Homework #1

1.

Import random, numpy, math를 하여 실행한다.

[ 1718 1829 3060 3501 3646 5190 6662 7273 7908 10159 10452 11088

11411 14180 14199 17187 19234 19981 20766 20935 21750 22942 24930 25030

26007 26320 26599 27223 29402 33680 34757 34933 35502 35838 35982 37229

39044 39572 40643 42025 42066 42318 42461 43784 44650 45734 46353 46645

47782 48783 49983 53017 54413 54537 56021 58222 58342 58549 60796 60817

0 69296

0 69296

69806 73698 74233 75246 76481 76863 77037 77204 79485 79579 79916 80662

80811 80815 81553 82249 85069 85379 86249 87939 88369 90241 93026 96578

97286 97571 99321 99740]

2.

임의의 4x4 matrix 2개를 만들어 이를 곱한 matrixC를 생성한다.

그리고 gauss\_jordan\_elimination 함수를 생성하여 C|I로 역함수 C를 찾아낸다.

그리고 그 inverse C matrix를 직접 C와 곱해보아 identity\_matrix를 확인한다.

[[ 5888 7143 10856 5945]

[ 2414 3159 1781 1551]

[ 7011 7575 8349 5922]

[ 6755 7576 4624 4625]]

The inverse matrix is:

[[-0.0088601 0.01841887 0.0161764 -0.01550075]

[ 0.00328586 -0.00561132 -0.00611487 0.00548777]

[-0.00310154 0.00751513 0.00597322 -0.00618177]

[ 0.01065899 -0.02522337 -0.01958173 0.02004687]]

[[ 1.00000000e+00 1.20813082e-13 6.92397528e-14 -6.37302711e-14]

[-1.36939071e-14 1.00000000e+00 3.48159002e-14 -3.00003078e-14]

[-5.09939313e-14 1.46702095e-13 1.00000000e+00 -1.00981723e-13]

[-5.43627643e-14 1.71938852e-13 1.03406866e-13 1.00000000e+00]]

3.

이분법 함수를 생성하여 이를 적용한다.

The root of (-10,-1) is: -1.3875071099027991

Number of iterations: 30

The root of (-1,0) is: -0.5104293450713158

Number of iterations: 27

The root of (0,10) is: 8.910696404054761

Number of iterations: 30

4.

뉴턴법 함수를 생성하여 문제를 푼다.

The root of x0=-0.1 is: -0.51042934

Number of iterations: 5

The root of x0=-10 is: -1.38750711

Number of iterations: 13

The root of x0=10 is: 8.91069640

Number of iterations: 6

5.

뉴턴법을 이용해 X\_0=0을 대입할 경우, f’(0)이 0이 되므로 런타임에러가 뜬다. 따라서 해를 구하지 못한다.

5.

시컨트법의 공식을 확인하여 함수를 생성하고 이를 적용시킨다.

The root of -10,-9,9 is: -1.387507105582635

Number of iterations: 18

The root of -0.1,-0.2 is: -0.5104293428178256

Number of iterations: 6

The root of 10,9.9 is: 8.910696402960298

Number of iterations: 7

6.

Iteration은 각 문제의 해답에 포함되어 있다.

Homework #2

1.

h = 0.2로 잡고 [-1,1]사이에서의 f(x)를 보간한다. 먼저 f(x)를 정의하고, lagrange\_\_basis의 함수를 정의해 Lagrange\_interpolation을 준비한다. Basis의 return값을 갖고 와 lagrange\_interpolation 본 함수에 적용해 lagrange 보간 함수를 구한다.

결과는 다음과 같다.

라인, 그래프, 도표, 경사이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

2.

이번에는 uniform nodes가 아닌, Chebyshev nodes를 적용해 보간 대상 x를 정의하고, 앞서 정의한 lagrange\_interpolation을 적용한다. 결과는 다음과 같다.

그래프, 라인, 도표, 텍스트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

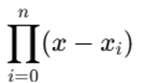
X = 0에서는 uniform node와 달리 오차가 다소 존재하지만, 그 외의 경로에서는 본 함수인 f(x)와 fit함을 확인할 수 있다.

3. 일반 보간 오차 식은 다음과 같다.

폰트, 화이트, 텍스트, 타이포그래피이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

좌항은 어떤 보간 method를 써도 똑같으므로, 오차율을 결정하는 식은



과 같다. 따라서 이 식을 Chebyshev method와 Uniform method를 적용해 보아 Plot 해보면, 다음과 같다.

텍스트, 라인, 그래프, 도표이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

여기서 흥미로운 점은, 시작점인 x=-1에서 Cheby dfference가 Uniform을 압도하는 것을 관찰할 수 있다. 이는 Cheby노드가 양 끝에 더 조밀하게 몰려 있고, -1근처에는 x-x\_i가 거의 0에 가까운 수 여러 개의 곱이 존재하여 product값이 급격히 변동한다고 이유를 들 수 있다. 그러나 시간이 x가 진행될수록 Cheby의 오차율은 Uniform보다 작아지고, Cheby method는 균일 오차 분포에 최적화가 되어있어 구간 끝점에서의 진동을 줄이기 위해 cos함수 기반으로 재배치 되었으므로 product term의 최대값이 최소화가 되어 Uniform method보다 정확함을 알 수 있다.

4.

Spline interpolation은 f\_i(x\_i) = f\_i+1(x\_i)와 f’\_i-1(x\_i)=f’\_i(x\_i)와 f’’\_i-1(x\_i)=f’’\_i(x\_i) 총 4개의 Condition과 f’’(-1)=0, f’’(1)=0인 boundary condition 2개를 합해 총 6개의 식을 정리하여 f\_i(x)=a\_ix^3+b\_ix^2+c\_ix+d\_i의 계수를 구하여 interpolation을 진행한다.

이 계수를 구하는 공식은 다음과 같다.

텍스트, 폰트, 라인, 화이트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

여기서 M은 우리가 구하고자 하는 보간 함수인 S(x)의 두번째 도함수 값이다. 이를 정리하면 삼대각선 행렬이 생기고, boundary condition을 대입해 정리한다. S\_0,S\_1~~~ S\_10까지 보간 함수가 나온 것을 np.concatenate 함수를 사용하여 합쳐서 plot 하면 다음과 같다.

그래프, 라인, 도표, 경사이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

그래프, 라인, 도표, 텍스트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

Cubic Spline의 보간 정확도가 lagrangian의 보간 정확도보다 높다. 이는 Lagrangian은 고차 다항식을 이용해 interpolation해서 불안정하지만, Cubic Spline은 차수가 비교적 낮기 때문에 정밀한 interpolation이 가능하다. 하지만, x=0 부분에서는 Lagrangian이 더 정확한 모습을 보여줬다.

5.

그래프, 라인, 도표이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

Error는 Lagrangian interpolation이 더 작음을 가시적으로 확인할 수 있지만, Spline의 차수가 Lagrangian보다 낮으므로 수치적인 Error는 더 낮을 것임을 알 수 있다.