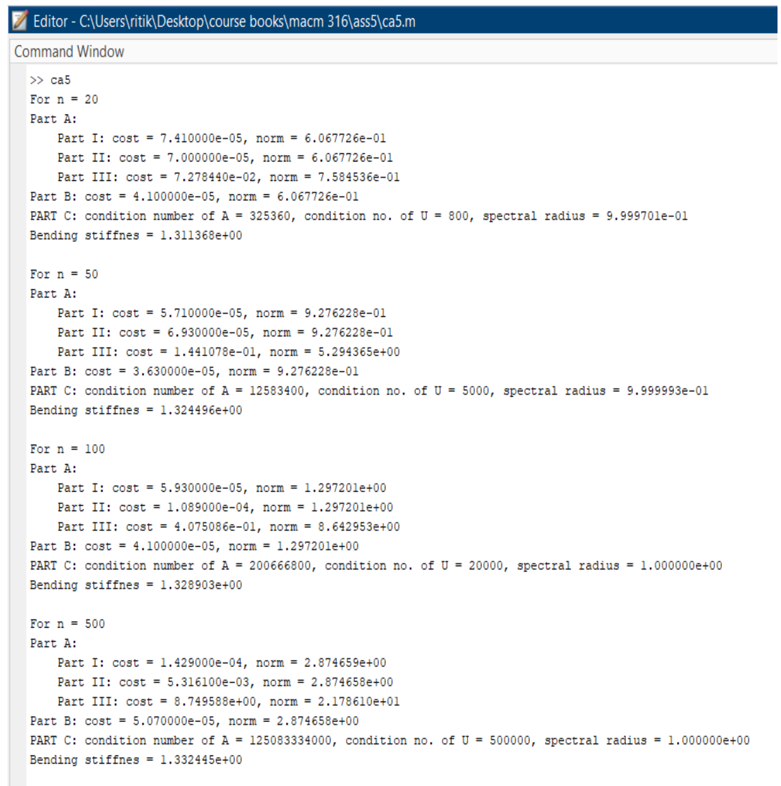
RITIKA GOYAL (301401516)

MACM316 CA5

The goal of this assignment is to calculate the deflection of cantilever beam by solving system of linear equations Az=b by using different methods, where A is a sparse matrix, b is the given load force acting along the beam ($b = F.h^4$) and z is the corresponding vertical deflections. The beam is divided into n equal sections of length h=L/n, defined by xi = ih where i = 1,2,…,n. The solution is then further used to calculate bending stiffness of the beam by extracting the maximum value of deflection.

Part A) The linear system Az = b is solved by using three methods- GE+PP algorithm for sparse linear system, GE+PP algorithm for dense linear systems, Gauss-Seidel iterative algorithm (tol = 1e-8, maxiter = 1e5). The solution with n=20,50,100 and 500 is computed and their cost and norms are displayed which are shown in the given figure on right. The cost and norm of first and second method is nearly same but they are large in the third method because in third method, gs2 function is used which has more operations as in part 1 and 2, only backslash operator of MATLAB is used. It is observed that first method is most accurate because of the nature of backslash operator in MATLAB which first identifies the matrix and then performs the optimal calculations depending on sparse and full nature of matrix. In method 1, the sparse matrix is used which gives more accurate result.

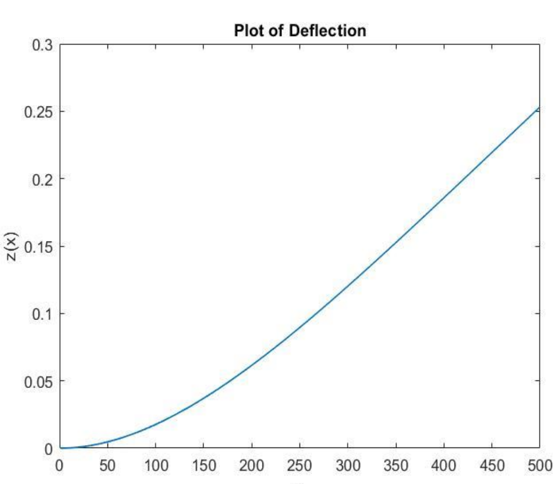


```
Editor - C:\Users\ritik\Desktop\course books\macm 316\ass5\ca5.m
Command Window
>> ca5
For n = 20
Part A:
    Part I: cost = 7.410000e-05, norm = 6.067726e-01
    Part II: cost = 7.000000e-05, norm = 6.067726e-01
    Part III: cost = 7.278440e-02, norm = 7.584536e-01
Part B: cost = 4.100000e-05, norm = 6.067726e-01
PART C: condition number of A = 325360, condition no. of U = 800, spectral radius = 9.999701e-01
Bending stiffnes = 1.311368e+00

For n = 50
Part A:
    Part I: cost = 5.710000e-05, norm = 9.276228e-01
    Part II: cost = 6.930000e-05, norm = 9.276228e-01
    Part III: cost = 1.441078e-01, norm = 5.294365e+00
Part B: cost = 3.630000e-05, norm = 9.276228e-01
PART C: condition number of A = 12583400, condition no. of U = 5000, spectral radius = 9.999993e-01
Bending stiffnes = 1.324496e+00

For n = 100
Part A:
    Part I: cost = 5.930000e-05, norm = 1.297201e+00
    Part II: cost = 1.089000e-04, norm = 1.297201e+00
    Part III: cost = 4.075086e-01, norm = 8.642953e+00
Part B: cost = 4.100000e-05, norm = 1.297201e+00
PART C: condition number of A = 200666800, condition no. of U = 20000, spectral radius = 1.000000e+00
Bending stiffnes = 1.328903e+00

For n = 500
Part A:
    Part I: cost = 1.429000e-04, norm = 2.874659e+00
    Part II: cost = 5.316100e-03, norm = 2.874658e+00
    Part III: cost = 8.749588e+00, norm = 2.178610e+01
Part B: cost = 5.070000e-05, norm = 2.874658e+00
PART C: condition number of A = 125083334000, condition no. of U = 500000, spectral radius = 1.000000e+00
Bending stiffnes = 1.332445e+00
```

As n increases, it is clearly observed that both cost and 2-norm of the matrix increases while using Gauss-Seidel methods. The cost increases due to more operations involved during calculation and norm increases because of the iterative nature of the algorithm which only yields approximate solution but not exact solution and the subtractive cancellation error is involved because non diagonal entries are shifted to another side to calculate solution.

Part B) The solution is calculated by taking another sparse matrix U and its transpose whose multiplication results A itself i.e. $A = UU^T$. The backslash command is used to solve two triangular solves. The norm of the resulting solution is similar to method 1 and 2 from part A as backslash is used in these three algorithms. The cost of this algorithm is less than method 1 as the forward or backward substitution is used which contributes towards less cost.

Part C) The condition number of A and U is computed. As n increases, the condition number of both A and U also increases because of the more ill conditioned (close to singular) nature of A and U matrices as the condition number of large sparse matrix can only be estimated. The spectral radius of Gauss-seidel iteration matrix T is also calculated which is close to 1 and thus will converge slower.



Part D) The solution in part b is used to plot the deflection z(x) versus x (shown below) because in part b, the lower and upper triangular matrices are used and backslash just performs substitution which reduces the cost of calculation. The slope of the plot is same for every n used. The maximum value of solution is extracted and used to calculate the bending stiffness (EI) of beam by using: $EI=FL^4/6z(L)$. Since EI (bending stiffness) is the material property, it is independent of the size of n and is approximately same with increasing n.

ALL THE RESULTS ARE DISPLAYED IN THE ABOVE IMAGE.

RITIKA GOYAL (301401516)

MATLAB CODE

```
function ca5()
nlist = [20,50,100,500];
%nlist = [10];
for n = nlist
fprintf("For n = %d\n",n);
e = ones(n,1);
Asparse = spdiags([e -4*e 6*e -4*e e],-2:2,n,n);
Asparse(1,1)= 9; Asparse(n,n-1) = -2; Asparse(n-1,n)=-2; Asparse(n-1,n-1)=5; Asparse(n,n)=1;
F = 0.4; L=  1.5; bi = F* (L/n)^4;
b = bi * ones(n,1); %fprintf("%d ",b);

%Part A)
%part I)
tic
x_asparse = Asparse \ b;
x_cost = toc;
x_norm = norm(x_asparse,2);
fprintf("Part A:\n\tPart I: cost = %d, norm = %d\n",x_cost , x_norm); %fprintf("x =  %d \n",x);

%part II)
Adense = full(Asparse);
tic
x_dense = Adense \ b;
x_dense_cost = toc;
x_dense_norm = norm(x_dense,2);
fprintf("\tPart II: cost = %d, norm = %d\n",x_dense_cost, x_dense_norm);%fprintf("x =  %d
\n",x_dense);

%part III)
z0 = ones(n,1);
tic
x_gs2=gs2(Asparse,b,z0,1e-8,1e5);
x_gs2_cost = toc;
x_gs2_norm= norm(x_gs2,2);
fprintf("\tPart III: cost = %d, norm = %d\n",x_gs2_cost, x_gs2_norm); %fprintf("x =  %d
\n",x_gs2);

%Part b)
U = spdiags([e -2*e e],0:2,n,n);
U(1,1) = 2;
U_transpose = U.';
tic
x_double_back = (U_transpose\(U\b));
x_double_bs_cost = toc;
x_double_bs_norm = norm(x_double_back,2);
fprintf("Part B: cost = %d, norm = %d\n",x_double_bs_cost, x_double_bs_norm); %fprintf("x =  %d
\n",x_double_back);

%Part c)
U_gs2  = triu(Asparse, 1);  % upper triangular part of A
DpL= Asparse - U_gs2;       % =D+L
T  = -DpL \ U_gs2;    % iteration matrix
condest_Asparse = condest(Asparse);
condest_U = condest(U);
spectral_radiusT = max(abs(eig(full(T))));
fprintf("PART C: condition number of A = %d, condition no. of U = %d, spectral radius =
%d\n",condest_Asparse,condest_U,spectral_radiusT);

%Part d)
x = 1:1:n;
plot(x,x_double_back,"LineWidth",1);xlabel("x");ylabel("z(x)"); title("Plot of Deflection");
bending_stiffness = (F * (L)^4)/(6*max(x_double_back));
fprintf("Bending stiffnes = %d\n\n",bending_stiffness);
end
end
```