SI 211: Numerical Analysis
Solution of Homework 1
Provided by 陈宏博

1. (a) x = π
   Exact solution is: 0 and Matlab solution is: -1.54597494809480e-13.
   So the numerical approximation error is: -1.54597494809480e-13.
   The reason for this error is that π is stored with finite precision in the computers.

   (b) f(x) is factorable, we can rewrite it as

$$a_0 = x,$$
$$a_1 = 10^4 a_0$$
$$a_2 = \sin(a_1)$$
$$a_3 = \frac{a_2}{a_0}$$

   In the worst case, the numerical errors associated with evaluating f(x)

$$\Delta a_0 \approx eps$$
$$\Delta a_1 \approx |10^4| * \Delta a_0 + eps$$
$$\Delta a_2 \approx |\cos(a_1)| * \Delta a_1 + eps$$
$$\Delta a_3 \approx \left|\frac{a_2}{a_0^2}\right| * \Delta a_0 + \left|\frac{1}{a_0}\right| * \Delta a_2 + eps$$

   We have error equals $\Delta a_3 \approx 0.04$ or 1e-2 (calculated by MATLAB with respect to above steps).

2. (a) Function *diff* code in Matlab:

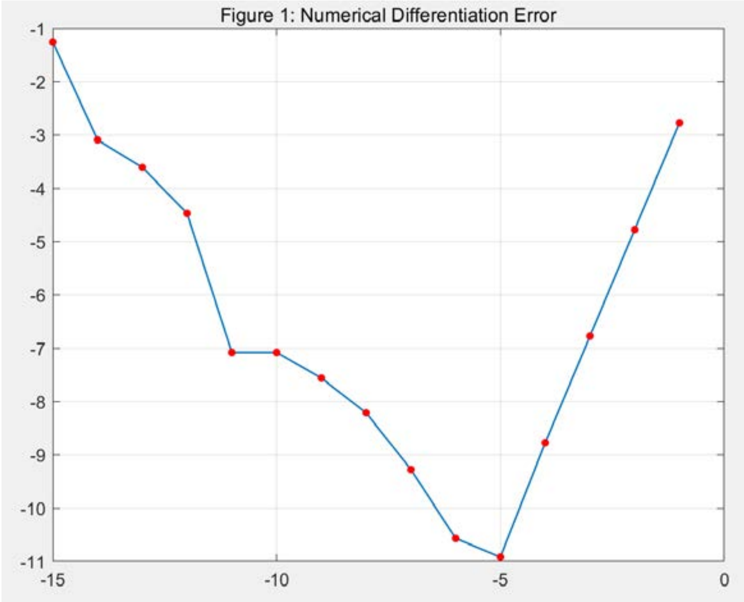
```
function r = newdiff(f,x,h)
   b = 2*h;
   r = (f(x+h)-f(x-h))/b;
end
```

   (b) Evaluate the derivative of the function *f(x)=exp(x)* at *x=0*, and h∈ $[10^{-15}, 10^{-1}]$

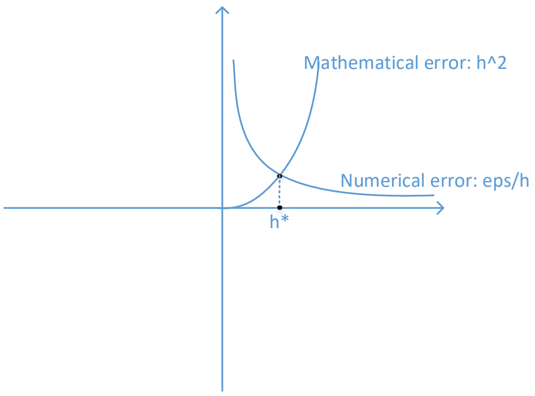
```
for i = 1 : 15
   h(i) = 10^-i;
end
x = 0.0;
exact_value = 1.0;
f = @(x) exp(x);
for i = 1 : 15
   e(i)=abs(newdiff(f,x,h(i))-exact_value);
end
```

```
x = log10(h);
y = log10(e);
plot(x,y);hold on
plot(x,y,'r.','MarkerSize',10);
```

The result is shown in the figure 1



Interpret the result:
The numerical differentiation error is produced by numerical error and mathematical error together, the error dominated by h^2 when h is greater than h^*, in the other hand, the error dominated by $\frac{eps}{h}$ when h is less than h^*.



$$\text{error(h)} \approx h^2 + \frac{eps}{h}$$

$$\frac{d}{dh} error(h^*) = 2h^* - \frac{eps}{{h^*}^2} = 0$$

$$h^* = \sqrt[3]{eps} \approx 10^{-5}$$

3. Algorithmic differentiation

$$\begin{aligned} b_0 &= 1 \\ b_1 &= -\sin(a_0) * b_0 \\ b_2 &= a_1 * b_1 + b_1 * a_1 \\ b_3 &= \cos(a_1) * b_1 \\ b_4 &= a_2 * b_3 + b_2 * a_3 \\ \mathrm{f}'(\mathrm{x}) &= b_4 \end{aligned}$$

The derivative of *f(x)* at *x = 0* is: 0 using AD code, the actual derivative is 0, so the numerical error is 0, and the order of magnitude of the numerical error also is 0.

Julia Code

```
import Base.*
import Base.sin
import Base.cos
mutable struct ADV
      a
      b
end

function *(A::ADV,B::ADV)
    return ADV(A.a*B.a,A.a*B.b+A.b*B.a);
end

function sin(A::ADV)
    return ADV(sin(A.a),cos(A.a)*A.b)
end

function cos(A::ADV)
    return ADV(cos(A.a),-sin(A.a)*A.b)
end

function f(x)
    return sin(cos(x))*cos(x*x)
end
```

Get the Result

```
x = ADV(0,1)
f(x)
ADV(0.8414709848078965, -0.0)
```