Chmielewska Urszula

300167

Numerical Methods

Analysis of accuracy of computer computation

Assignment A nr. #8

Advisor: dr inż. Andrzej Miękina

Theoretical introduction

Propagation of uncertainty (propagation of error) is a consequence of uncertainties of the variables and functions, composed of themselves. During experimental measurements, values of the uncertainties depend, for example, on the instrument precision or on the observer's lack of precision. In case of computation uncertainties, variables values become less precise, because of rounding errors, generated during rounding the intermediate results of computing. One of the method of proper error approximation is epsilon calculus.

The approach in epsilon calculus states to represent each possible small error by different epsilons. It is important, especially for scientists, to compute and understand the magnitude of errors possible to obtain, before starting measurements.

The project consists of three tasks, all concerning to the accuracy of computer computations. Short introduction, specifying each problem, is stated at the beginning of every task.

TASK 1.

In the first task I was supposed to determine the functions characterizing the propagation of the relative errors corrupting the date and the functions characterizing the propagation of the relative errors caused by rounding the intermediate results of computing, for the following *z* function:

$$z \equiv \frac{x + \frac{\sin(y)}{3}}{y^3 + \ln(y)} \quad \text{for } (x, y) \epsilon \ D \ \equiv \ \{x \epsilon [1,10], y \epsilon [1,10]\}$$

Theoretical background

In each step of computation I used rules of epsilon algebra. Here I place all of the formulas which are necessary to know, to understand the way of solving problem stated below:

$$(1+\varepsilon_1)(1+\varepsilon_2) \cong 1+\ \varepsilon_1+\ \varepsilon_2$$

$$(1+\varepsilon)^{a} \cong 1+a\varepsilon$$

$$f(z(1+\ \delta)) \cong f(z)(1+T_f(z)\delta),\ where$$

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

Epsilon calculus computation

$$\check{Z} \cong \frac{\left[x(1+\varepsilon_x)+\frac{\sin\left(y(1+\varepsilon_y)\right)}{3}(1+\eta_{sin})(1+\eta_{div3})\right](1+\eta_{\mathrm{sum1}})(1+\eta_{div})}{\left[\left(y(1+\varepsilon_y)\right)^3(1+\ \eta_{pow})+\ln(y(1+\varepsilon_y))(1+\eta_{ln})\right](1+\ \eta_{\mathrm{sum2}})}$$

$$\cong \frac{\left[x(1+\varepsilon_x)+\frac{\sin(y)(1+y\cot(y)\varepsilon_y)}{3}(1+\eta_{sin}+\eta_{div3})\right](1+\eta_{\mathrm{sum1}}+\eta_{div})}{\left[(y^3\left(1+3\varepsilon_y\right)\left(1+\ \eta_{pow}\right)+\ln(y)(1+\frac{1}{\ln(y)}\varepsilon_y))(1+\eta_{ln})\right](1+\ \eta_{\mathrm{sum2}})}$$

$$\cong \frac{\left[x+x\varepsilon_x+\frac{\sin(y)(1+y\cot(y)\varepsilon_y+\eta_{sin}+\eta_{div3})}{3}\right](1+\eta_{\mathrm{sum1}}+\eta_{div})}{\left[y^3\left(1+3\varepsilon_y+\eta_{pow}\right)+\ln(y)(1+\frac{1}{\ln(y)}\varepsilon_y+\eta_{ln})\right](1+\ \eta_{\mathrm{sum2}})}$$

$$\cong \frac{\left[x+x\varepsilon_x+\frac{\sin(y)(1+y\cot(y)\varepsilon_y+\eta_{sin}+\eta_{div3})}{3}\right](1+\eta_{\mathrm{sum1}}+\eta_{div})}{\left[y^3+3y^3\varepsilon_y+y^3\eta_{pow}+\ln(y)(1+\frac{1}{\ln(y)}\varepsilon_y+\eta_{ln})\right]}$$

$$\cong \frac{\left[x+x\varepsilon_x+\frac{\sin(y)+y\sin(y)\cot(y)\varepsilon_y+\sin(y)\eta_{sin}+\sin(y)\eta_{div3}}{3}\right](1+\eta_{\mathrm{sum1}}+\eta_{div})}{\left[y^3+3y^3\varepsilon_y+y^3\eta_{pow}+\ln(y)+\varepsilon_y+\ln(y)\eta_{\mathrm{ln}}\right](1+\ \eta_{\mathrm{sum2}})}$$

$$\cong \frac{\left(x+\frac{\sin(y)}{3}\right)\left[1+\frac{x\varepsilon_x+\frac{1}{3}y\sin(y)\cot(y)\varepsilon_y+\frac{1}{3}\sin(y)\eta_{sin}+\frac{1}{3}\sin(y)\eta_{div3}}{\left(x+\frac{\sin(y)}{3}\right)}\right](1+\eta_{\mathrm{sum1}}+\eta_{div})}{(y^3+\ln(y))\left(1+\frac{3y^3\varepsilon_y+y^3\eta_{pow}+\varepsilon_y+\ln(y)\eta_{\mathrm{ln}}}{(y^3+\ln(y))}\right)(1+\ \eta_{\mathrm{sum2}})}$$

$$\cong \frac{\left(x+\frac{\sin(y)}{3}\right)}{(y^3+\ln(y))}\,\frac{\left(1+\frac{x\varepsilon_x+\frac{1}{3}\text{y}\sin(y)\cot(y)\varepsilon_y+\frac{1}{3}\sin(y)\eta_{sin}+\frac{1}{3}\sin(y)\eta_{div3}}{\left(x+\frac{\sin(y)}{3}\right)}+\eta_{\text{sum1}}+\eta_{div}\right)}{\left(1+\frac{3y^3\varepsilon_y+y^3\eta_{pow}+\varepsilon_y+\ln(y)\eta_{\ln}}{(y^3+\ln(y))}\right)(1+\ \eta_{\text{sum2}})}$$

$$\cong \frac{\left(x+\frac{\sin(y)}{3}\right)}{(y^3+\ln(y))}\left(1+\frac{x\varepsilon_x+\frac{1}{3}\text{y}\sin(y)\cot(y)\varepsilon_y+\frac{1}{3}\sin(y)\eta_{sin}+\frac{1}{3}\sin(y)\eta_{div3}}{\left(x+\frac{\sin(y)}{3}\right)}+\eta_{\text{sum1}}+\eta_{div}\right)\left(1+\right.$$
$$\left.\frac{3y^3\varepsilon_y+y^3\eta_{pow}+\varepsilon_y+\ln(y)\eta_{\ln}}{(y^3+\ln(y))}-\ \eta_{\text{sum2}}\right)^{-1}$$

$$\cong \frac{\left(x+\frac{\sin(y)}{3}\right)}{(y^3+\ln(y))}\left(1+\frac{x\varepsilon_x+\frac{1}{3}\text{y}\sin(y)\cot(y)\varepsilon_y+\frac{1}{3}\sin(y)\eta_{sin}+\frac{1}{3}\sin(y)\eta_{div3}}{\left(x+\frac{\sin(y)}{3}\right)}+\eta_{\text{sum1}}+\eta_{div}\right)\left(1-\right.$$
$$\left.\frac{3y^3\varepsilon_y+y^3\eta_{pow}+\varepsilon_y+\ln(y)\eta_{\ln}+\eta_{\text{sum2}}}{(y^3+\ln(y))}-\eta_{\text{sum2}}\right)$$

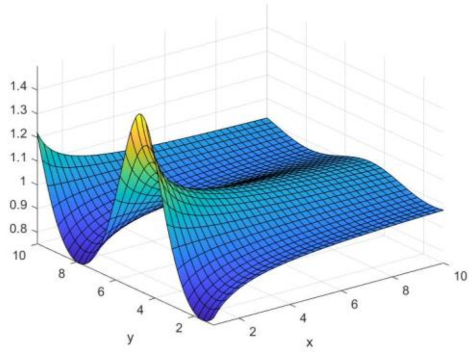
$$\cong \frac{\left(x+\frac{\sin(y)}{3}\right)}{(y^3+\ln(y))}\left(1+\frac{x\varepsilon_x+\frac{1}{3}\text{y}\sin(y)\cot(y)\varepsilon_y+\frac{1}{3}\sin(y)\eta_{sin}+\frac{1}{3}\sin(y)\eta_{div3}}{\left(x+\frac{\sin(y)}{3}\right)}+\eta_{\text{sum1}}+\eta_{div}-\right.$$
$$\left.\frac{3y^3\varepsilon_y+y^3\eta_{pow}+\varepsilon_y+\ln(y)\eta_{\ln}}{(y^3+\ln(y))}-\eta_{\text{sum2}}\right)$$

$$\check{Z}\cong z\big(1+T_x\varepsilon_x+T_y\varepsilon_y+K_{sin}\eta_{sin}+K_{div3}\eta_{div3}+K_{pow}\eta_{pow}+K_{div}\eta_{div}+K_{sum1}\eta_{sum1}+K_{sum2}\eta_{sum2}+K_{ln}\eta_{ln}\big)$$

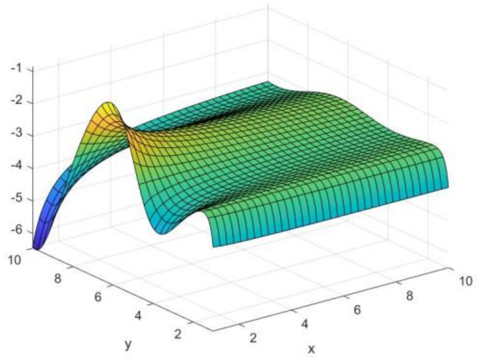
MatLab computation

Obtained results were checked also by calculations in MatLab. All of the computed error functions, were the same as the results of computation, made above epsilon calculus. Below, there is a list of all functions characterizing the propagation of the relative errors in *z* function, with proper graph representations:

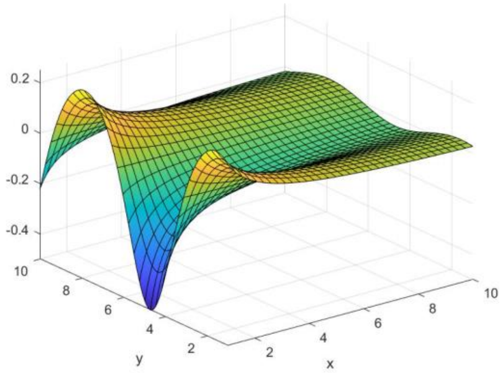
$$T_x = \frac{x}{\left(x + \frac{\sin(y)}{3}\right)}$$



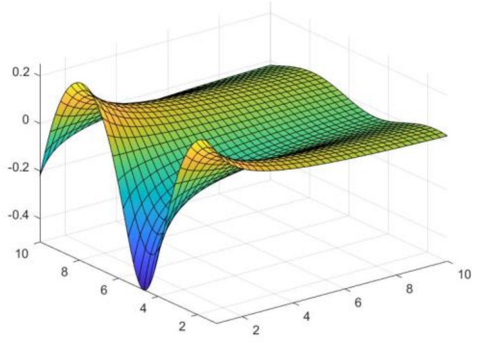
$$T_y = \frac{\frac{1}{3}y\cos(y)}{\left(x + \frac{\sin(y)}{3}\right)} - \frac{1+3y^3}{(y^3 + \ln(y))}$$



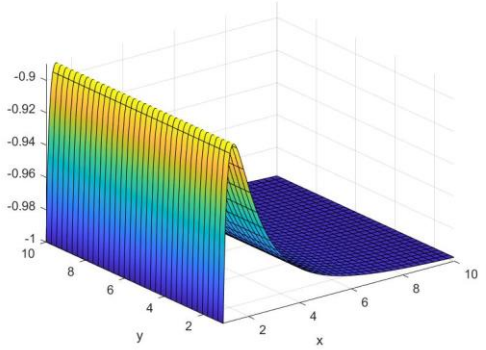
$$K_{sin} = \frac{\frac{1}{3}\sin(y)}{\left(x + \frac{\sin(y)}{3}\right)}$$



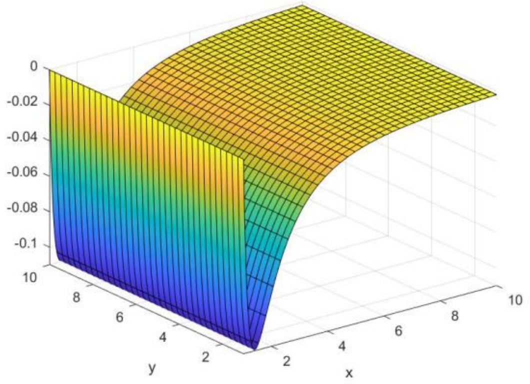
$$K_{div3} = \frac{\frac{1}{3}\sin(y)}{\left(x+\frac{\sin(y)}{3}\right)}$$



$$K_{pow} = -\frac{y^3}{(y^3 + \ln(y))}$$



$$K_{ln} = -\frac{\ln(y)}{(y^3 + \ln(y))}$$



$$K_{div} = 1$$

$$K_{sum1} = 1$$

$$K_{sum2} = -1$$

Analytical differentiation

Using this method it is possible to evaluate formulas for $T_x(x,y)$ and $T_y(x,y)$

1) $$T_x(x,y) = \frac{x}{f(x,y)} * \frac{df(x,y)}{dx} = x * \frac{y^3 + \ln(y)}{x + \frac{\sin(y)}{3}} * \frac{df(x,y)}{dx} = x * \frac{y^3 + \ln(y)}{x + \frac{\sin(y)}{3}} * \frac{1}{y^3 + \ln(y)}$$

$$T_x(x,y) = \frac{x}{x + \frac{\sin(y)}{3}}$$

2) $$T_y(x,y) = \frac{y}{f(x,y)} * \frac{df(x,y)}{dy} = y * \frac{y^3 + \ln(y)}{x + \frac{\sin(y)}{3}} * \frac{df(x,y)}{dy} =$$

$$= y * \frac{y^3 + \ln(y)}{x + \frac{\sin(y)}{3}} * \left(\frac{((y^3 + \ln(y))(\frac{\cos(y)}{3})) - (3y^2 + \frac{1}{y})(x + \frac{\sin(y)}{3})}{(y^3 + \ln(y))^2} \right) =$$

$$= y * \frac{y^3 + \ln(y)}{x + \frac{\sin(y)}{3}} \left(\frac{\frac{1}{3}\cos(y)}{(y^3 + \ln(y))} - \frac{\left(3y^2 + \frac{1}{y}\right)\left(x + \frac{\sin(y)}{3}\right)}{(y^3 + \ln(y))^2} \right) =$$

$$= \frac{\frac{1}{3} y cos(y)}{\left(x + \frac{\sin(y)}{3}\right)} - \frac{1+3y^3}{(y^3 + \ln(y))}$$

$$T_y(x,y) = \frac{\frac{1}{3} y \cos(y)}{\left(x + \frac{\sin(y)}{3}\right)} - \frac{1+3y^3}{(y^3 + \ln(y))}$$

CONCLUSIONS

Results of errors Tx and Ty by all of three methods were the same. In epsilon calculus and MatLab calculations of all other errors, obtained results were also exact. I haven’t noticed significant differences in the time required to solve in each approach. Only analytical approach was slightly faster than epsilon calculus, however it is not possible to use it for calculating relative errors caused by rounding the intermediate results of computing. Precision in all methods is the same, however it is a desirable approach, especially for in research and science, to compare results from various methods. This is the best way to affirm that the outcome is correct.

TASK 2.

Second task is related to the calculation of the total error of computing the value of z, by maximizing the indicator $\delta z^{(1)}_{sup}$. The floating-point in my case is *eps* = $3 \cdot 10^{-13}$

$$\delta z^{(1)}_{sup} = sup\{|T_x(x,y)| + |T_y(x,y)| + |K_{sin}(x,y)| + |K_{div3}(x,y)| + |K_{pow}(x,y)| + |K_{div}(x,y)| + |K_{sum1}(x,y)| + |K_{ln}(x,y)| + |K_{sum2}(x,y)|\ \} \epsilon \mathbb{D}\}{*}eps$$

Formula of my specified indicator has a form:

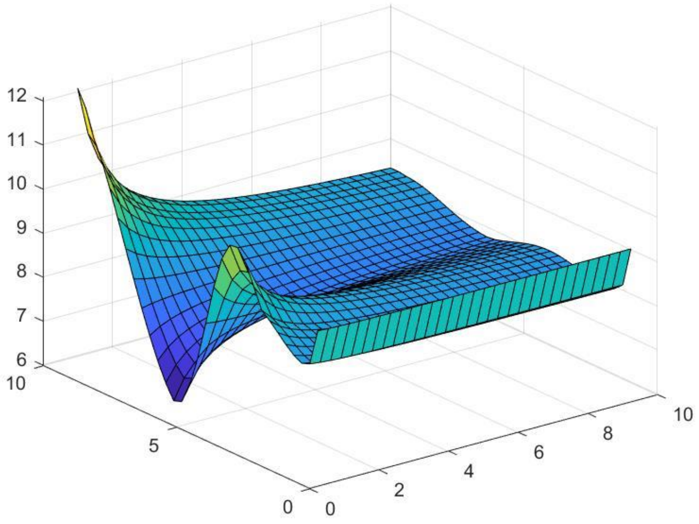
$$\delta z^{(1)}_{sup} = sup\left\{ \left| \frac{x}{\left(x + \frac{\sin(y)}{3}\right)} \right| + \left| \frac{\frac{1}{3} y \cos(y)}{\left(x + \frac{\sin(y)}{3}\right)} - \frac{1+3y^3}{(y^3 + \ln(y))} \right| + \left| \frac{\frac{1}{3}\sin(y)}{\left(x + \frac{\sin(y)}{3}\right)} \right| + \left| \frac{\frac{1}{3}\sin(y)}{\left(x + \frac{\sin(y)}{3}\right)} \right| + \left| -\frac{y^3}{(y^3 + \ln(y))} \right| + |1| + |1| + \left| -\frac{\ln(y)}{(y^3 + \ln(y))} \right| + |-1| \right\} \epsilon \mathbb{D}\}{*}eps$$

To solve the problem, in the first step, I created a function which consists of the sum of the absolute values of errors, calculated in the first task.

task2=abs(T_x)+ abs(T_y)+ abs(K_{sin})+…

Obtained function was a symbolic function. Using *matlabFunction(task2)*, I transformed it to the numerical one. The main purpose of this solution was to find the least precise matrix for which the obtained maximum value of z is the same as for more precise ones. In this way obtained solution became more efficient. The minimum division, of the given interval [1;10], was 0.30.

Graph represents the three-dimensional surface plot with the values contained in matrix



By the *max()* function I was able to find the maximum *z* value multiplied by given *eps*, which was equal to:

$$\delta z^{(1)}_{sup} = 3.6225 \cdot 10^{-12}$$

TASK 3.

The aim of the third task was to obtain total error by means of the simulation method.

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

In z function calculation, there are 9 possible errors to get, which gives $2^9 = 512$ possible combinations of *-eps* and *+eps*. Instead of manual calculation of each option, I generated matrix with *512* elements by binary representation of decimal numbers from 0 to 511. Substituting 0’s with -1’s gave me matrix with 9 columns with all possible to obtain combinations of -1 and 1. After all I took this first z function formula with errors:

$$\frac{\left[x(1+\varepsilon_x)+\frac{\sin\left(y(1+\varepsilon_y)\right)}{3}(1+\eta_{sin})(1+\eta_{div3})\right](1+\eta_{\mathrm{sum1}})(1+\eta_{div})}{\left[\left(y(1+\varepsilon_y)\right)^3(1+\eta_{pow})+\ln(y(1+\varepsilon_y))(1+\eta_{ln})\right](1+\eta_{\mathrm{sum2}})}$$

- and substituted each error with +*eps* and -*eps*.

Using *for* loops I calculated all possible values of errors in range for *x*,*y [1;10]*. I took precision equal to *0.30*, as in the *Task 2*. The maximum value obtained in *512* iterations was my solution, equal to:

$$\delta z_{sup}^{(2)} = 3.6224 \cdot 10^{-12}$$

FINAL CONCLUSIONS

In results obtained in *Task 2* and *Task 3* there is visible a slight difference. Obtained values are not exactly the same, however the difference is significant enough to conclude, that both error calculation methods are likewise precise.

REFERENCES

- ENUME Lecture notes, spring semester 2020

- youtube.com (ENUME 2020 Assignment A: Introduction)

- wikipedia.org

APPENDIX – FULL MATLAB SOURCE CODE

```
close all
clear
clc

%===========================TASK 1===========================

syms x y z v
z=(x+sin(y)/3)/(y^3+log(y));

Tx=x/z*diff(z,x)
fsurf(Tx, [1,10,1,10]);
Tx2=x/(x+sin(y)/3);
fsurf(Tx2, [1,10,1,10]);

Ty=y/z*diff(z,y)
Ty2=(y*cos(y))/(3*(x+sin(y)/3))-(1+3*y^3)/(y^3+log(y));
fsurf(Ty, [1,10,1,10]);
fsurf(Ty2, [1,10,1,10]);

zs=subs(z,sin(y),v);
Ksin=v/zs*diff(zs,v);
Ksin=subs(Ksin,v,sin(y))
fsurf(Ksin, [1,10,1,10]);
Ksin2=(sin(y)/3)/(x+sin(y)/3);
fsurf(Ksin2, [1,10,1,10]);

zd=subs(z,sin(y)/3,v);
Kdiv3=v/zd*diff(zd,v);
Kdiv3=subs(Kdiv3,v,sin(y)/3)
fsurf(Kdiv3, [1,10,1,10]);
Kdiv32=(sin(y)/3)/(x+sin(y)/3);
fsurf(Kdiv32, [1,10,1,10]);

zp=subs(z,y^3,v);
Kpow=v/zp*diff(zp,v);
Kpow=subs(Kpow,v,y^3)
fsurf(Kpow, [1,10,1,10]);
Kpow2=-y^3/(y^3+log(y));
fsurf(Kpow2, [1,10,1,10]);

zdiv=subs(z,(x+sin(y)/3)/(y^3+log(y)),v);
Kdiv=v/zdiv*diff(zdiv,v);
Kdiv=subs(Kdiv,v,(x+sin(y)/3)/(y^3+log(y)))
fsurf(Kdiv, [1,10,1,10]);

zs1=subs(z,(x+sin(y)/3),v);
Ksum1=v/zs1*diff(zs1,v);
Ksum1=subs(Ksum1,v,(x+sin(y)/3))
fsurf(Ksum1,[1,10,1,10]);

zs2=subs(z,(y^3+log(y)),v);
Ksum2=v/zs2*diff(zs2,v);
Ksum2=subs(Ksum2,v,(y^3+log(y)))
fsurf(Ksum2,[1,10,1,10]);

zl=log(y);
zl=subs(z,log(y),v);
Kln=v/zl*diff(zl,v);
Kln=subs(Kln,v,log(y))
```

```
fsurf(Kln,[1,10,1,10]);

%============================TASK 2============================

task2=abs(Tx)+abs(Ty)+abs(Kdiv)+abs(Kdiv3)+abs(Kln)+abs(Kpow)+abs(Ksin)+abs
(Ksum1)+abs(Ksum2);
eps=3.*10.^(-13);

task2num=matlabFunction(task2);
[X,Y]=meshgrid(1:0.30:10,1:0.30:10);
task2Matrix=task2num(X,Y);

surf(X,Y,task2Matrix);
disp("Maximum value of z for matrix precise to 0.30")
zmax=max(max(task2Matrix))

%========check for a more precise matrix====================
[X,Y]=meshgrid(1:0.20:10,1:0.20:10);
task2Matrix2=task2num(X,Y);
disp("Maximum value of z for more precise matrix (to 0.20)")
zmax2=max(max(task2Matrix2))
disp("Obtained values 'zmax' and 'zmax2' are the same, 0.30 is precise
enough")
%obtained values are the same
%final error calculation:
disp("Delta z supremum 1 value:")
DeltaZsup1=zmax*eps
%============================TASK 3============================

bin=de2bi(0:511);
bin(bin==0)=-1;
task1=@(x,y)(x + sin(y)./3)./(y.^3 + log(y));
DeltaZsup2=0;

for i=1:512

task3=@(x,y)((x.*(1+eps.*bin(i,1))+(sin(y.*(1+eps.*bin(i,2)))/3).*(1+eps.*b
in(i,3))).*(1+eps.*bin(i,4))).*(1+eps.*bin(i,5)).*(1+eps.*bin(i,6)))./((((y.
*(1+eps.*bin(i,7)))^3).*(1+eps.*bin(i,7))+log(y.*(1+eps.*bin(i,2)).*(1+eps.
*bin(i,9))))).*(1+eps.*bin(i,8)));
    for j=1:0.30:10
        for k=1:0.30:10
            if DeltaZsup2 < abs((task3(j,k)-task1(j,k))/task1(j,k))
                DeltaZsup2 = abs((task3(j,k)-task1(j,k))/task1(j,k));
            end
        end
    end
end
disp("Delta z supremum 2 value:")
DeltaZsup2
```