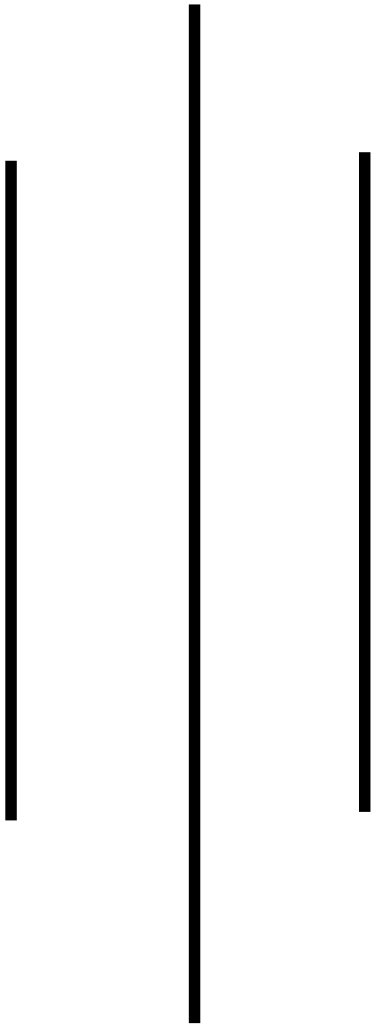
Deerwalk Institute Of Technology

Sifal, Kathmandu



Simulation and Modeling
Practical

Submitted By:
Name:
Roll No:
Section:

Submitted To:
Binod Sitaula
DWIT

Date:

Background Theory

For solution of ordinary differential equations, we implement different methods and measure its accuracy with comparison with the exact solution. Some methods are: Euler's method, Heun's method, Runge-Kutta methods.
We use different formulas for mentioned methods to calculate the expected answer. Among the above mentioned methods, Range-Kutta method give the nearest possible accurate answer.
Formulas are:

Euler Method

$$y = y + m * h$$

where, m = f(x,y)
h = Step Size (generally 0.1)

Heun's Method

$$y = y + (\frac{m1 + m2}{2}) * h$$

where, m1 = f(x,y)
m2 = f(x+h, y+m1*h)
h = Step Size

Range-Kutta method

$$y = y + \frac{m1 + (2 * m2) + (2 * m3) + m4}{6} * h$$

where, m1 = f(x,y)
m2 = f(x+h/2,y+m1*h/2)
m3 = f(x+h/2,y+m2*h/2)
m4 = f(x+h,y+m3*h)
h = Step Size

Program coding

```
#include <stdio.h>
#define function1(x,y) 2*y/x
void euler(float , float , float);
void hunes(float , float , float);
void rk4(float , float , float);
FILE *fp;
int main()
{
      float x, y, xp;

      printf("Enter value of x,y,xp\n");
      scanf("%f%f%f",&x,&y,&xp);

      fp = fopen("methods.xls","w+");
      euler(x,y,xp);
      hunes(x,y,xp);
      rk4(x,y,xp);
      fclose(fp);
      return 0;
}
void euler(float x, float y , float xp)
{
      printf("EULER'S METHOD\n");
      fprintf(fp,"EULERs\nx\ty\texact\n");
      float stepSize=0.1;
      float exact;
      int n;
      n = (xp-x)/stepSize;
      int i;
      float m;
      for(i=0;i<=n;i++)
      {
            exact = 2*x*x;
            printf("x = %f, y = %f, exact = %f\n",x,y,exact);
            fprintf(fp,"%f\t%f\t%f\n",x,y,exact);
            m = function1(x,y);
            y = y+ m*stepSize;
            x = x+ stepSize;
      }

printf("\n----------------------------------------------------------\n");
}
void hunes(float x, float y, float xp)
{
      printf("HUNE's METHOD\n");
      fprintf(fp,"HUNESs\nx\ty\texact\n");
      float stepSize = 0.1;
      float exact;
```

```
	int n,i;
	n = (xp-x)/stepSize;
	float m1, m2;
	for(i=0;i<=n;i++)
	{
		exact = 2*x*x;
		printf("x = %f, y = %f, exact = %f\n",x,y,exact);

		fprintf(fp,"%f\t%f\t%f\n",x,y,exact);
		m1 = function1(x,y);
		m2 = function1((x+stepSize),(y+m1*stepSize));
		y = y+ ((m1+m2)/2)*stepSize;
		x = x+ stepSize;
	}

printf("\n----------------------------------------------------------\n");
}
void rk4(float x, float y, float xp)
{
	printf("RK-4 METHOD\n");
	fprintf(fp,"RK-4s\nx\ty\texact\n");
	float stepSize = 0.1;
	float exact;
	int n,i;
	n = (xp-x)/stepSize;
	float m1, m2, m3, m4;
	for(i=0;i<=n;i++)
	{
		exact = 2*x*x;
		printf("x = %f, y = %f, exact = %f\n",x,y,exact);

		fprintf(fp,"%f\t%f\t%f\n",x,y,exact);
		m1 = function1(x,y);
		m2 = function1((x+(stepSize/2)),(y+m1*(stepSize/2)));
		m3 = function1((x+(stepSize/2)),(y+m2*(stepSize/2)));
		m4 = function1((x+stepSize),(y+m3*stepSize));
		y = y+ ((m1+(2*m2)+(2*m3)+m4)/6)*stepSize;
		x = x+ stepSize;
	}

printf("\n----------------------------------------------------------\n");
}
```

Program Output

x	exact	EULER	HUNES	RK-4
1	2	2	2	2
1.1	2.42	2.4	2.418182	2.419996
1.2	2.88	2.836364	2.876171	2.879992
1.3	3.38	3.309091	3.37397	3.379988
1.4	3.920001	3.818182	3.91158	3.919984
1.5	4.500001	4.363636	4.489004	4.49998
1.6	5.120001	4.945455	5.106242	5.119976
1.7	5.780001	5.563636	5.763295	5.779972
1.8	6.480001	6.218182	6.460164	6.479968
1.9	7.220002	6.909091	7.196849	7.219963
2	8.000002	7.636363	7.973351	7.999958

Fig: Program Output

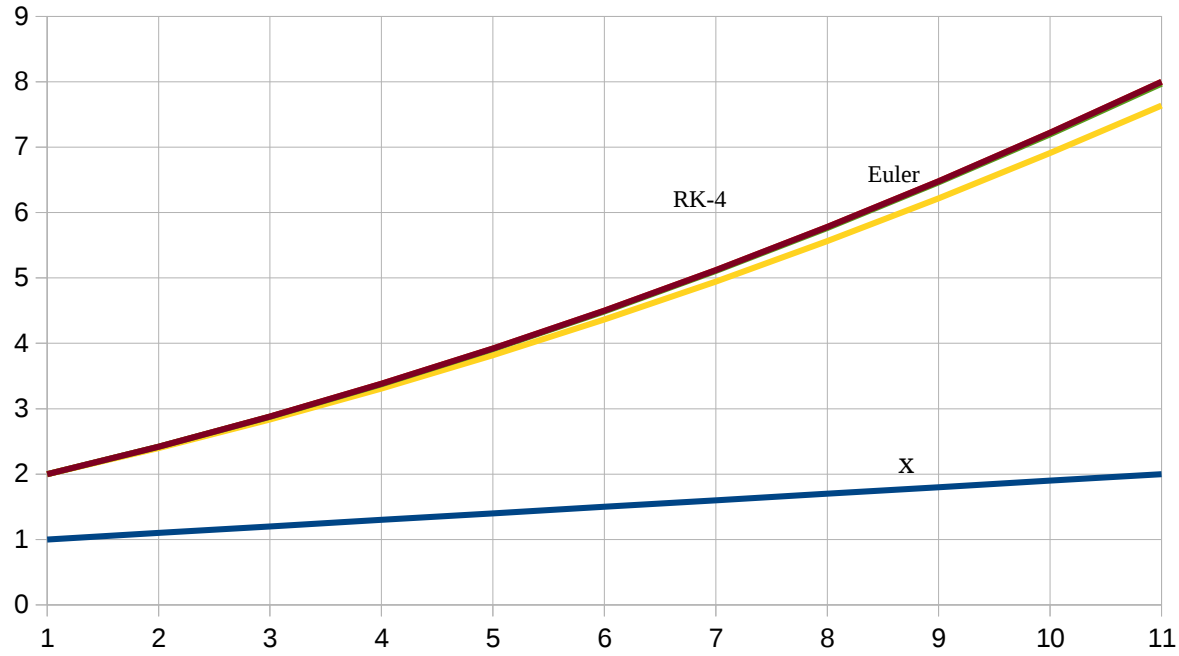


Illustration 1: Graphical representation of outputs

Conclusion:

After matching results from the exact solution to the all methods, it is found that the RK-4 methods gives the closest answer by reducing the error to minimum. Hence, RK-4 method for solving ordinary differential equation is more accurate than the Euler and Heune's method.