200020059

HIMANSHU CHOUDHARY

CL249: ASSIGNMENT 4

PROBLEM

We are given matrices A and B of size 15x15 and 15x1 respectively. We have to solve this equation Ax=b using Jacobi and Gauss Siedel Iteration Techniques, and have to compare the number of operations of Jacobi, Gauss-Siedel and Gauss Elimination.

Description of Method

- Jacobi Iteration Method,

In Jacobi method, we have $x^{(0)}$ = initial guess

$$Ax = B$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n1} & & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

that is

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$

$$x_1^{(1)} = \frac{b_1 - \sum_{j=2}^{n} a_{1j}x_j^{(0)}}{a_{11}}$$

In general,

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1,\, j\neq i}^{n} a_{ij}x_j^{(k)}}{a_{ii}}$$

we basically update values of $x^{(k+1)}$ ($k+1^{th}$ iteration) using k^{th} iteration.

- Gauss Siedel

we have $Ax = b$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & & & \vdots \\ \vdots & & \ddots & \\ a_{n1} & \cdots & & a_{nn} \end{bmatrix} \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ \vdots \\ x_n^{(0)} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$a_{11} x_1^{(0)} + a_{12}(x_2^{(0)}) + \cdots + a_{1n} x_n^{(0)} = b_1$$

$$x_1^{(1)} = \frac{b_1 - \sum_{j=2}^{n} a_{ij} x_j^{(0)}}{a_{11}}$$

$\vdots$

$$x_k^{(1)} = \frac{b_1 - \sum_{j=k+1}^{n} a_{ij} x_j^{(0)} - \sum_{j=1}^{k-1} a_{ij} x_j^{(1)}}{a_{ii}}$$

in general

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}}{a_{ii}}$$

Converging condition

$$\left| \frac{X_f - X_i}{X_f} \right| < \text{tolerance}$$

M T W T F S S
Page No.:
Date:
YOUVA

PSEUDO-CODE

jacobi.m

declare X_g and X_f and err
iteration loop {
  loop through rows of A
    $X_f(i) = B(i) - \text{sum}\sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j$

    add operations
  $X_i = X_f$
iterate through rows of X_f
  if any element $- X_i$ < error
    continue
  else break

main file just declare matrices and uses these functions.

gauss_seidel.m

declare X_g and X_f
iterating loop
  iterate through A rows (i) {
  iterate through 1 to i-1
    sum += $A_{ij} X_f(j)$
  iterate through i+1, n
    sum += $A_{ij} X_g(j)$
  $X_f(i) = \dfrac{B(i) - \text{sum}}{A_{ii}}$

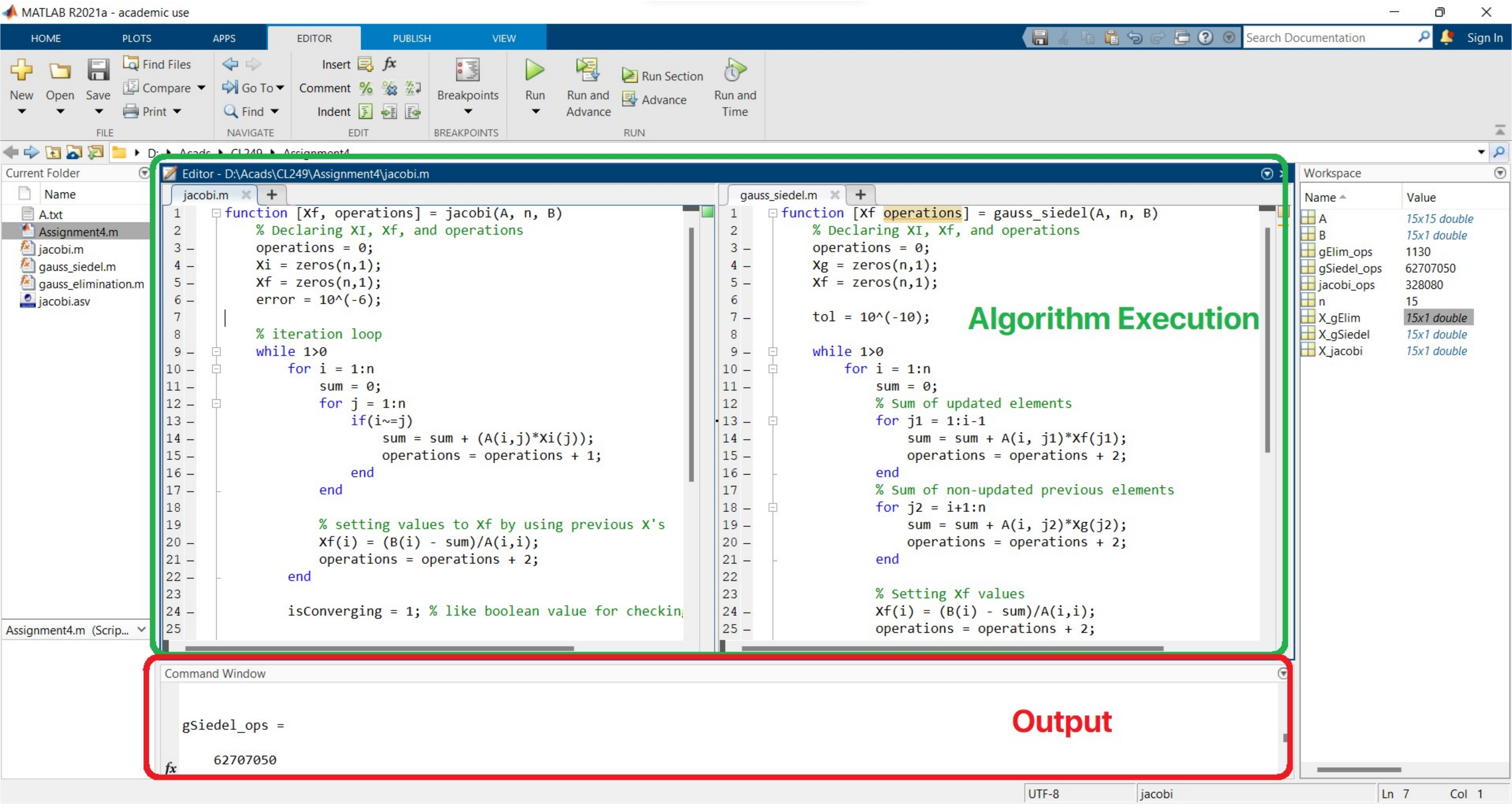
iterate through X_f rows
} if $\dfrac{X_f(i) - X_i(i)}{X_f}$ < tol
    continue
  else break

```
A = load('A.txt');
n = length(A);
B = zeros(n, 1) + 61;

[X_jacobi jacobi_ops] = jacobi(A, n, B);  % Solving through Jacobi method
[X_gSiedel gSiedel_ops] = gauss_siedel(A, n, B);  % Solving through Gauss Siedel ↙
method
[X_gElim gElim_ops] = gauss_elimination(A, B, n, n);  % Solving through Gauss ↙
Elimination

X_jacobi
jacobi_ops

X_gSiedel
gSiedel_ops

X_gElim
gElim_ops
```

```
function [Xf, operations] = jacobi(A, n, B)
    % Declaring XI, Xf, and operations
    operations = 0;
    Xi = zeros(n,1);
    Xf = zeros(n,1);
    error = 10^(-6);

    % iteration loop
    while 1>0
        for i = 1:n
            sum = 0;
            for j = 1:n
                if(i~=j)
                    sum = sum + (A(i,j)*Xi(j));
                    operations = operations + 1;
                end
            end

            % setting values to Xf by using previous X's
            Xf(i) = (B(i) - sum)/A(i,i);
            operations = operations + 2;
        end

        isConverging = 1; % like boolean value for checking convergence

        % checking convergence for every element
        for k = 1:n
            if abs((Xf(k) - Xi(k))/Xf(k)) > error
                isConverging = 0;
            end
        end

        if(isConverging == 1) % if converging, then break
            break;
        end
        Xi = Xf;
    end
    return
end
```

```
function [Xf operations] = gauss_siedel(A, n, B)
    % Declaring XI, Xf, and operations
    operations = 0;
    Xg = zeros(n,1);
    Xf = zeros(n,1);

    tol = 10^(-10);

    while 1>0
        for i = 1:n
            sum = 0;
            % Sum of updated elements
            for j1 = 1:i-1
                sum = sum + A(i, j1)*Xf(j1);
                operations = operations + 2;
            end
            % Sum of non-updated previous elements
            for j2 = i+1:n
                sum = sum + A(i, j2)*Xg(j2);
                operations = operations + 2;
            end

            % Setting Xf values
            Xf(i) = (B(i) - sum)/A(i,i);
            operations = operations + 2;
        end

        isConverging = 1; % like boolean value for checking convergence

        % checking convergence for every element
        for k = 1:n
            if abs((Xf(k) - Xg(k))/Xf(k)) > tol
                isConverging = 0;
            end
        end

        if(isConverging == 1) % break if every element is converging
            break;
        end

        Xg = Xf;
    end
    return
end
```

```matlab
function [Xf operations] = gauss_siedel(A, n, B)
    % Declaring XI, Xf, and operations
    operations = 0;
    Xg = zeros(n,1);
    Xf = zeros(n,1);

    tol = 10^(-10);

    while 1>0
        for i = 1:n
            sum = 0;
            % Sum of updated elements
            for j1 = 1:i-1
                sum = sum + A(i, j1)*Xf(j1);
                operations = operations + 2;
            end
            % Sum of non-updated previous elements
            for j2 = i+1:n
                sum = sum + A(i, j2)*Xg(j2);
                operations = operations + 2;
            end

            % Setting Xf values
            Xf(i) = (B(i) - sum)/A(i,i);
            operations = operations + 2;
        end

        isConverging = 1; % like boolean value for checking convergence

        % checking convergence for every element
        for k = 1:n
            if abs((Xf(k) - Xg(k))/Xf(k)) > tol
                isConverging = 0;
            end
        end

        if(isConverging == 1) % break if every element is converging
            break;
        end

        Xg = Xf;
    end
    return
end
```


MATLAB R2021a - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
Current Folder
Name
A.txt
Assignment4.m
jacobi.m
gauss_siedel.m
gauss_elimination.m
jacobi.asv
Assignment4.m (Scrip...
Editor - D:\Acads\CL249\Assignment4\jacobi.m
jacobi.m
function [Xf, operations] = jacobi(A, n, B)
% Declaring XI, Xf, and operations
operations = 0;
Xi = zeros(n,1);
Xf = zeros(n,1);
error = 10^(-6);
% iteration loop
while 1>0
for i = 1:n
sum = 0;
for j = 1:n
if(i~=j)
sum = sum + (A(i,j)*Xi(j));
operations = operations + 1;
end
end
% setting values to Xf by using previous X's
Xf(i) = (B(i) - sum)/A(i,i);
operations = operations + 2;
end
isConverging = 1; % like boolean value for checkin
gauss_siedel.m
function [Xf operations] = gauss_siedel(A, n, B)
% Declaring XI, Xf, and operations
operations = 0;
Xg = zeros(n,1);
Xf = zeros(n,1);
tol = 10^(-10);
while 1>0
for i = 1:n
sum = 0;
% Sum of updated elements
for j1 = 1:i-1
sum = sum + A(i, j1)*Xf(j1);
operations = operations + 2;
end
% Sum of non-updated previous elements
for j2 = i+1:n
sum = sum + A(i, j2)*Xg(j2);
operations = operations + 2;
end
% Setting Xf values
Xf(i) = (B(i) - sum)/A(i,i);
operations = operations + 2;
Algorithm Execution
Workspace
Name
Value
A
15x15 double
B
15x1 double
gElim_ops
1130
gSiedel_ops
62707050
jacobi_ops
328080
n
15
X_gElim
15x1 double
X_gSiedel
15x1 double
X_jacobi
15x1 double
Command Window
gSiedel_ops =
62707050
Output
UTF-8
jacobi
Ln 7
Col 1

```
>> Assignment4

X_jacobi =

   NaN
   NaN
   NaN
   NaN
   NaN
   NaN
   NaN
   NaN
   NaN
   NaN
   NaN
  -Inf
   NaN
   NaN
   NaN


jacobi_ops =

      328080


X_gSiedel =

   1.0e+05 *

    0.0183
    0.1006
    0.2385
    0.4239
    0.6496
    0.9089
    1.1956
    1.5043
    1.8300
    2.1685
    2.5162
    2.8700
    3.2275
    3.5868
    3.9467


gSiedel_ops =

    62707050


X_gElim =

   1.0e+05 *
```

```
    0.0183
    0.1006
    0.2385
    0.4239
    0.6496
    0.9089
    1.1956
    1.5043
    1.8300
    2.1685
    2.5162
    2.8700
    3.2275
    3.5868
    3.9467


gElim_ops =

        1130

>>
```

Page No.:
Date:

YOUVA

Answer

Operations from Jacobi = 328080
But vector not converging to a real value

Operations from Gauss Siedel : 62707050
Converged to a value

Operations from Gauss Elimination : 1130

Gauss elimination is working best for this given matrix