

MAT3008 - Homework 2

박준영

1 Numerical recipes in C Summary

1.1 How to use pointers for memory allocation

C에서 메모리는 malloc 함수를 이용해 heap 영역에 동적으로 할당받을 수 있으며, free 함수를 이용해 할당 해제할 수 있다. 이때, malloc 함수는 할당받은 메모리의 주소를 pointer 타입으로 반환한다.

N 차원 벡터를 할당받을 때는 malloc 함수를 통해 다음과 같이 메모리를 할당받을 수 있다.

```
type* vec = malloc(N * sizeof(type));
```

위 코드에서 type은 int, float 등 원하는 타입을 적으면 된다. 할당 해제는 다음과 같이 수행할 수 있다.

```
free(vec);
```

heap 영역 뿐만 아니라 stack 영역의 배열을 pointer로 지시할 수 있다. 이러한 사실을 이용해 zero-offset인 C에서 unit-offset을 사용하도록 할 수 있다.

```
int a[N]; int* aa = a - 1;
```

위의 코드를 사용하면 이제 aa[1]이 a[0]에 대응되므로 1부터 index가 시작하는 unit-offset을 사용할 수 있게된다.

1.2 How to use pointer to function

Pointer to function이란 함수의 실행 가능한 코드의 위치를 가리키는 pointer이다. 지칭할 함수의 반환 타입과 함수 인자와 함께 선언하게 되는데, 예컨대 두 개의 int 형 데이터를 인자로 받고 int 형 데이터를 반환하는 함수의 pointer to function은 다음과 같이 선언할 수 있다.

```
int (*func)(int, int);
```

일반 pointer와 같이 assignment operator(=)를 이용해 값을 할당할 수 있고, 일반 함수 호출문과 같이 pointer to function이 지칭하는 함수를 호출할 수 있다.

2 Solve the problems

#3.6

첫 번째 급수를 사용한 경우 결과는 다음과 같다.

Iteration	Approx Value	Approx Relative Err	Relative Err
0	1.000000	N/A	-147.413159
1	-4.000000	5.000000	594.652636
2	8.500000	3.125000	-1260.511852
3	-12.333333	2.450980	1831.428962
4	13.708333	2.111486	-2033.497056
5	-12.333333	1.899696	1831.428962
6	9.368056	1.759572	-1389.342719
7	-6.132937	1.654665	911.208482
8	3.555184	1.579687	-526.636019
9	-1.827105	1.513927	272.166481
10	0.864039	1.472901	-127.234769
11	-0.359208	1.415732	54.311254
12	0.150478	1.418916	-21.332922
13	-0.045555	1.302737	7.760992
14	0.024457	1.536858	-2.629692
15	0.001119	0.954230	0.833869
16	0.008412	-6.515129	-0.248494
17	0.006267	0.254981	0.069848
18	0.006863	-0.095069	-0.018580
19	0.006706	0.022846	0.004691

두 번째 급수를 사용한 결과는 다음과 같다.

Iteration	Approx Value	Approx Relative Err	Relative Err
0	1.000000	N/A	-147.413159
1	0.166667	0.833333	-23.735527
2	0.054054	0.675676	-7.022333
3	0.025424	0.529661	-2.773216
4	0.015296	0.398343	-1.270182
5	0.010939	0.284868	-0.623480
6	0.008840	0.191847	-0.312020
7	0.007775	0.120519	-0.153897
8	0.007230	0.070048	-0.073069
9	0.006959	0.037458	-0.032874
10	0.006832	0.018385	-0.013885
11	0.006775	0.008287	-0.005483
12	0.006752	0.003441	-0.002023
13	0.006743	0.001322	-0.000698
14	0.006739	0.000472	-0.000226

15	0.006738	0.000157	-0.000069
16	0.006738	0.000049	-0.000020
17	0.006738	0.000014	-0.000005
18	0.006738	0.000004	-0.000001
19	0.006738	0.000001	-0.000000

위 결과를 통해 두 번째 series가 더 빠르게 수렴하는 것을 볼 수 있다.

#3.7

(1) 3-digit arithmetic with chopping.

$$6x = 6 \times 0.577 = 0.346 \times 10$$

$$x^2 = (0.577)^2 = 0.332$$

$$3x^2 = 3 \times 0.332 = 0.996$$

$$1 - 3x^2 = 0.004$$

$$f'(0.557) = (0.346 \times 10)/(0.004)^2 = 216 \times 10^3$$

(2) 4-digit arithmetic with chopping.

$$6x = 6 \times 0.577 = 0.3461 \times 10$$

$$x^2 = (0.577)^2 = 0.3329$$

$$3x^2 = 3 \times 0.3329 = 0.9986$$

$$1 - 3x^2 = 0.0014$$

$$f'(0.557) = (0.3461 \times 10)/(0.0014)^2 = 1765 \times 10^3$$

#4.2

$\cos(\pi/3) = 0.5$ 이다. 이때, 유효 숫자 2자리까지 오차가 없으려면 다음을 만족해야한다.

$$(\text{Approx} - 0.5) < 0.005$$

따라서 위 조건을 만족할 때까지 Maclaurin series에 term을 추가한 결과는 다음과 같다.

Iteration	Approx Value	Approx Relative Err (%)	Relative Err (%)
0	1.000000	N/A	-100.000000
1	0.451689	121.391441	9.662271
2	0.501796	-9.985639	-0.359240

즉, 다음과 같은 Maclaurin series로 유효숫자 2개까지 정확한 $\cos(\pi/3)$ 를 근사할 수 있다.

$$\cos(\pi/3) \approx 1 - \frac{(\pi/3)^2}{2} + \frac{(\pi/3)^4}{4!}$$

#4.5

$f(x) = 25x^3 - 6x^2 + 7x - 88$ 에 대하여 $f(3) = 554$ 이다.

$f'(x) = 75x^2 - 12x + 7$, $f''(x) = 150x - 12$, $f'''(x) = 150$ 이므로 $f(1) = -62$, $f'(1) = 70$, $f''(1) = 138$, $f'''(1) = 150$ 이다.

(1) zero-order Taylor series

$g(x) = f(1)$ 에서 $g(3) = -62$ 이므로 이때의 true percent relative error는 다음과 같다.

$$\frac{554 - (-62)}{554} \times 100 \approx 111.19\%$$

(2) first-order Taylor series

$g(x) = f(1) + f'(1)(x - 1)$ 에서 $g(3) = 78$ 이므로 이때의 true percent relative error는 다음과 같다.

$$\frac{554 - 78}{554} \times 100 \approx 85.92\%$$

(3) second-order Taylor series

$g(x) = f(1) + f'(1)(x - 1) + \frac{f''(1)}{2!}(x - 1)^2$ 에서 $g(3) = 354$ 이므로 이때의 true percent relative error는 다음과 같다.

$$\frac{554 - 354}{554} \times 100 \approx 36.10\%$$

(4) third-order Taylor series

$g(x) = f(1) + f'(1)(x - 1) + \frac{f''(1)}{2!}(x - 1)^2 + \frac{f'''(1)}{3!}(x - 1)^3$ 에서 $g(3) = 554$ 이므로 이때의 true percent relative error는 다음과 같다.

$$\frac{554 - 554}{554} \times 100 = 0\%$$

#4.12

$\frac{gm}{c} (1 - e^{-(c/m)t})$ 에서 변수는 c 와 m 이다. 따라서 v 를 c 와 m 으로 각각 편미분 하면 다음과 같다.

$$\begin{aligned} \frac{\partial v}{\partial c} &= -\frac{gt}{c^2} (1 - e^{-(c/m)t}) + \frac{gm}{c} e^{-(c/m)t} \\ \frac{\partial v}{\partial m} &= \frac{g}{c} (1 - e^{-(c/m)t}) - \frac{gt}{m} e^{-(c/m)t} \end{aligned}$$

위 두 식에서 각 변수에 수를 대입하면 $\partial v / \partial c \approx -1.3881$, $\partial v / \partial m \approx 0.3470$ 이다. 따라서

$$\begin{aligned}
\Delta v &= \left| \frac{\partial v}{\partial c} \right| \Delta c + \left| \frac{\partial v}{\partial m} \right| \Delta m \\
&= |-1.3881| \times 1.5 + |0.3470| \times 2 \\
&= 2.7761
\end{aligned}$$

$$\therefore v \approx 30.484 \pm 2.7761$$