Warsaw University of Technology
Faculty of Electronics and Information Technology
Course : ENUME-2020

#11 Assignment A: Analysis of Accuracy of Computer Computation

Author: Keshav Dandeva [302333]
Advisor: Paweł Mazurek

Warsaw, 14^{th} April 2020

Contents

1. Introduction

This assignment examines and analyses the accuracy of computation done by a computer system with the aid of MATLAB software and implementation and systematic testing of numerical algorithms (NAs).Numerical algorithm can be defined as ordered sequence of operations transforming vectors of data to desired result vectors while fulfilling the requirements. Numerical analysis is the study of algorithms that use numerical approximation (as opposed to symbolic manipulations) for the problems of mathematical analysis (as distinguished from discrete mathematics). The assignment was to determine the functions characterising the propagation of the relative errors and the functions characterising the propagation of the relative errors caused by rounding the intermediate results of computing a given function. The results were obtained by two different means : 1.Analytical Differentiation and 2.Epsilon Calculus . The rest of the assignment was to assess the total error of computing the value of given function by inputting all possible combinations of error value as positive or negative epsilon ($eps = 10^{-12}$) by simulation in MATLAB and comparing it with the maximum value of the error computed by maximising the function and calculating the error.

2. Task 1

In this task, the functions characterising the propagation of the relative errors corrupting the data - $T_x(x,y), T_y(x,y)$ and the functions characterising the propagation of the relative errors caused by rounding the intermediate results of computing - $K_1(x,y), K_2(x,y)$... are calculated using means of two different methods for the following function:

$$z \equiv \frac{x^3 + \dfrac{cos(y)}{3}}{y - \dfrac{sin(y)}{2}} for(x,y) \in D \equiv \{x \in [1.10], y \in [1,10]\}$$

2.1 Calculations

2.1.1 Using Epsilon Calculus

To calculate the error coefficients using this method, we will break the equation in two parts: the numerator-(i) and the denominator-(ii) and calculate the relative errors by corrupting the data and then rounding the result for each part separately. In the end, both the equations will be combined and the error coefficients for the whole equation will be found.

$(i) x^3 + \dfrac{cos(y)}{3}$

$=> \left[x^3(1+\varepsilon_x)^3(1+\eta_{pow}) + \dfrac{1}{3}cos(y(1+\varepsilon_y))(1+\eta_{cos})(1+\eta_3)\right](1+\eta_{sum})$

$=> \left[x^3(1+3\varepsilon_x+\eta_{pow}) + \dfrac{1}{3}cos(y(1+\varepsilon_y))(1+\eta_{cos}+\eta_3)\right](1+\eta_{sum})$

Transforming $cos(y(1+\varepsilon_y))$ into $cos(y)(1+T_{cos}\varepsilon_y)$

$=> T_{cos}(y) = \dfrac{y}{cos(y)}\dfrac{d(cos(y))}{dy}$

$=> T_{cos}(y) = -tan(y)$

Substituting this in the equation, we get:

$=> \left[x^3(1+3\varepsilon_x+\eta_{pow}) + (\dfrac{1}{3}cos(y) - \dfrac{1}{3}y \cdot tan(y) \cdot cos(y) \cdot \varepsilon_y)(1+\eta_{cos}+\eta_3)\right](1+\eta_{sum})$

$=> \left[x^3 + x^3 \cdot 3\varepsilon_x + x^3 \cdot \eta_{pow} + \dfrac{1}{3}cos(y)(1 - y \cdot tan(y) \cdot \varepsilon_y \cdot +\eta_{cos} + \eta_3)\right](1+\eta_{sum})$

$=> \left[x^3 + x^3 \cdot 3\varepsilon_x + x^3 \cdot \eta_{pow} + \dfrac{1}{3}cos(y) - \dfrac{1}{3}y \cdot sin(y) \cdot \varepsilon_y + \dfrac{1}{3}cos(y) \cdot \eta_{cos} + \dfrac{1}{3}cos(y) \cdot \eta_3\right](1+\eta_{sum})$

$=> \left[x^3 + \dfrac{cos(y)}{3}\right](1 + \dfrac{x^3 \cdot 3\varepsilon_x + x^3 \cdot \eta_{pow} - \frac{1}{3}y \cdot sin(y) \cdot \varepsilon_y + \frac{1}{3}cos(y) \cdot \eta_{cos} + \frac{1}{3}cos(y) \cdot \eta_3}{x^3 + \frac{cos(y)}{3}} + \eta_{sum})$

$(ii) y - \dfrac{sin(y)}{2}$

$$=> \left[y(1+\varepsilon_y) - \frac{1}{2} sin(y(1+\varepsilon_y))(1+\eta_{sin})(1+\eta_2) \right] (1+\eta_{sub})$$

Transforming $sin(y(1+\varepsilon_y))$ into $sin(y)(1+T_{sin}\varepsilon_y)$

$$=> T_{sin}(y) = \frac{y}{sin(y)} \frac{d(sin(y))}{dy}$$

$$=> T_{sin}(y) = y \cdot cot(y)$$

Substituting this in the equation, we get:

$$=> \left[y + y \cdot \varepsilon_y - (\frac{1}{2} sin(y) + \frac{1}{2} sin(y) \cdot y \cdot cot(y) \cdot \varepsilon_y)(1+\eta_{sin}+\eta_2) \right] (1+\eta_{sub})$$

$$=> \left[y + y \cdot \varepsilon_y - \frac{1}{2} sin(y)(1 + y \cdot cot(y) \cdot \varepsilon_y + \eta_{sin} + \eta_2) \right] (1+\eta_{sub})$$

$$=> \left[y + y \cdot \varepsilon_y - \frac{1}{2} sin(y) - \frac{1}{2} y \cdot cos(y) \cdot \varepsilon_y - \frac{1}{2} sin(y) \cdot \eta_{sin} - \frac{1}{2} sin(y) \cdot \eta_2 \right] (1+\eta_{sub})$$

$$=> \left[y - \frac{sin(y)}{2} \right] (1 + \frac{y \cdot \varepsilon_y - \frac{1}{2} y \cdot cos(y) \cdot \varepsilon_y - \frac{1}{2} sin(y) \cdot \eta_{sin} - \frac{1}{2} sin(y) \cdot \eta_2}{y - \frac{sin(y)}{2}} + \eta_{sub})$$

Now, substituting the simplified equations (i) and (ii) in z:

$$\widetilde{z} \equiv \frac{(i)}{(ii)} \cdot (1+\eta_{div})$$

$$=> \widetilde{z} \equiv z(1 + \left(\frac{3x^3}{x^3 + \frac{cos(y)}{3}} \right) \varepsilon_x + \left(-\frac{y \cdot sin(y)}{3x^3 + cos(y)} - \frac{y}{y - \frac{sin(y)}{2}} + \frac{\frac{y}{2} \cdot cos(y)}{y - \frac{sin(y)}{2}} \right) \varepsilon_y + \left(\frac{x^3}{x^3 + \frac{cos(y)}{3}} \right) \eta_{pow} + \left(\frac{\frac{cos(y)}{3}}{x^3 + \frac{cos(y)}{3}} \right) \eta_{cos} + \left(\frac{\frac{cos(y)}{3}}{x^3 + \frac{cos(y)}{3}} \right) \eta_3 + \eta_{sum} - \eta_{sub} + \eta_{div} + \left(\frac{sin(y)}{2y - sin(y)} \right) \eta_{sin} + \left(\frac{sin(y)}{2y - sin(y)} \right) \eta_2)$$

This is the final equation determined by using epsilon calculus method.

Results:

After analysing the equation and comparing it to the general form, the coefficients of the errors are as follows:

1. $T_x = \left(\dfrac{3x^3}{x^3 + \frac{cos(y)}{3}} \right)$
2. $T_y = \left(-\dfrac{y \cdot sin(y)}{3x^3 + cos(y)} - \dfrac{y}{y - \frac{sin(y)}{2}} + \dfrac{\frac{y}{2} \cdot cos(y)}{y - \frac{sin(y)}{2}} \right)$
3. $K_{pow} = \left(\dfrac{x^3}{x^3 + \frac{cos(y)}{3}} \right)$
4. $K_{cos} = \left(\dfrac{\frac{cos(y)}{3}}{x^3 + \frac{cos(y)}{3}} \right)$
5. $K_3 = \left(\dfrac{\frac{cos(y)}{3}}{x^3 + \frac{cos(y)}{3}} \right)$
6. $K_{sum} = 1$

7. $K_{sub} = -1$

8. $K_{sin} = \left(\dfrac{sin(y)}{2y - sin(y)}\right)$

9. $K_2 = \left(\dfrac{sin(y)}{2y - sin(y)}\right)$

10. $K_{div} = 1$

2.1.2 Using Analytical Differentiation

In this method, we substitute the term for which the error has to be analysed with a variable and use the following formula to find the coefficient.

$$T_f(z) = \frac{z}{f(z)} \cdot \frac{df(z)}{dz}$$

The values of all the coefficients of the errors in the equation are calculated using MATLAB and the code is presented in the Appendix for reference.

Result:
From this method, coefficient functions were in different form than calculated from the epsilon calculus method but their values and graph were all identical.

2.2 Graphical Representation

Now, the graphs for all the coefficients of the errors in the equation will be plotted except for the coefficient with value 1 or -1. The graphs from both functions are identical so only one graph for both methods is being presented.

The function used to plot the graph is $fsurf(x, y, z)$ that creates a 3-Dimensional graph and the colorful presentation helps to view the increase and decrease in slope of the graph.

Figure 2.1: T$_x$

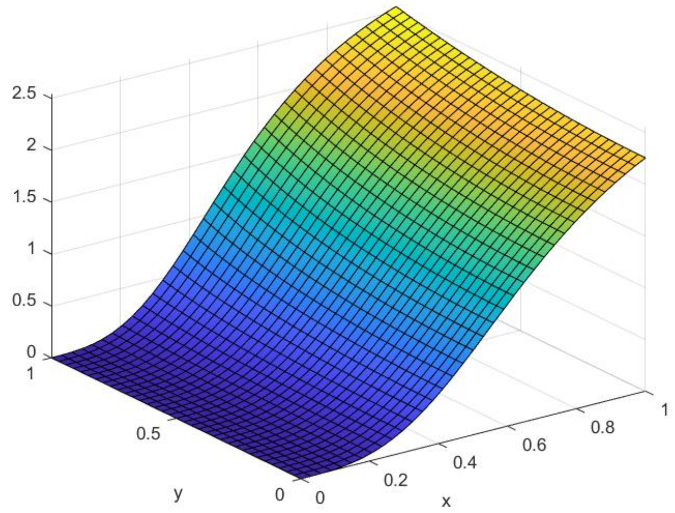


Figure 2.2: T$_y$

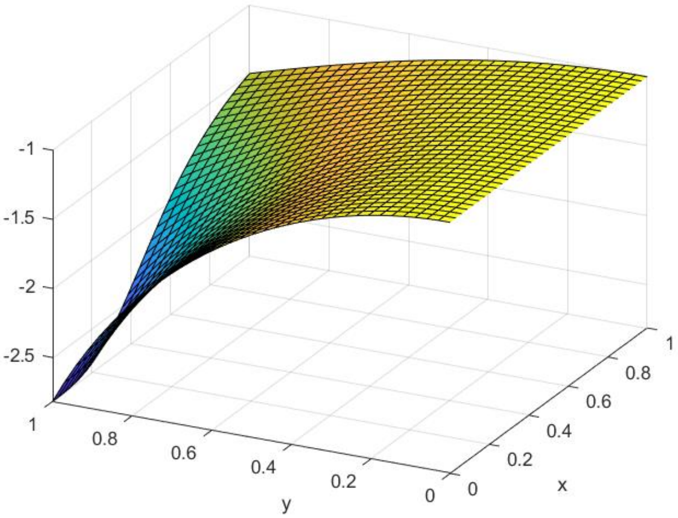


Figure 2.3: K$_{pow}$

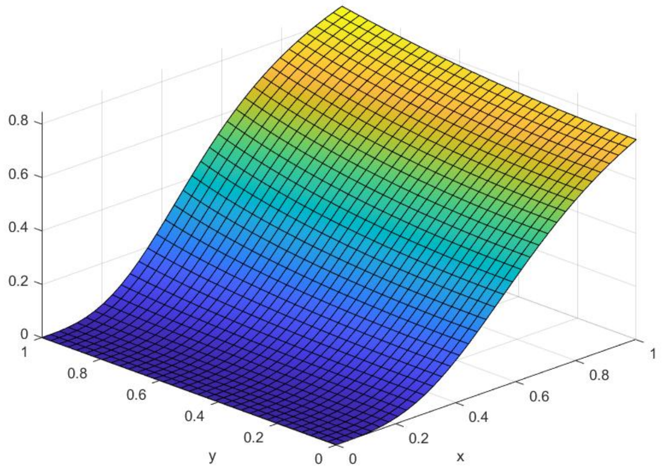


Figure 2.4: K$_{cos}$

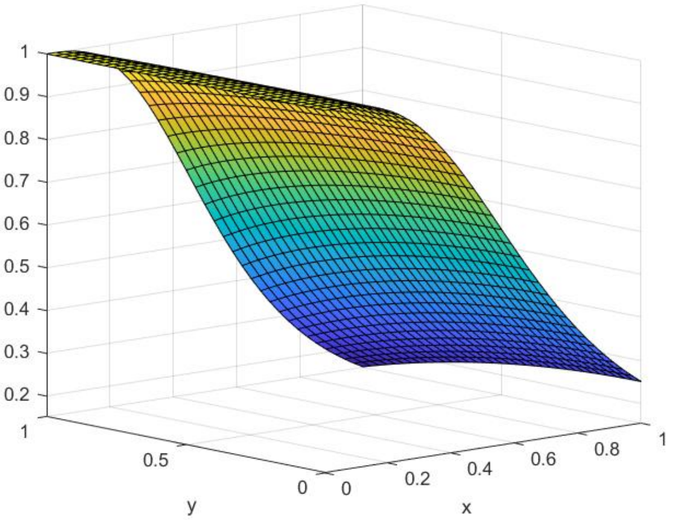


Figure 2.5: K$_3$

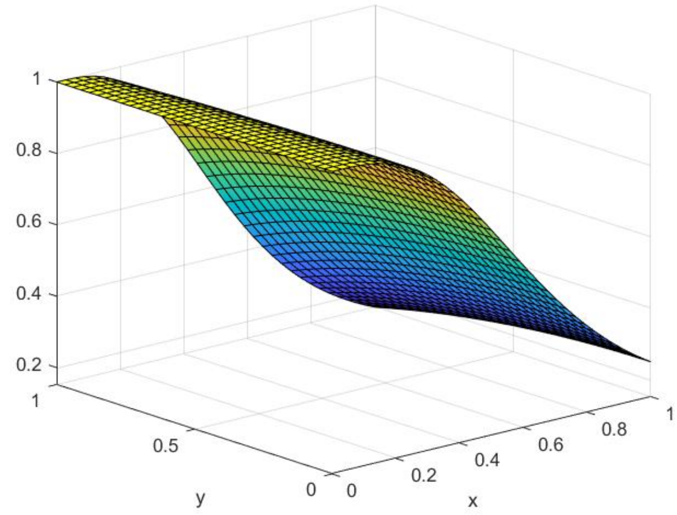


Figure 2.6: K$_{sin}$

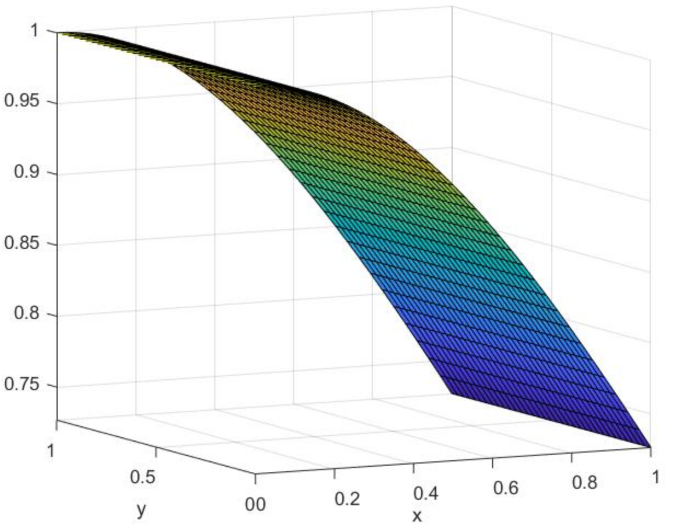
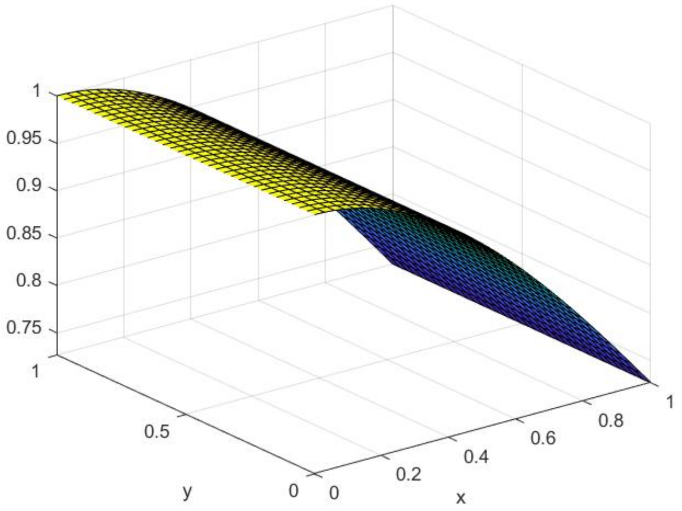


Figure 2.7: K_2



3. Task 2

In this task, we have to evaluate the total error of computing the value of the given equation $z(x, y)$ by maximising the following function:

$$\delta z_{sup}^{(1)} = sup\,\{|T_x(x, y))| + |T_y(x, y))| + |K_1(x, y))| + |K_2(x, y))| + ... \mid (x, y) \in D\} * eps$$

3.1 Calculations

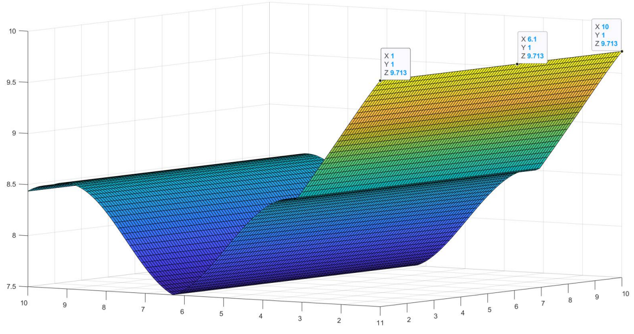
First, the absolute values of the coefficients of errors obtained from Task 1 are calculated and then added to form a function. Then, assuming that the indicator of the accuracy of the floating-point representation is $eps = 10^{-12}$, eps is multiplied to the supremum of the function to obtain the final value of the total error of the equation $z(x, y)$.

This task is performed in MATLAB with the aid of *meshgrid* function to visualise the function in 3-Dimension and max function to calculate the maximum value of the function. The code to perform this task is attached in Appendix for reference.

3.2 Graphical Representation

The graph for the function obtained by adding the absolute values of the coefficient of error is plotted with the help of the function $meshgrid(x, y)$. It provides a 3-Dimensional view of the function with the maximum value of z.

Figure 3.1: 3-D Graph



3.3 Results

The maximum value of the function obtained with the help of *max* function in MATLAB and analysis of the graph is 9.7126. The value multiplied by the given $eps = 10^{-12}$ results in the total error of the equation, that is :

$$\delta z_{sup}^{(1)} = 9.7126 \cdot 10^{-12}$$

4. Task 3

In this task, the maximum value of the relative error of the function z is to be calculated using MATLAB simulation method.

$$\delta z^{(2)}_{sup} = sup\{|\delta z(x,y))| \mid (x,y) \in D\}$$

4.1 Calculations

In this task, we have to calculate all the possible values of relative error of z by substituting all possible combinations of +eps and -eps with the error terms. All the calculations for this task are done in MATLAB. The MATLAB code for this task is attached in appendix for reference.

There are 2^{10} possible combinations for substituting +eps and -eps because there are 10 different types of error in the equation.Therefore, the matrix *d* is created with the help of *de2bi* function which will hold all the possible combinations of eps substitution.

The major part of the task is done inside 2 nested for loops iterating for substituting values in different functions. The inner loop is used to create a new vector from the d vector and all 10 values of eps are copied into this new vector. Then the rest of the calculations are done in outer loop.

In the outer loop, values from the new vector are substituted in the function z. Then using *meshgrid* function and *max* function, the values obtained of the actual z function are stored in a vector m.

Now, the expected z function is calculated the sameway but without error values. Then, the relative errors are stored in a vector using the following formula for relative error:

$$\delta_z(x,y) = \left| \frac{z\_actual - z\_expected}{z\_expected} \right|$$

.

Finally, the maximum value from the relative_error vector is found using the simple *max* function.

4.2 Results

The maximum value of the relative error obtained by the means of simulation method in MATLAB is:

$$\delta z^{(2)}_{sup} = 9.7135 \cdot 10^{-12}$$

The difference between the values of error found in Task 2 and Task 3 is :

$$\delta z^{(2)}_{sup} - \delta z^{(1)}_{sup} = 9.7135 \cdot 10^{-12} - 9.7126 \cdot 10^{-12} = 0.0009 \cdot 10^{-12}$$

5. Conclusions

5.1 Task 1

In the first task, the values of the the functions characterising the propagation of the relative errors corrupting the data and the functions characterising the propagation of the relative errors caused by rounding the intermediate results of computing, obtained using analytical differentiation and using epsilon calculus are identical and hence resulting in successful calculation and graph plot. This shows that values calculated using both methods were right.

5.2 Task 2

In the second task, the value of the relative error of the function z is equal to $9.7126 \cdot 10^{-12}$. The relative error is based on the assumption that the indicator of the accuracy of the floating-point representation is $eps = 10^{-12}$.

5.3 Task 3

In the third task, the value of the relative error of the function z found with the help of simulation in MATLAB is equal to $9.7135 \cdot 10^{-12}$. On comparing this value with the value obtained in task 2, it is found that the magnitude of difference between the both values is of $0.0009 \cdot 10^{-12}$ which is subsequently very less but still lacks precision. MATLAB provides precision till the 4th digit decimal place.

6. Appendix

6.1 Task 1

```
clear
close all
clc

syms x y z v

z=(x^3 + (cos(y))/3)/(y-(sin(y))/2)

Tx=x/z*diff(z,x)

figure(1)
fsurf(Tx,[0,1,0,1])
xlabel('x')
ylabel('y')

Tx2= (3*x^3)/(x^3 + (cos(y))/3)

figure(2)
fsurf(Tx2,[0,1,0,1])
xlabel('x')
ylabel('y')


Ty=y/z*diff(z,y)

figure(3)
fsurf(Ty,[0,1,0,1])
xlabel('x')
ylabel('y')

Ty2=((-y/3)*sin(y))/(x^3+(cos(y))/3) - y/(y-0.5*sin(y)) + (0.5*y*cos(y))/(y-0.5*sin(y))

figure(4)
fsurf(Ty2,[0,1,0,1])
xlabel('x')
ylabel('y')


zs=subs(z,x^3,v)
Kpow=v/zs*diff(zs,v)
Kpow=subs(Kpow,v,x^3)
Kpow2=(x^3)/(x^3 + (cos(y))/3)

figure(5)
fsurf(Kpow,[0,1,0,1])
xlabel('x')
```

```
ylabel('y')

figure(6)
fsurf(Kpow2,[0,1,0,1])
xlabel('x')
ylabel('y')


zs=subs(z,cos(y),v)
Kcos=v/zs*diff(zs,v)
Kcos=subs(Kcos,v,cos(y))
Kcos2=((cos(y))/3)/(x^3 + (cos(y))/3)

figure(7)
fsurf(Kcos,[0,1,0,1])
xlabel('x')
ylabel('y')

figure(8)
fsurf(Kcos2,[0,1,0,1])
xlabel('x')
ylabel('y')


zs=subs(z,(1/3),v)
K3=v/zs*diff(zs,v)
K3=subs(K3,v,(1/3))
K3_2=((cos(y))/3)/(x^3 + (cos(y))/3)

figure(9)
fsurf(K3,[0,1,0,1])
xlabel('x')
ylabel('y')

figure(10)
fsurf(K3_2,[0,1,0,1])
xlabel('x')
ylabel('y')


zs=subs(z,(x^3 + (cos(y))/3),v)
Ksum=v/zs*diff(zs,v)
Ksum=subs(Ksum,v,(x^3 + (cos(y))/3))
Ksum2= 1

zs=subs(z,(y-(sin(y))/2),v)
Ksub=v/zs*diff(zs,v)
Ksub=subs(Ksub,v,(y-(sin(y))/2))
Ksub2= -1


zs=subs(z,(x^3 + (cos(y))/3)/(y-(sin(y))/2),v)
Kdiv=v/zs*diff(zs,v)
Kdiv=subs(Kdiv,v,(x^3 + (cos(y))/3)/(y-(sin(y))/2))
Kdiv2= 1

zs=subs(z,sin(y),v)
Ksin=v/zs*diff(zs,v)
Ksin=subs(Ksin,v,sin(y))
Ksin2=((sin(y))/2)/(y-(sin(y))/2)

figure(11)
```

```
fsurf(Ksin,[0,1,0,1])
xlabel('x')
ylabel('y')

figure(12)
fsurf(Ksin2,[0,1,0,1])
xlabel('x')
ylabel('y')


zs=subs(z,sin(y)/2,v*sin(y))
K2=v/zs*diff(zs,v)
K2=subs(K2,v,1/2)
K2_2=((sin(y))/2)/(y - (sin(y))/2)

figure(13)
fsurf(K2,[0,1,0,1])
xlabel('x')
ylabel('y')

figure(14)
fsurf(K2_2,[0,1,0,1])
xlabel('x')
ylabel('y')
```

6.2 Task 2

```
clear
close all
clc

syms x y z

z =@(x,y) (abs((3.*x^3)./(x^3 + (cos(y))./3)) + abs(((-y./3).*sin(y))./(x^3+(cos(y))./3) - ...
    y./(y-(0.5).*sin(y)) + ((0.5).*y.*cos(y))./(y-(0.5).*sin(y))) + abs((x^3)./(x^3 + ...
    (cos(y))./3)) + abs(((cos(y))./3)./(x^3 + (cos(y))./3)) + abs(((cos(y))./3)./(x^3 + ...
    (cos(y))./3)) + 3 + abs(((sin(y))./2)./(y-(sin(y))./2)) + abs(((sin(y))./2)./(y - ...
    (sin(y))./2)));

x = linspace(1,10,1000);
y = linspace(1,10,1000);

[X,Y] = meshgrid(x,y);
sh = surf(X, Y, z(X,Y));

zd=get(sh,'zdata');
zmax=max(max(zd))
```

6.3 Task 3

```
clear
close all
clc


d = de2bi(0:1023);
d(d==0) = -1;
d = 10^(-12)*d;

vec = zeros(1,10);

 for i = 1:1024
     for j = 1:10
         vec(j) = d(i,j);
     end

     z =@(x,y) abs(((x^3.*(1+ vec(1))^3.*(1+vec(2)) + (1./3).*cos(y.*(1+vec(3))).*(1+ ...
         vec(4)).*(1+ vec(5))).*(1+ vec(6)).*(1 + vec(7)))./((y.*(1+ vec(3)) - ...
         (1./2).*sin(y.*(1+ vec(3))).*(1+vec(8)).*(1+vec(9))).*(1+vec(10))));

     x = linspace(1,10,500);
     y = linspace(1,10,500);

     [X,Y] = meshgrid(x,y);
     sh = surf(X, Y, z(X,Y));

     zd=get(sh,'zdata');
     zmax=max(max(zd));
     m(i) = zmax;
     zs =@(x,y) (x^3 + (cos(y))./3)./(y-(sin(y))./2);
     x = linspace(1,10,500);
     y = linspace(1,10,500);

     [X,Y] = meshgrid(x,y);
     sh = surf(X, Y, zs(X,Y));

     zd=get(sh,'zdata');
     zmax=max(max(zd));
     expected_z = zmax;

     relative_error(i) = abs((m(i) - expected_z)./expected_z);
 end

max(relative_error)
```