수치해석과최적화 중간고사

1. 학번과 이름 그리고 별명을 쓰시오.
학번:2015251026
이름:김진우
별명:막학기 21.5

2. (Handwriting Problem) 다음의 단어들의 정의를 쓰고 예시를 만드시오.

(a) Absolute Error and Relative Error

Absolute Error 정의 : 측정값(approximate value)과 실제값(true value)의 차이이다.
(Absolute error = approximate value − true value)
Relative Error 정의: 절대오차(Absolute Error)를 실제값(true value)으로 나눈 값이다.
(Relative error = $\frac{absolute\ \ error}{true\ \ value}$)
예시: 원의 넓이의 실제값(true value)이 πcm^2일 때, 측정값(approximate value)이 $3.14cm^2$가 나왔다면,
Absolute Error = (3.14 - π)cm^2이고, Relative Error = $\frac{3.14-\pi}{\pi}$이다.

(b) Data Error and Computational Error

Data Error 정의 : 올바른 함수에 대해 잘못된 값을 넣었을 때의 결과와 올바른 값을 넣었을 때의 결과의 차이이다.
Computational Error 정의 : 잘못된 함수에 값을 넣었을 때의 결과와 올바른 함수에 값을 넣었을 때의 결과의 차이이다.

예시 : 원의 넓이를 구하는 올바른 함수 f=(반지름)^2 $\times$(원주율)=$r^2 \times \pi$이고, 잘못된 함수 $\hat{f}$=$3.14r^2$이라고 하자.
반지름이 1인 원의 넓이를 구할 때 잘못된 값(1.1)을 넣으면, $data\ \ error = f(1.1) - f(1) = 0.21\pi$이다.
만약 잘못된 함수에 r=1.1을 넣으면, $Computational\ \ Error = \hat{f}(1.1) - f(1.1) = 1.21\pi - 3.7994$이다.

(c) Truncation Error and Rounding Error

Truncation Error 정의 : 정확한 수학식을 근사식으로 나타내면서 생기는 오차이다.
예시 : 무한급수인 테일러 급수의 어느 항(term)이상을 절단(truncation)하여 근사시킨다. 이때 실제값과 근사시킨 값의 차이가 Truncation Error이다.

$$T_f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n = f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2 + \frac{1}{6}f'''(a)(x-a)^3 + \cdots$$

Rounding Error 정의 : 컴퓨터에서 수치계산 시, 컴퓨터가 계산할 수 있는 자릿수의 제한으로 인해 생긴 오류로 올림, 버림, 반올림할 때 발생한다.
예시 : π=3.141592...를 소수 둘째자리까지 반올림하면 $\pi \approx 3.14$이다. 실제 π와 반올림 한 값인 3.14 사이의 차이가 Rounding Error이다.

(d) Forward Error and Backward Error

Forward Error : 함수 $y=f(x)$에 대해 값을 넣어 나온 값($\hat{y}$)과 실제 값(y)의 차이이다.

$(Forward\ \ Error = \hat{y} - y)$

Backward Error : 함수 $y=f(x)$에 대해 실제로 넣은 값($\hat{x}$)과 정확한 값(x)의 차이이다.

$(Backward\ \ Error = \hat{x} - x, \quad f(\hat{x}) = \hat{y})$

예시 : 올바른 함수 $f=\sqrt{x}$ 이고, 잘못된 함수 $\hat{f}=\sqrt{x+3}$ 이라고 하자.

x=1일 때, $Forward\ \ Error = \hat{y} - y = 2 - 1 = 1$이고,

$Backward\ \ Error = \hat{x} - x = 4 - 1 = 3$이다.($\hat{y} = 2 = \sqrt{4} = f(\hat{x})$)

(e) 'well-posed' and 'ill-posed'

well-posed와 ill-posed 정의 : 해가 유일하게 존재하고, 안정적(문제 데이터에 지속적으로 의존)이라면 well-posed라고 하고, well-posed의 반대 개념을 ill-posed라고 한다.

예시 : $\int_0^1 x\,dx$ 의 값은 1/2로 유일하게 존재하고 안정적이므로 well-posed이다.

$\int_0^1 f(x)\,dx = 1/2$를 만족하는 $f(x)$는 $f(x)=x$ 또는 $f(x)=1/2$ 등등 다양하게(유일하지 않게) 존재하므로 ill-posed이다.

(f) 'well-conditioned' and 'ill-conditioned'

well-conditioned 정의 : 입력의 상대적 변화가 해의 상대적 변화와 유사하게 일어나면 well-conditioned라고 한다.
ill-conditioned 정의 : 입력의 상대적 변화보다 해의 상대적 변화가 상당히 크면 ill-conditioned라고 한다.

구분하는 방법 : $cond = \dfrac{|(f(\hat{x}) - f(x))/f(x)|}{|(\hat{x} - x)/x|} = \dfrac{|TRIANGLEy/y|}{|TRIANGLEx/x|}$

Condition number를 통해 'well-conditioned'과 'ill-conditioned'을 구분할 수 있는데
Condition number 1보다 작으면 'well-conditioned', 크면 'ill-conditioned''이다.

예시 : 함수 $f(x) = \sqrt{x}$ 의 $x=4$이고 $\hat{x}$=4.41일 때, $cond = \dfrac{|0.1/2|}{|0.41/4|} \approx 0.48$이므로 well-conditioned,

함수 $g(x) = x^2$의 x=2이고 $\hat{x}$=2.1일 때, $cond = \dfrac{0.41/4}{0.1/2} = 2.05$이므로 ill-conditioned이다.

3. (Handwriting Problem) 선형방정식 Ax = b가 유일한 한 개의 근을 갖기 위한 조건을 span이라는 개념을 사용하여 설명하시오.

'span'이란, 해당 집합의 벡터들의 linear combinnation($\sum_i c_i v^{(i)}$)으로 얻을 수 있는 모든 점들의 집합을 말한다. ($v^{(i)}$는 벡터 집합으로 A의 각 열, c_i는 x의 각 성분)
R^m의 모든 점을 포함하기 위해 최소 m개의 독립인 span들이 존재하면, Ax=b가 유일한 한 개의 근을 갖는다.

4. (Handwriting Problem) Gaussian Elimination을 사용하여 선형방정식을 풀 때, pivoting을 하는 이유를 설명하시오. [5]

예를 들어, A= $\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix}$ (ε는 0에 근사하는 양수)일 때, pivoting 없이 Gaussian Elimination하면

U= $\begin{bmatrix} \varepsilon & 1 \\ 0 & 1-\frac{1}{\varepsilon} \end{bmatrix}$ = $\begin{bmatrix} \varepsilon & 1 \\ 0 & -\frac{1}{\varepsilon} \end{bmatrix}$ (pivot이 작으면 큰 multiplier가 되기 때문), L= $\begin{bmatrix} 1 & 0 \\ \frac{1}{\varepsilon} & 1 \end{bmatrix}$ 이다.

$LU = \begin{bmatrix} 1 & 0 \\ \frac{1}{\varepsilon} & 1 \end{bmatrix} \begin{bmatrix} \varepsilon & 1 \\ 0 & -\frac{1}{\varepsilon} \end{bmatrix} = \begin{bmatrix} \varepsilon & 1 \\ 1 & 0 \end{bmatrix} \neq A$ 이 되어 문제가 발생한다.

따라서 위의 예시처럼 leading diagonal가 0일 때, pivoting하면 문제없이 해를 구할 수 있다.

5. (Handwriting Problem) Interpolation을 사용하는 이유 5가지를 쓰시오. [5]

① Plotting smooth curve through discrete data points
② Reading between lines of table
③ Differentiating or integrating tabular data
④ Quick and easy evaluation of mathematical function
⑤ Replacing complicated function by simple one

6. (Handwriting Problem) O/X 문제 (맞으면 +2점, 틀리면 -2점, 몰라서 ?표기하면 0점)

(a) 주어진 data point를 interpolation하기 위해서는 polynomial만 사용해야 한다. (X)

(b) Interpolating 함수와 주어진 data 값이 정확하게 일치한다면, basis 함수의 linear combinnation에서 계수가 잘 결정되었음을 의미한다. (O)

(c) Monomial Basis, Lagrange Basis, Newton Basis를 사용한 interpolation의 결과는 다르다. (X)

7.

명령 창

```
>> A=[1 2 3;1 0 -2;4 1 -1];
>> b=[1;2;3];
>> [x, LU]=HW8_prob7(A, b)

x =

     0
     2
    -1


LU =

     1     2     3
     1     0    -2
     4     1    -1

fx >>
```

편집기 - C:\Users\k\Desktop\자료\4학년\2학기\수치해석과 최적화\MATLAB\중간\H...

HW8_prob7.m

```matlab
%% Gauss elimination
% MA(=U), Mb
function [x, LU]=HW8_prob7(A, b)
    [A_row, A_col]=size(A);
    [b_row , ~]=size(b);
    n=length(A);
    if A_row ~= A_col
        error('행렬A의 크기가 n x n 이 아닙니다.')
    end

    if b_row ~= A_col
        error('행렬A의 크기와 벡터 b의 크기가 일치하지 않습니다.')
    end

    for k = 1:(n-1)
        if A(k,k) == 0
            break;
        end
        M_{k}=eye(n);
        for i = k+1: n
            M_{k}(i,k) = -A(i,k)/A(k,k);
        end
        A=M_{k}*A;
    end
    U=A;
    M=eye(n);
    for k=1:(n-1)
        M=M_{k}*M;
    end
    L=inv(M);
    LU=L*U;
    c=forward_substitution(L,b,n);
    x=back_substitution(U,c,n);
end

function x=forward_substitution(A,b,n)

    for j = 1:n
        if A(j,j) == 0
            break;
        end
        x(j,1) = b(j,1)/A(j,j);
        for i=j+1:n
```

작업 공간

이름	값
A	[1,2,3;1,0,-2;4,1,-1]
b	[1;2;3]
LU	[1,2,3;1,0,-2;4,1,-1]
x	[0;2;-1]

8.

명령 창

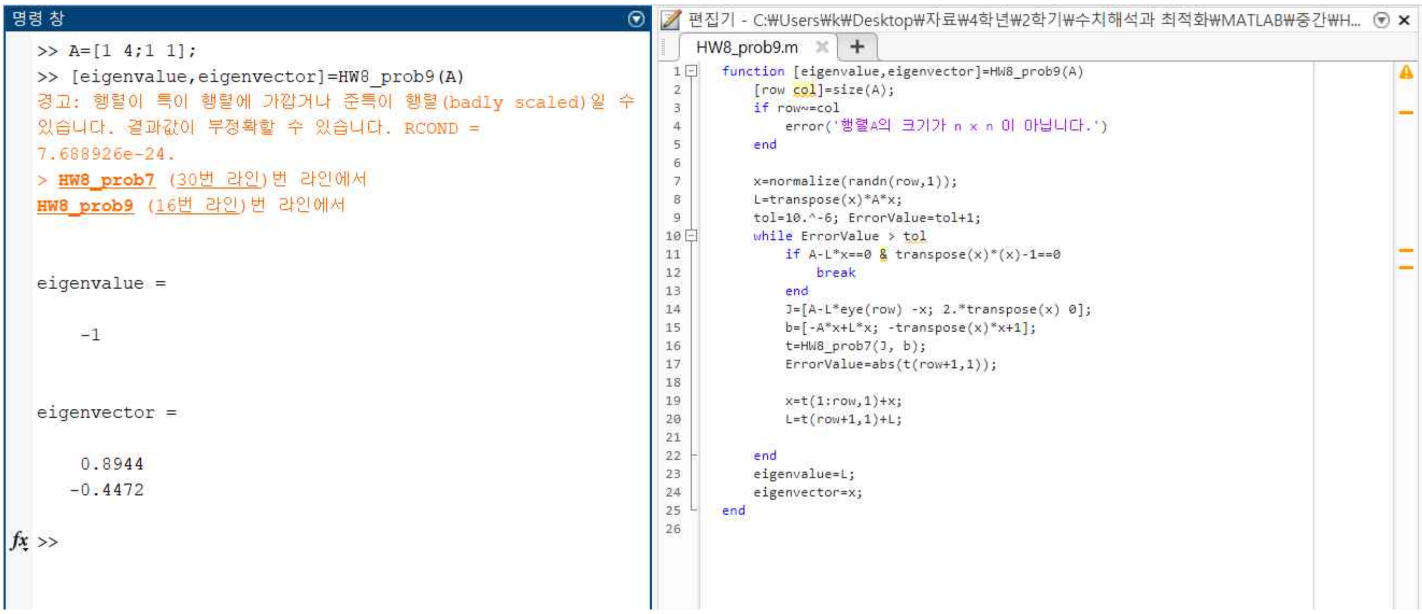
```
>> A=[1 2 2;4 4 2;4 6 4];
>> b=[3;6;10];
>> [x, LU]=HW8_prob7(A, b)

x =

    -1
     3
    -1


LU =

     1     2     2
     4     4     2
     4     6     4

>> x=HW8_prob8(LU,x)

x =

    0.5000
    1.2500
   -2.0000
```

편집기 - C:\Users\k\Desktop\자료\4학년\2학기\수치해석과 최적화\MATLAB\중간\H...

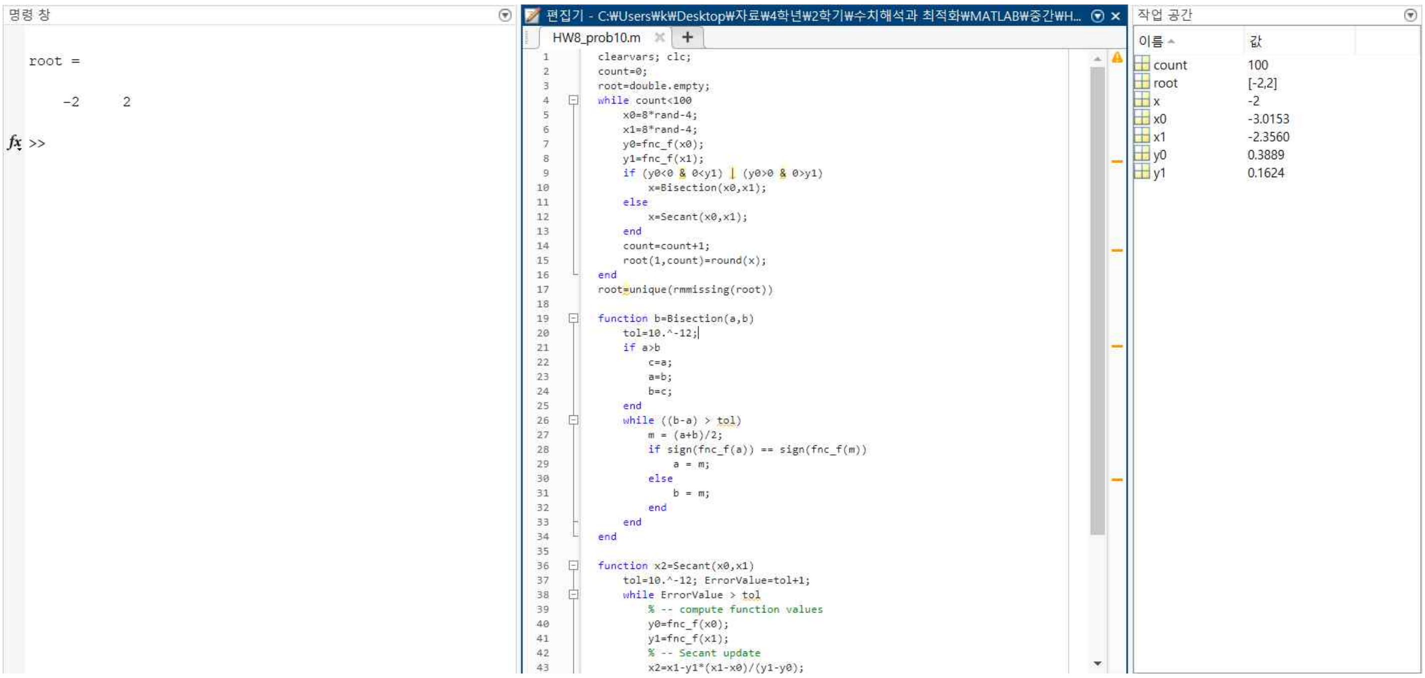
HW8_prob8.m

```matlab
function x=HW8_prob8(A,b)
    u=[0; 0; 1];
    v=[0; 2; 0];
    x=inv(A-u*v.')*b;
end
```

9.


명령 창
>> A=[1 4;1 1];
>> [eigenvalue,eigenvector]=HW8_prob9(A)
경고: 행렬이 특이 행렬에 가깝거나 준특이 행렬(badly scaled)일 수 있습니다. 결과값이 부정확할 수 있습니다. RCOND = 7.688926e-24.
> HW8_prob7 (30번 라인)번 라인에서
HW8_prob9 (16번 라인)번 라인에서
eigenvalue =
-1
eigenvector =
0.8944
-0.4472
fx >>
편집기 - C:₩Users₩k₩Desktop₩자료₩4학년₩2학기₩수치해석과 최적화₩MATLAB₩중간₩H...
HW8_prob9.m
function [eigenvalue,eigenvector]=HW8_prob9(A)
[row col]=size(A);
if row~=col
error('행렬A의 크기가 n x n 이 아닙니다.')
end
x=normalize(randn(row,1));
L=transpose(x)*A*x;
tol=10.^-6; ErrorValue=tol+1;
while ErrorValue > tol
if A-L*x==0 & transpose(x)*(x)-1==0
break
end
J=[A-L*eye(row) -x; 2.*transpose(x) 0];
b=[-A*x+L*x; -transpose(x)*x+1];
t=HW8_prob7(J, b);
ErrorValue=abs(t(row+1,1));
x=t(1:row,1)+x;
L=t(row+1,1)+L;
end
eigenvalue=L;
eigenvector=x;
end

10.


명령 창
root =
-2 2
fx >>
편집기 - C:₩Users₩k₩Desktop₩자료₩4학년₩2학기₩수치해석과 최적화₩MATLAB₩중간₩H...
HW8_prob10.m
clearvars; clc;
count=0;
root=double.empty;
while count<100
x0=8*rand-4;
x1=8*rand-4;
y0=fnc_f(x0);
y1=fnc_f(x1);
if (y0<0 & 0<y1) | (y0>0 & 0>y1)
x=Bisection(x0,x1);
else
x=Secant(x0,x1);
end
count=count+1;
root(1,count)=round(x);
end
root=unique(rmmissing(root))
function b=Bisection(a,b)
tol=10.^-12;
if a>b
c=a;
a=b;
b=c;
end
while ((b-a) > tol)
m = (a+b)/2;
if sign(fnc_f(a)) == sign(fnc_f(m))
a = m;
else
b = m;
end
end
end
function x2=Secant(x0,x1)
tol=10.^-12; ErrorValue=tol+1;
while ErrorValue > tol
% -- compute function values
y0=fnc_f(x0);
y1=fnc_f(x1);
% -- Secant update
x2=x1-y1*(x1-x0)/(y1-y0);
작업 공간
이름
값
count 100
root [-2,2]
x -2
x0 -3.0153
x1 -2.3560
y0 0.3889
y1 0.1624

11.

명령 창

```
convergence rate of newton method : 2
>>
```

편집기 - C:₩Users₩k₩Desktop₩자료₩4학년₩2학기₩수치해석과 최적화₩MATLAB₩중간₩H...

HW8_prob11.m

```matlab
clearvars; clc;
x=2;
r=0;
tol=10.^-6; ErrorValue=tol+1;
while ErrorValue > tol
    % compute function values
    [y,yp]=fnc_newton_f(x);
    % Newton update
    newx=x-y/yp;
    % compute error
    ErrorValue=abs(newx-x);
    r=r+y/yp;
    % update x
    x=newx;
end
fprintf('convergence rate of newton method : %d\n',r)

function [y,yp]=fnc_newton_f(x)
y=exp(x)-1;
yp=exp(x);
end
```

작업 공간

이름	값
ErrorValue	2.9911e-11
newx	1.0660e-16
r	2
tol	1.0000e-06
x	1.0660e-16
y	2.9911e-11
yp	1.0000