Problem Set 1 **R08244002 Shin-Rong, Tsai**

- **Machine Precision**

1. Results

machine_precision.c

```
cindytsai@TURQUOISEA /cygdrive/d/GitHub/Computational_Physics/Assignment/ProblemSet_1
$ gcc machine_precision.c -o a.out

cindytsai@TURQUOISEA /cygdrive/d/GitHub/Computational_Physics/Assignment/ProblemSet_1
$ ./a.out
single precision (float),
machine precision epsilon : 5.9604644775e-08     (2^-24)
double precision (double),
machine precision epsilon : 1.1102230246e-16     (2^-53)
```

2. Discussion

After running the code *machine_precision.c*, we get the biggest number (ε) for each data type that the computer "cannot" distinguish.

Data Type	ε	
Single precision (float)	$5.9604644775\times 10^{-8}$	2^{-24}
Double precision (double)	$1.1102230246\times 10^{-16}$	2^{-53}

For single precision number, there are 23 fraction bits. So the biggest number that the computer cannot distinguish is 2^{-24}, which is reasonable. Since the number shifts out of the range of single precision number's fraction bits.

This reason also stands for double precision number.

- **Richardson Extrapolation**

1. Results

richardson_extrapolation.c

```
cindytsai@TURQUOISEA /cygdrive/d/GitHub/Computational_Physics/Assignment/ProblemSet_1
$ gcc richardson_extrapolation.c -o a.out

cindytsai@TURQUOISEA /cygdrive/d/GitHub/Computational_Physics/Assignment/ProblemSet_1
$ ./a.out
f(x) = x * exp(x)
f''(x) = 2 * exp(x) + x * exp(x)
f''(2) = 2.9556224396e+01
============================================================================
Using Richardson Extrapolation Method:
 h              D1              D2              D3              D4
0.40  3.0151567480e+01
0.20  2.9704268474e+01  2.9555168806e+01
0.10  2.9593186100e+01  2.9556158642e+01  2.9556224631e+01
0.05  2.9565461742e+01  2.9556220290e+01  2.9556224399e+01  2.9556224396e+01
```

2. Discussion

I'm not sure if the first course derived the Richardson Extrapolation for the second order derivative, since I have not chosen this course yet.

So I try to reason this relationship for the second order case.

$$f(x-h) = f(x) - h \cdot f'(x) + \frac{h^2}{2!} \cdot f''(x) - \frac{h^3}{3!} \cdot f^{(3)}(x) + \cdots$$

$$f(x+h) = f(x) + h \cdot f'(x) + \frac{h^2}{2!} \cdot f''(x) + \frac{h^3}{3!} \cdot f^{(3)}(x) + \cdots$$

Since we want the second order term, we must try to cancel out the first order.
So add two equation together and organize it, we have

$$f''(x) = \frac{f(x+h) - 2 \cdot f(x) + f(x-h)}{h^2} - 2 \cdot \frac{h^2}{4!} \cdot f^{(4)}(x) - 2 \cdot \frac{h^4}{6!} \cdot f^{(6)}(x) - \cdots.$$

Let

$D_1(h) = \frac{f(x+h) - 2 \cdot f(x) + f(x-h)}{h^2}$, this is same as three-point formula. When we change h to h/2, we can have the similar form of equation with different coefficients. So that with these two equations, we can cancel the 4th order term, and have a better approximation.

With this relation, we can derive Richardson Extrapolation for second order derivatives. And because D_i will canceled out i terms of irrelevant fluctuation, it will be more accurate as i becomes larger. We can see this from the code's result above.