**Homework#2**

2017029589 컴퓨터소프트웨어학부 류지범

1. Chapter 1 summarize

how to use pointers for memory allocation

포인터란 메모리의 주소값을 저장하는 변수이다. 배열은 데이터의 연속을 의미하며 a라는 배열을 선언하게 되면 \*a는 항상 a[0]을 가리키게 된다.

1차원 배열에서 b[4]를 선언하게 되면 b[0], b[1], b[2], b[3]이 유효한 주소를 가진다. index의 범위는 0~3이 되고, 우리가 다수의 알고리즘을 사용할 때 이러한 index는 불편함을 안겨준다. 일반적인 알고리즘들은 1~M의 범위를 가지지만 M의 크기를 가지는 배열을 선언하면 0~M-1의 범위를 가지게 되기 때문이다. 이때 새로운 bb라는 포인터를 선언하여 이 문제를 해결할 수 있다. `\*bb = b - 1;`을 사용하면 bb[1], bb[2], bb[3], bb[4]로 4의 크기를 가지며 1~4의 범위를 가질 수 있게 된다. 이러한 bb를 `unit-offset vector`라고 한다. 이 방식을 사용하면 여러 알고리즘에서 이득을 얻을 수 있다.

how to use pointer to function

위의 방식을 이용하여 범위를 조정해주는 함수들이 `nrutil.c`에 존재하며 `float \*vector(long nl, long nh)`, `int \*ivector(long nl, long nh)` 등이 있다. 물론 메모리 공간을 free시켜주는 함수들도 존재한다.

2차원배열도 1차원배열과 마찬가지로 데이터들의 연속을 의미하기 때문에, 이러한 방식은 2차원 배열에도 적용되며 `pointer to pointer` 를 사용한다.

2. Solve Problem

3.6

code in 3-6.cpp

true value: 6.737949e-03

first method: 6.745540e-03

second method: 6.737948e-03

relative error of first method: -1.126618e-03

relative error of second method: 2.158695e-07

case of terms are 100

first method: 6.737947e-03

second method: 6.737947e-03

relative error of first method: 2.969621e-07

relative error of second method: 2.969620e-07

relative error is lower when terms are added

3.7

3 digit chopping

- 6x = 3.462 = 0.3462 \* 10 -> 0.346 \* 10 = 3.46

- x^2 = 0.332929 -> 0.332

- 3x^2 = 0.996

- 1 - 3x^2 = 0.004

- 3.46/0.004^2 = 216250

4 digit chopping

- 6x = 3.462 = 0.3462 \* 10 -> 3.462

- x^2 = 0.332929 -> 0.3329

- 3x^2 = 0.9987

- 1 - 3x^2 = 0.0013

- 3.462 / 0.0013^2 = 2048520.710059171597633

4.2

code in 4-2.cpp

- `es = (0.5 \* 10^(2-n))% = 0.005` is two significant figures

- `et` is relative error and `ea`is Approximate relative error

- program will terminate when `ea < es`

result

estimated value: 4.516886e-01, relative error: 9.66227% approximate relative error: 1.21391

estimated value: 5.017962e-01, relative error: -0.35924% approximate relative error: 0.0998564

estimated value: 4.999646e-01, relative error: 0.00708693% approximate relative error: 0.00366353

program terminate with 3 step

4.5

- true value = f(3) = 554

- zero order

- p0(x) = f(x0) -> f(1) = -62

- `(554 - (-62)) / 554 => 111.19133574%`

- first order

- p1(x) = f(x0) + f'(x0)(x-x0) = 78

- `(554 - 78) / 554 => 85.9205776%`

- second order

- p2(x) = f(x0) + f'(x0)(x - x0) + f''(x0)(x-x0)^2/2! = 354

- `(554 - 354) / 554 => 36.101083%`

- third order

- p3(x) = f(x0) + f'(x0)(x - x0) + f''(x0)(x-x0)^2/2! + f'''(x0)(x-x0)^3/3! = 554

- `(554 - 554) / 554 = 0%`

4.12

텍스트이(가) 표시된 사진

자동 생성된 설명- solution process is in 4-12.imeg

result

30.484 ± 2.776