목차

1. 예제: 4.3 급수전개를 이용한 함수의 근사

$X_{i+1} = \pi\ /3$에서 $f(x) = \cos x$를 근사하기 위해, $x_i = \pi/4$에서의 $f(x)$값 및 그 도함수 값들을 이용하여 Taylor 급수 전개를 n=0부터 6까지 수행하라. 이는 $h = \frac{\pi}{3} - \frac{\pi}{4} = \pi/12$임을 의미한다.

풀이

$$f\left(\frac{\pi}{3}\right) = 0.5$$

라는 참값은 이미 기존에 알려져 있으며 이 참값을 기준으로 상대오차를 구해볼 수 있다.

0차 cos(pi/4)에서의 값은

$$f\left(\frac{\pi}{4}\right) \text{ float}, 10 \ \rightarrow 0.7071067812 = 0.707$$

무한 소수로 표현되는 것을 float를 통해서 소수점 유효숫자 10자리 단위로 출력하게 설정하였다.

$$\varepsilon := \left|\frac{\left(0.5 - f\left(\frac{\pi}{4}\right)\right)}{0.5}\right| \text{ float}, 10 \ \rightarrow 0.4142135624 = 0.414$$

상대오차를 구하기 위한 함수를 지정하고 다음과 같은 백분율 상대 오차를 구한다.

계속해서

$$f(x) \text{ series}, x, \left(x = \frac{\pi}{4}\right), 7, \text{float}, 9$$

다음과 같이 series함수를 급수를 지정을 하고, 실수의 9번째 자리까지 출력하게 설정을 하면

$$1.0 - 0.308425138 - 0.000325991887 + 0.0158543442 + -0.707143046\cdot x + 0.555388849 - 0.353714603\cdot(x - 0.785398163)^2 + 0.117442109\cdot(x - 0.785398163)^3$$

자동으로 급수의 값을 계산해 준다. 각 항에 대해서 근사한 참값에 대한 오차를 구하면

Order n	$F^{(n)}(x)$	$F(\pi/3)$	$\lvert\varepsilon_t\rvert$
0	$\cos x$	0.707106781	41.4
1	$-\sin x$	0.521986659	4.40
2	$-\cos x$	0.497754491	0.449
3	$\sin x$	0.499869147	1.62x10^-2
4	$\cos x$	0.500007551	1.51x10^-3
5	$-\sin x$	0.500000304	6.08x10^-5

6	$-\cos x$	0.499999988	2.44x10^-6

테일러 급수에서 주의해야 할 것은 좌변과 우변이 모든 x에 대해서 같은 것이 아니라 x=a 근처에서만 성립한다. 즉 x가 a에서 멀어지면 멀어질수록 큰 오차를 가지게 되며 근사다항식의 차수는 높으면 높을수록 함수f(x)에 대해서 좀더 잘 근사하게 된다. 3차항을 추가할 때까지 오차는 0.026%로 줄어들게 되는데 이는 근사값이 참값의 99.974%했다는 것을 의미하며, 이보다 더 항을 추가하면 스텝사이즈를 줄이니 그만큼 오차를 줄일 수 있겠지만 반대로 마무리오차가 증가할 수 있기 때문에 유의미한 오차의 감소는 얻기 힘들 수 있다. 실제 환경에서도 절대적인 값을 발견 하는 것이 아니라, 설령 시도하더라도 불가능 하므로 어느 적절한 위치를 찾는 것이 수치해석의 과제이다.

2. 문제: 4.13

다음 함수에서 f(3)을 계산하기 위해 x=1을 기준점으로 하여 0차에서부터 3차까지의 Taylor 급수전개를 사용하라. 각 근사값에 대한 참 백분율 상대오차 ε를 구하라. $f(x) = 25x^3 - 6x^2 + 7x - 88$

풀이

$$f(x) := 25x^3 - 6x^2 + 7x - 88$$

예제 4.3과 마찬가지로 급수를 적용할 함수를 설정한 다음 참값을 구한다.

$$f(3) = 554$$

Mathcad를 통해서 참값은 f(3) = 554로 나온다.

$$f(x) \text{ series}, 4, x, x = 1 \rightarrow -132 + 70 \cdot x + 69 \cdot (x-1)^2 + 25 \cdot (x-1)^3$$

f(x)의 0부터 3차항까지 기준점은 1을 가지고 급수를 취하면 위와 같은 식이 나온다. 이를 표로 정리해보면

Order n	$F^{(n)}(x)$	F(3)	$\lvert\varepsilon_t\rvert$
0	$25x^3 - 6x^2 + 7x - 88$	-132	123.8%
1	$75x^2 - 12x + 7$	78	85.9%
2	$150x - 12$	354	36.1%
3	150	554	0%

3차항에서 상대오차 0%를 기록함을 알 수 있다.

결론

지금까지 해온 공학공부에서는 수학과 물리적인 개념을 배우고 원리를 이해하는 것이 목적이다. 이미 나온 공식에 단순 계산을 위해서 공학용 계산기를 이용한 것을 제외하면 별다른 컴퓨터의 사용이 없었으며, 제한적이었고 없어도 공부하는 것에 지장이 없을 만큼 크게 필요하지도 않았다. 하지만 수치해석에서는 이전 공학과목들과의 목표와 달리, 값을 찾는 것 자체에 의미가 있는 과목이다. 답을 찾는 수단으로 기존에 배운 학문적인 지식과 컴퓨터를 적극적으로 이용해서 적절한 답을 찾아나가는 과정을 배운다. 컴퓨터 소프트웨어로 대중적인 MATLAB와

다소 생소한 PTC에서 제작한 Mathcad가 있다. 교재가 많고 흔하게 접하기 쉬운 MATLAB 대신 교수님의 권유로 Mathcad를 이용해서 과제를 해결하였으며 각 예제와 문제를 풀어 보았다.

Mathcad를 쓰면서 아쉬웠던 점은 프로그램 자체의 문제이기 보다는 아직 익숙하지 않고 사용 방법에 대해 서투르다 보니 생긴 문제이다. 프로그램을 이용하는 것은 사람의 손을 최대한 덜 쓰기 위함이고, 편리함을 위한 것인데 아직 사용 방법의 미숙으로 데이터테이블을 만들 때 손으로 직접 하나하나 쳐보았는데, 다른 프로그래밍 언어처럼 조건문과 반복문을 적절하게 사용 하는 방법을 배워 하나의 프로그램을 작성하는 것이 연습이 된 다면 이 과정 역시 자동으로 데이터 테이블을 출력을 할 수 있을 것으로 본다.

MATLAB와 Mathcad의 프로그램 자체만 두고 보았을 때 차이점은 표현하는 방식에 있다. MATLAB의 경우 전용 언어인 동시에 C와 C++과 같은 프로그램을 직접 프로그래밍이 가능한 일반 언어이기 때문에 객체(object)지향 언어의 클래스 개념, 변수 지정 등 기존의 프로그램 형식을 따르는 것을 볼 수 있으며, 한줄 한줄 각 줄을 따라 위에서 아래로 절차적으로 진행되며, 이러한 개념이 있어야 체계적으로 학습하고 이용할 수 있다. Mathcad의 경우 Symbolic을 기반으로 직관적인 사용이 가능하며 마치 공책에 낙서처럼 문제풀이를 적어나가는 것처럼 worksheet에 바로바로 문제를 적고 답을 도출하는 것을 보여주며 한 노트에 어디에 적어 두어도 상관 없는 것처럼 형식에 있어서도 자유로운 것을 볼 수 있다. 특히 unit를 아무런 방해없이 사용하며 가장 큰 특징은 2차원 커서지원이다. 매트랩에서는 1차원 커서로 띄어쓰기, 공백을 생성할 때 쓰지만 Mathcad에서는 2차원 커서를 지원하며 공백을 넣는 용도가 아닌 연산자의 피 연산자를 지정하기 위해서 사용되는 점이 다르다. 추가적으로 이 과제를 하는데 필요한 급수 개념은 Mathcad에서는 자체적으로 내장되어 있어서 계산하는데 편리 하였지만 MATLAB에서는 지원은 하지만 직관적으로 출력 값을 보기가 어려워 난해 하였다. 프로그램 외적인 차이는 규모에 있다. MATLAB는 모든 다른 컴퓨터 언어에 비해서 많은 점유율을 차지하고 있는 것은 아니지만 상위 20위권에 들며, 이공계말고도 경제학에 있어서도 대학원 과정에서 널리 쓰이는 등 범용성을 자랑하며, 그만큼 교재나 인터넷 검색등에서도 쉽게 사용 방법을 찾을 수 있다. Mathcad에서는 범용성이 다소 떨어져 이용하는 사람이 적어 피드백을 받기 어려운 점이 있다. 이번 문제를 풀면서 Series를 이용해서 급수를 취하는데 특정한 x값에 대해서 전개하는 방법을 몰라서 한참 동안 찾아보았으나 인터넷으로는 답을 얻을 수가 없었다.