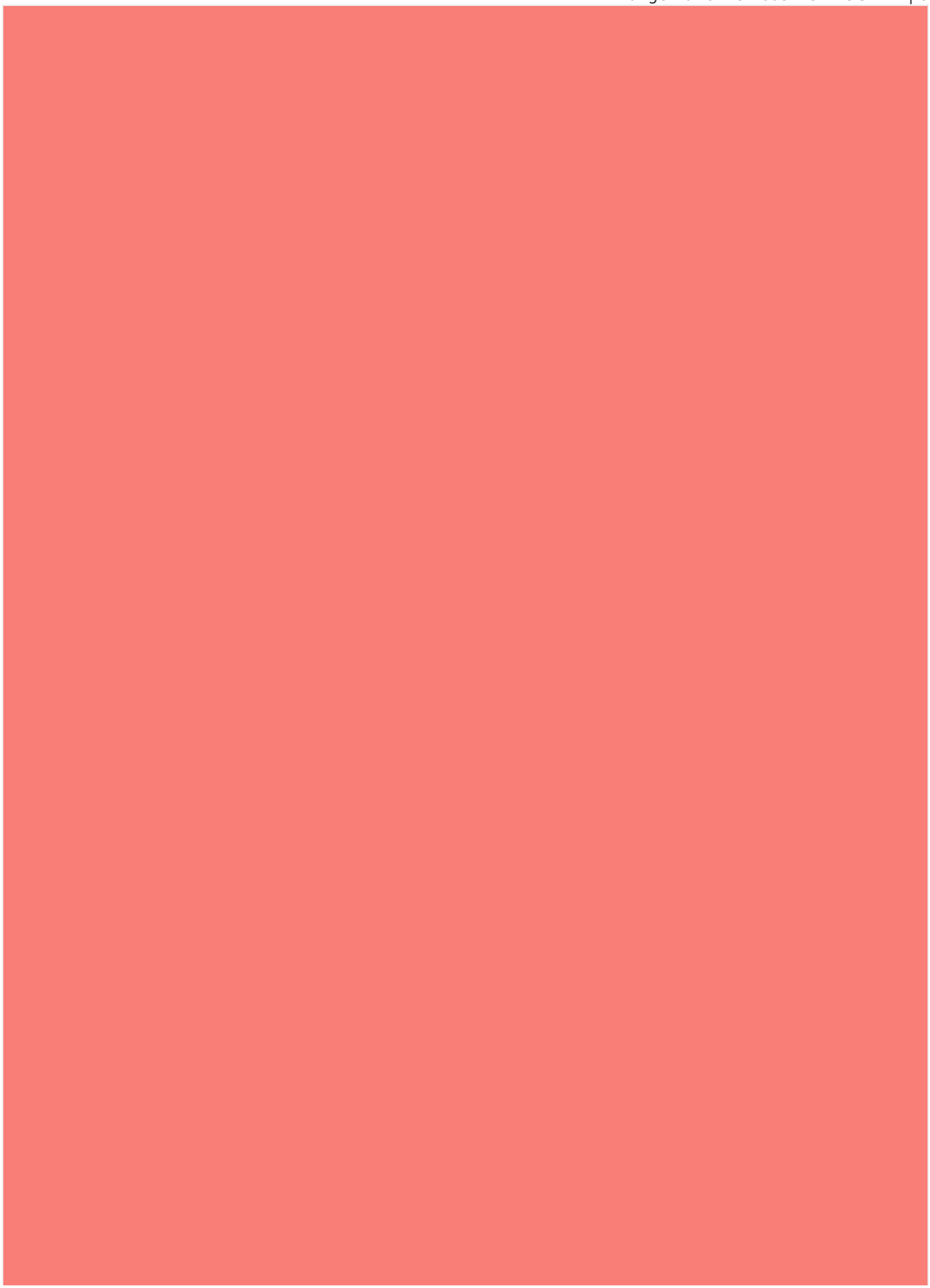
OUTPUT:

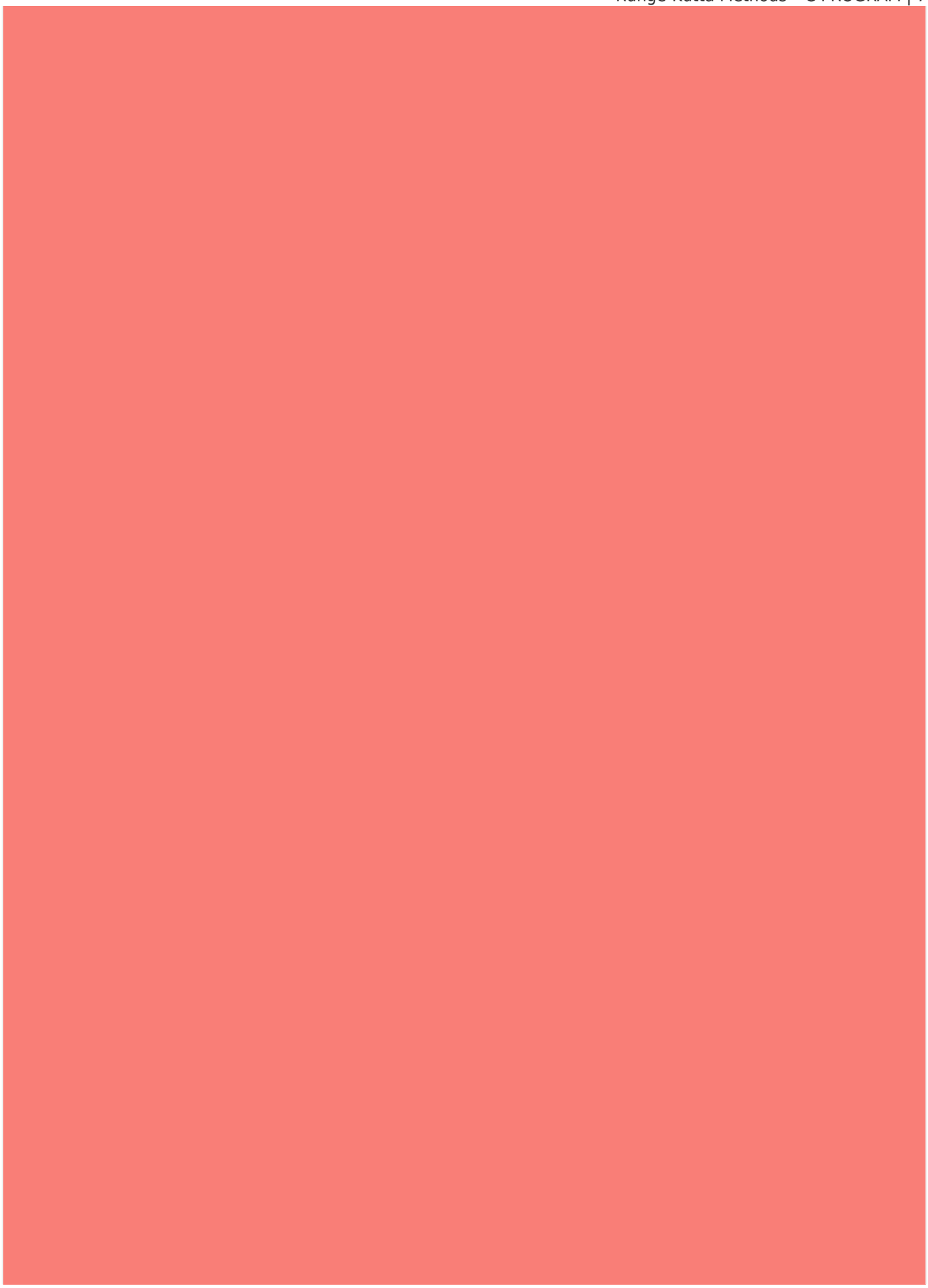
```
Enter the initial condition for y: 0
Enter the initial condition for x: 0
Enter the value of x for which y is required: 1
Enter the step-width h: 0.1
x               y               k1              k2              k3              k4              k_avg
-------------------------------------------------------------------------------------------------------
0.000000        0.000000        0.000000        0.005000        0.005000        0.010000        0.005000
0.100000        0.005000        0.010000        0.015000        0.015000        0.020000        0.015000
0.200000        0.020000        0.020000        0.025000        0.025000        0.030000        0.025000
0.300000        0.045000        0.030000        0.035000        0.035000        0.040000        0.035000
0.400000        0.080000        0.040000        0.045000        0.045000        0.050000        0.045000
0.500000        0.125000        0.050000        0.055000        0.055000        0.060000        0.055000
0.600000        0.180000        0.060000        0.065000        0.065000        0.070000        0.065000
0.700000        0.245000        0.070000        0.075000        0.075000        0.080000        0.075000
0.800000        0.320000        0.080000        0.085000        0.085000        0.090000        0.085000
0.900000        0.405000        0.090000        0.095000        0.095000        0.100000        0.095000
1.000000        0.500000
-------------------------------------------------------------------------------------------------------
The value of y is 0.500000
```

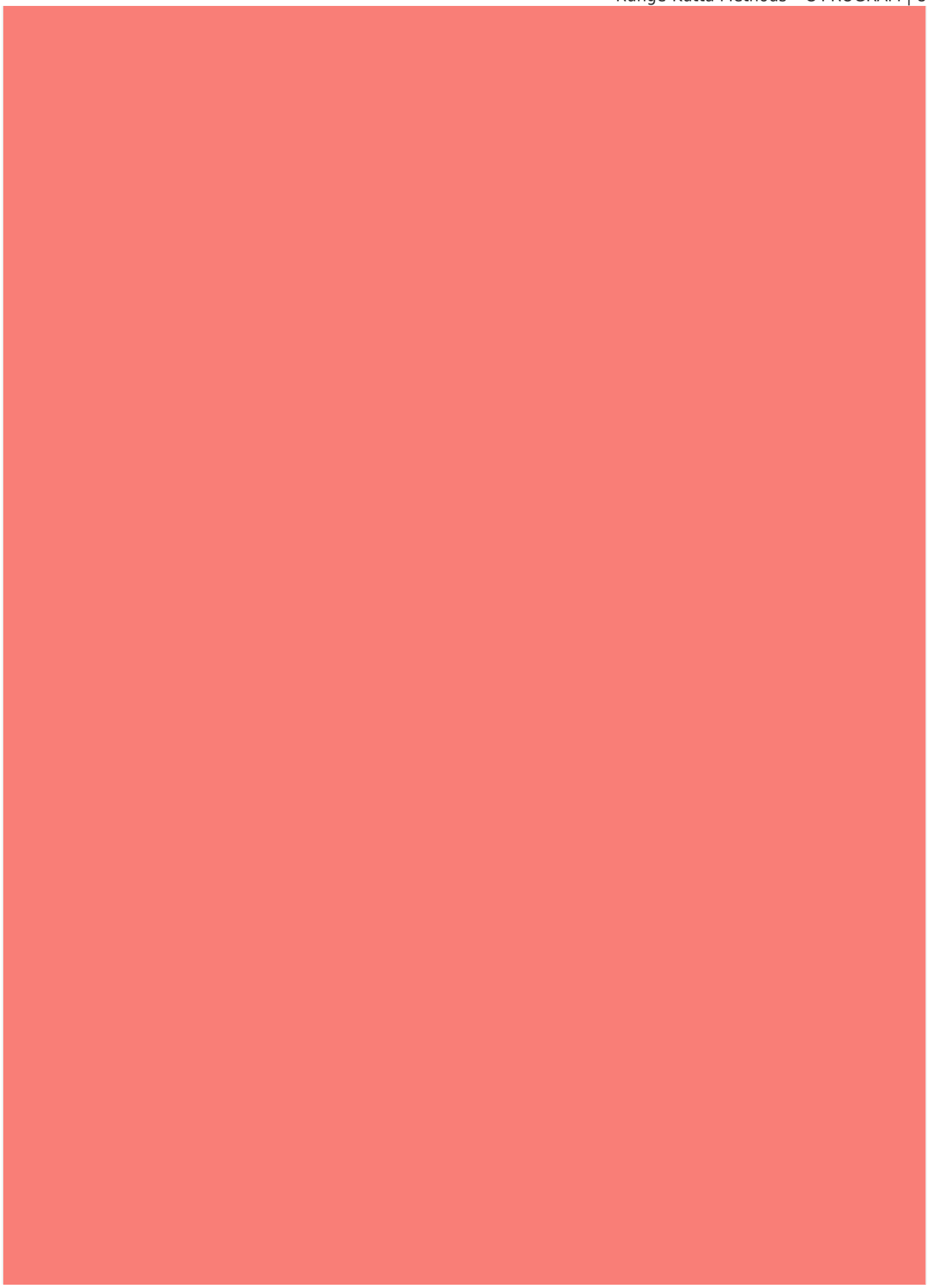
Manas Sharma

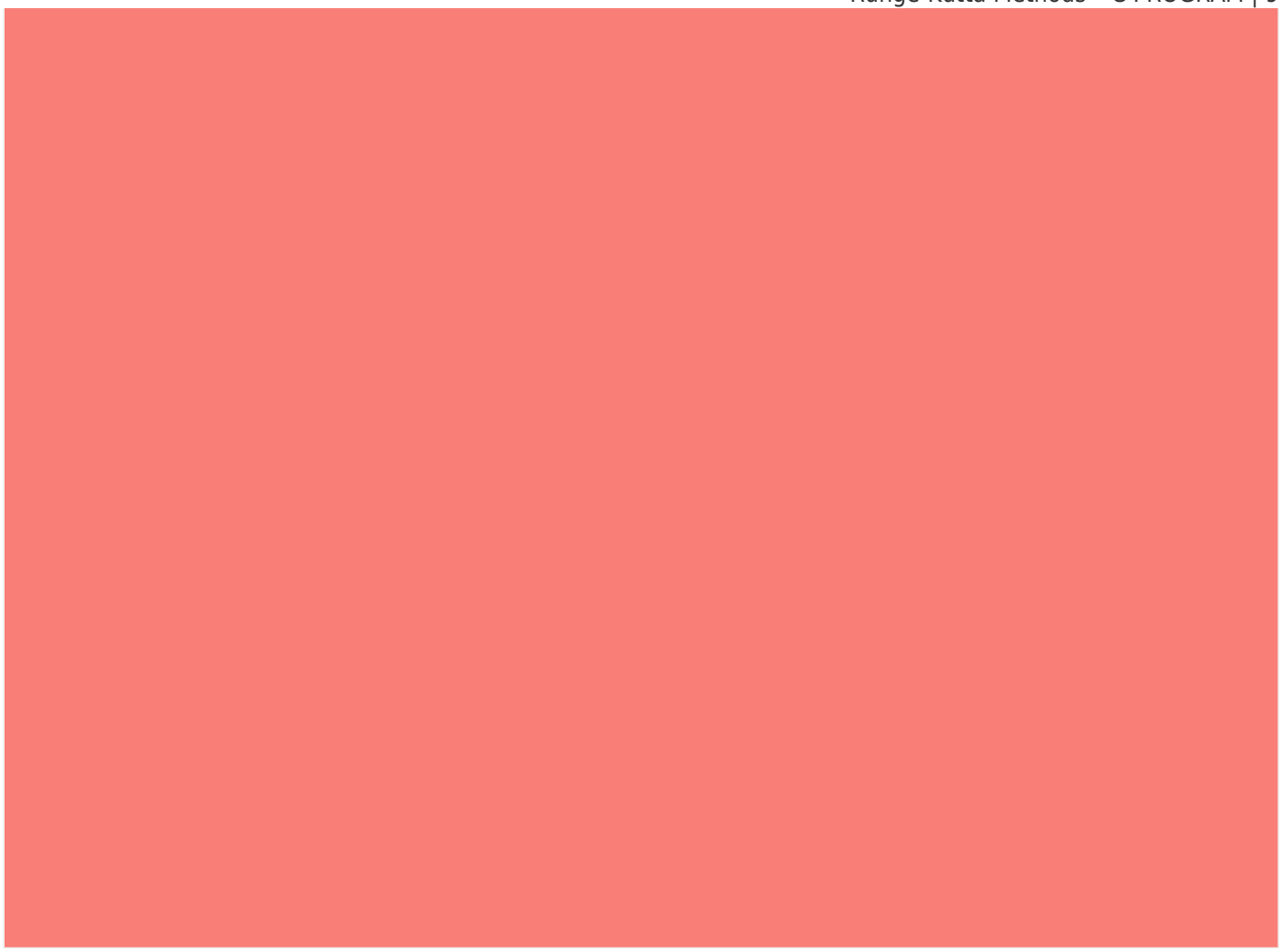
I’m a physicist specializing in computational material science with a PhD in Physics from Friedrich-Schiller University Jena, Germany. I write efficient codes for simulating light-matter interactions at atomic scales. I like to develop Physics, DFT, and Machine Learning related apps and software from 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"]