

수치해석 과제

과제번호: HW1

과제제목: The Secant Method

자가 채점: 4

1점: 과제내용 설명

2점: + 소스코드

3점: + 실행결과

4점: + 실행결과가 유효숫자까지 정확

제 출 날 짜: 25 09 16

학 과: 소프트웨어학부

학 번: 2020203090

이 름: 한용욱

과제 내용

문제 설명

다음 방정식의 근을 Secant method로 구하는 문제이다

$$f(x) = x^3 + 4x^2 - 10 = 0$$

문제에서 시작 값은 $p_0 = 1, p_1 = 2$ 로 둔다

$n = 2, 3, 4, 5, 6$ 에 대해 $p_n, f(p_n)$ 을 표 형태로 출력한다

필요한 수학 : 할선법의 유도과 점화식

할선법은 어떤 구간의 함수를 할선으로 근사하는 방법이다

할선과 x축의 교점을 다음 점으로 삼아 반복적으로 방정식의 근으로 근사하는 방법이다

할선은 두 점 $(p_{n-1}, f(p_{n-1})), (p_n, f(p_n))$ 을 지나는 직선의 방정식이므로 x절편을 구해

아래의 식과 같이 다음 점을 구한다

$$p_{n+1} = p_n - \frac{f(p_n)(p_n - p_{n-1})}{f(p_n) - f(p_{n-1})}$$

코드 구현법

```
double f(double x)
```

과제의 대상 함수 $f(x) = x^3 + 4x^2 - 10$ 을 그대로 구현한다

```
double next_p(double (*f)(double), double p_n, double p_n_1)
```

이전 두 점을 받아 점화식 $p_{n+1} = p_n - \frac{f(p_n)(p_n - p_{n-1})}{f(p_n) - f(p_{n-1})}$ 그대로 계산해

결과 p_{n+1} 을 반환한다

```
double secant(int n, double (*f)(double), double p0, double p1)
```

p_0, p_1 를 초기값으로 받아 Secant method로 구한 p_n 을 반환한다

$n = 0$ 이면 $p_0, n = 1$ 이면 p_1 을 반환하고, $n \geq 2$ 에서는 `next_p`를 반복 호출하여 p_n 을 구한 뒤 반환한다

```
int main()
```

$n = 2, 3, 4, 5, 6$ 에 대해 `secant(n, f, 1, 2)`를 호출하여 $p_n, f(p_n)$ 을 출력한다

```

// 전체 소스코드

#include <stdio.h>
#include <math.h>

double f(double x)
{
    return pow(x, 3) + 4 * pow(x, 2) - 10;
}

double next_p(double (*f)(double), double p_n, double p_n_1)
{
    return p_n - f(p_n) * (p_n - p_n_1) / (f(p_n) - f(p_n_1));
}

double secant(int n, double (*f)(double), double p0, double p1)
{
    if (n == 0) return p0;
    else if (n == 1) return p1;

    double p_n = 0, p_n_1 = p1, p_n_2 = p0;

    for(int i = 1; i < n; i += 1)
    {
        p_n = next_p(f, p_n_1, p_n_2);
        p_n_2 = p_n_1; p_n_1 = p_n;
    }

    return p_n;
}

void format(int n, double p_n, double f_p_n)
{
    printf("%d", n);

    if (p_n >= 0) printf(" ");
    else printf(" ");

    printf("%.10f", p_n);

    if (f_p_n >= 0) printf(" ");
    else printf(" ");

    printf("%.10f", f_p_n);

    printf("\n");
}

int main()
{
    printf("n      p_n      f(p_n)\n");
    for(int i = 2; i < 7; i += 1)
    {
        double p_n = secant(i, f, 1, 2);
    }
}

```

```
        format(i, p_n, f(p_n));  
    }  
  
    return 0;  
}
```

실행결과

n	p_n	f(p_n)
2	1.2631578947	-1.6022743840
3	1.3388278388	-0.4303647480
4	1.3666163947	0.0229094308
5	1.3652119026	-0.0002990679
6	1.3652300011	-0.0000002032