

비선형 방정식의 풀이 기초

고급소프트웨어실습1: 4주차 실습

실습 내용

- 실습 문제 1-1
 - Newton-Raphson, Secant 방법을 구동하는 프로그램 구현
- 실습 문제 1-2
 - Newton-Raphson 방법으로 특정 함수의 모든 실근을 계산
- 실습 문제 1-4
 - 정밀도(precision)와 구한 근의 오차에 대한 상관관계
- 실습 문제 1-3, 1-5는 이번 실습에서 제외

실습 방식

- 사이버 캠퍼스->과제->4주차 실습에서 첨부파일을 다운
- 사이버 캠퍼스 4주차 강의 내용을 바탕으로 실습 문제 1-1, 1-2, 1-4 부분을 작성
- 실습을 다하면, 작성한 코드와 입력 값, 출력을 캡처 하여 slack으로 확인
- 조교 확인 후 실습 파일을 제출
- 사이버 캠퍼스->과제->4주차 실습에 작성한 코드를 첨부파일로 제출

실습 파일 구성

- 4주차 강의 내용을 바탕으로 다음을 작성하여 압축하여 제출
 - main.cpp : 실습 문제를 수행하는 메인 함수
 - my_solver.h : 종료조건 인자를 포함한 헤더
 - function.cpp
 - program1_1.cpp
 - program1_2.cpp
 - sp_function.cpp
 - sp_program1_1.cpp
 - sp_program1_2.cpp

실습 파일 구성

- [sp_]program1_1.cpp
 - Newton-Raphson 방법을 구현하는 부분
- [sp_]program1_2.cpp
 - Secant 방법을 구현하는 부분

실습 파일 구성

- [sp_]function.cpp
 - 함수와 도함수를 정의하는 부분
- 실습에 사용하기 위해 정의한 함수를 모두 포함해야 함
(과제 제출시에는 과제에서 정의한 함수도 모두 포함)
- 채점 시 다른 파일로 대체하기도 하기 때문에 형식에 맞게 작성

실습 파일 구성

- main.cpp
 - 실습 문제를 풀기 위해 프로그램을 실행시키는 부분
 - 채점 시 목적에 따라 다른 파일로 대체될 수 있음

• 예)

```
#include "my_solver.h"

int main() {
    program1_1();    // Newton-Raphson method (double type)
    program1_2();    // Secant method

    sp_program1_1(); // Newton-Raphson method (float type)
    sp_program1_2(); // Secant method
}
```

비선형 방정식의 풀이

- 방정식의 근

- 임의의 함수 $f: \mathbb{R} \rightarrow \mathbb{R}$ 에 대해, $f(\alpha) = 0$ 을 만족하는 값 α

- 방정식 $f(x) = 0$ 를 푼다는 것은 근을 찾는 것을 의미

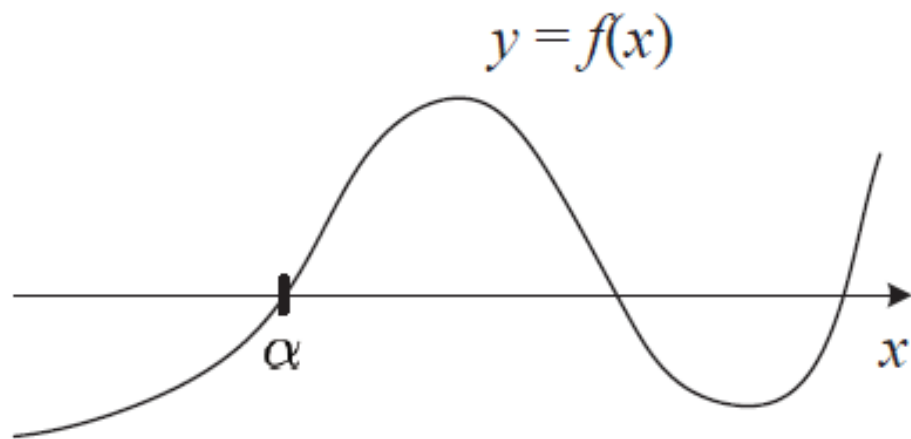


그림 2: 방정식의 근

비선형 방정식의 풀이

- Newton-Raphson 방법
 - 함수와 도함수를 이용한 방정식 풀이 방법

```

$$\begin{aligned} & x_0 \leftarrow \text{an initial guess;} \\ & \text{for } n = 0, 1, 2, 3, \dots \quad \{ \\ & \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}; \\ & \} \end{aligned}$$

```

- Secant 방법
 - Newton-Raphson 방법의 근사
 - 도함수를 근사하여 사용

```

$$\begin{aligned} & x_0, x_1 \leftarrow \text{two initial guesses;} \\ & \text{for } n = 1, 2, 3, \dots \quad \{ \\ & \quad x_{n+1} = x_n - f(x_n) \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}; \\ & \} \end{aligned}$$

```

비선형 방정식의 풀이

- 반복문의 종료 조건

- 반복문을 적절한 순간에 종료해야 해를 얻을 수 있음

- 일반적으로 다음과 같은 종료 조건을 사용하여 반복문을 종료

(a) 현재 구한 x_{n+1} 에 대해 함수 값이 충분히 작은가? $\rightarrow \boxed{|f(x_{n+1})| < \delta}$ ¹⁾

(b) 충분히 많은 회수만큼 반복문을 수행하였는가? $\rightarrow \boxed{n \geq N_{max}}$ ²⁾

(c) 현재 구한 x_{n+1} 이 직전에 구한 x_n 에 비해 더 이상 의미 있는 진전을 하지 않는가? \rightarrow

$$\boxed{|x_{n+1} - x_n| < \varepsilon} \quad ^{3)}$$

실습/과제 제출

제출 안내

- 사이버캠퍼스 실습/과제 란을 통해 제출
- 제출 기한
 - 실습 : 다음날 14:59분 까지
 - Late 없음. 0점 처리함
 - 예시) 수요일 반의 경우,
 - 실습 : 목요일 14시 59분 까지

제출 방식

- 제출 양식

- 첨부 파일

- [고소실_0주차실습]0반_20181600_홍길동.zip

- [고소실_0주차과제]0반_20181600_홍길동.zip

- 예시)[고소실_4주차실습]2반_20181600_홍길동.zip

- 형식 틀릴 시 감점!

- 형식 미 준수로 인한 불이익은 본인 책임(과제 유실 우려)

첨부 파일 제출 시 유의사항

- 제출 순서
 - 빌드->솔루션 정리 후 저장
 - Debug, Release 폴더 삭제
 - .vs 폴더 삭제
 - 프로젝트 폴더 자체를 압축
- 만약 이렇게 했는데도 압축파일 크기가 30MB를 넘는다면, 실습 시 사용했던 cpp 파일만 압축하여 제출