

수치해석 HW#1

2018008559 컴퓨터소프트웨어학부 신상운

1. Introduction

1. 목적

float와 double 자료형의 machine accuracy를 2가지 방법으로 찾아본다.

2. Process & Result

2.1 Method 1

machar()를 이용하여 찾아본다.

"C:\Users\Wkas12\OneDrive\바탕 화면\한양대\3학년 2학기\수치해석\

```
quantity: numeric_limits<Doub> says (we calculate)
radix: 2 (2)
mantissa digits: 53 (53)
round style: 1 (5) [our 5 == IEEE 1]
guard digits: [not in numeric_limits] (0)
epsilon: 2.22045e-16 (2.22045e-16)
neg epsilon: [not in numeric_limits] (1.11022e-16)
epsilon power: [not in numeric_limits] (-52)
neg epsilon power: [not in numeric_limits] (-53)
exponent digits: [not in numeric_limits] (11)
min exponent: -1021 (-1022)
max exponent: 1024 (1024)
minimum: 2.22507e-308 (2.22507e-308)
maximum: 1.79769e+308 (1.79769e+308)
```

```
Process returned 0 (0x0)   execution time : 0.752 s
Press any key to continue.
```

왼쪽결과는 C++ 내장함수를 이용한 결과이고 오른쪽 괄호 안의 결과가 machar()을 이용하여 나온 결과이다.

위의 실험결과 double 자료형의 machine accuracy는 $2.22045e-16$ 이었다.

float 자료형은 nrtypes_nr.h, nrutil_nr.h에서 DP의 자료형을 float로 바꾸고 실행시켜주면 된다.

```
선택 "C:\Users\Wkas12\OneDrive\바탕 화면\한양대\3학년 2학기\수치해
quantity: numeric_limits<Doub> says (we calculate)
radix: 2 (2)
mantissa digits: 24 (24)
round style: 1 (5) [our 5 == IEEE 1]
guard digits: [not in numeric_limits] (0)
epsilon: 1.19209e-07 (1.19209e-07)
neg epsilon: [not in numeric_limits] (5.96046e-08)
epsilon power: [not in numeric_limits] (-23)
neg epsilon power: [not in numeric_limits] (-24)
exponent digits: [not in numeric_limits] (8)
min exponent: -125 (-126)
max exponent: 128 (128)
minimum: 1.17549e-38 (1.17549e-38)
maximum: 3.40282e+38 (3.40282e+38)

Process returned 0 (0x0)    execution time : 0.414 s
Press any key to continue.
```


실험결과 float 자료형의 machine accuracy는 $1.19209e-07$ 이었다.

2.2 Method 2

직접 짠 코드 get_eps()를 이용하여 찾아본다.

코드는 $1 = 1 + 2^{-n}$ 을 만족하는 가장 작은 자연수 n 을 찾도록 다음과 같이 짰다.

```
for(;;){  
    if(one == tmp + pow(2,-cnt)) break;  
    cnt++;  
}
```

 "C:\Users\Wkas12\OneDrive\바탕 화면\한양대\3학년 2학기\수치해석\

```
double  
1 = 1 + 2^-53  
epsilon : 2.22045e-16  
neg epsilon : 1.11022e-16
```

```
float  
1 = 1 + 2^-24  
epsilon : 1.19209e-07  
neg epsilon : 5.96046e-08
```

```
Process returned 0 (0x0)   execution time : 0.703 s  
Press any key to continue.
```

실험결과 값이 완전히 똑같아지는 neg_epsilon값은 double, float 각각 1.11022e-16, 5.96046e-08이었고,

값이 바뀌는 가장 작은 값 epsilon 즉, machine accuracy 값은 각각 neg epsilon에 2를 곱한 2.22045e-16, 1.19209e-07이었다.

get_eps(), machar()의 neg_epsilon, epsilon 값은 완전히 같았다.

3. Reference

[1] <http://numerical.recipes/webnotes/nr3web27.pdf>

[2] <http://numerical.recipes/book/book.html>