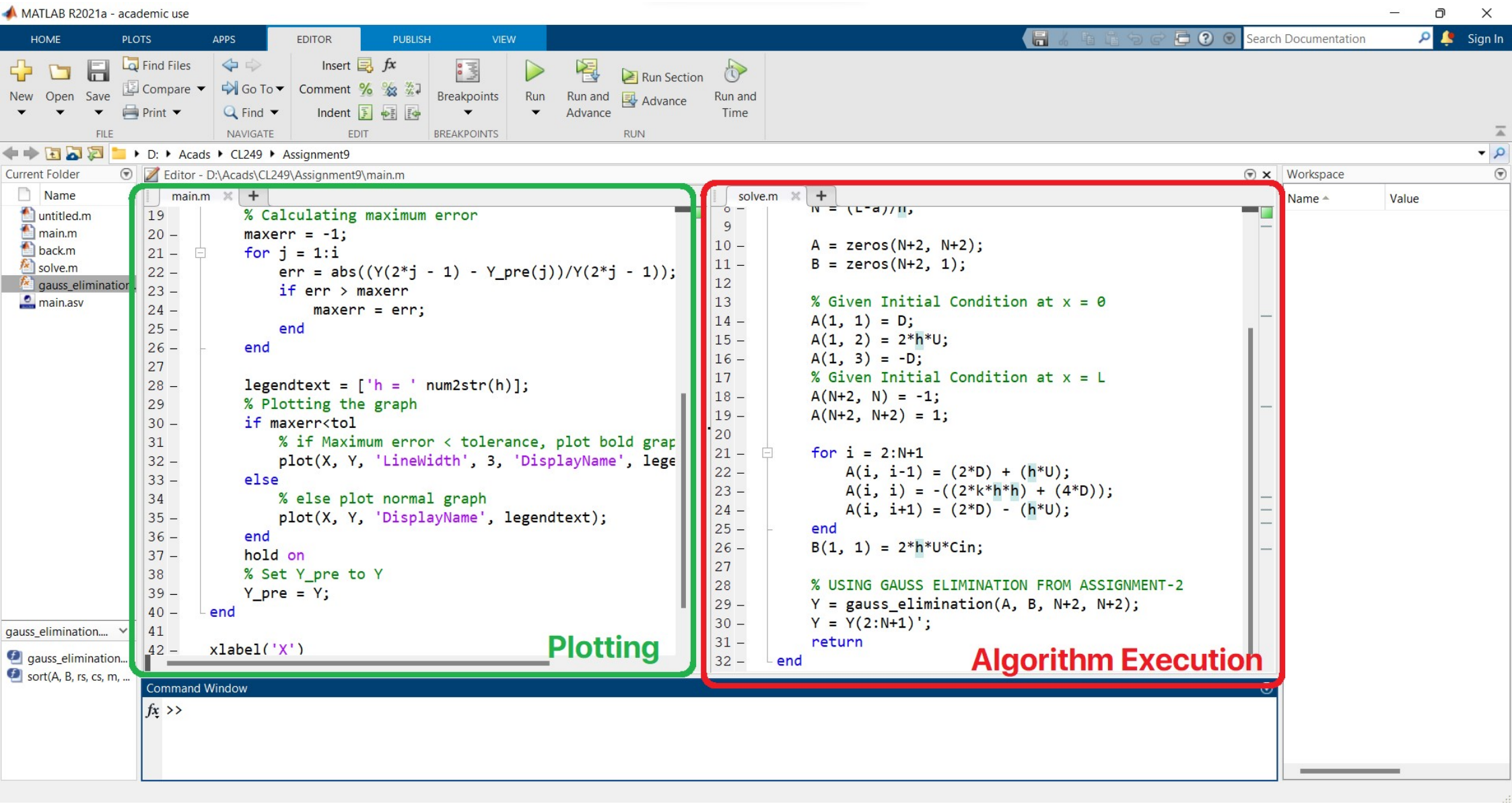
MATLAB R2021a - academic use
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
D: ▸ Acads ▸ CL249 ▸ Assignment9
Current Folder
Name
untitled.m
main.m
back.m
solve.m
gauss_elimination...
main.asv
Editor - D:\Acads\CL249\Assignment9\main.m
main.m
% Calculating maximum error
maxerr = -1;
for j = 1:i
err = abs((Y(2*j - 1) - Y_pre(j))/Y(2*j - 1));
if err > maxerr
maxerr = err;
end
end
legendtext = ['h = ' num2str(h)];
% Plotting the graph
if maxerr<tol
% if Maximum error < tolerance, plot bold grap
plot(X, Y, 'LineWidth', 3, 'DisplayName', lege
else
% else plot normal graph
plot(X, Y, 'DisplayName', legendtext);
end
hold on
% Set Y_pre to Y
Y_pre = Y;
end
xlabel('X')
Plotting
solve.m
A = zeros(N+2, N+2);
B = zeros(N+2, 1);
% Given Initial Condition at x = 0
A(1, 1) = D;
A(1, 2) = 2*h*U;
A(1, 3) = -D;
% Given Initial Condition at x = L
A(N+2, N) = -1;
A(N+2, N+2) = 1;
for i = 2:N+1
A(i, i-1) = (2*D) + (h*U);
A(i, i) = -((2*k*h*h) + (4*D));
A(i, i+1) = (2*D) - (h*U);
end
B(1, 1) = 2*h*U*Cin;
% USING GAUSS ELIMINATION FROM ASSIGNMENT-2
Y = gauss_elimination(A, B, N+2, N+2);
Y = Y(2:N+1)';
return
end
Algorithm Execution
Workspace
Name
Value
Command Window
>>
gauss_elimination....
gauss_elimination...
sort(A, B, rs, cs, m, ...