**MACM 316 – Assignment 5**

a)

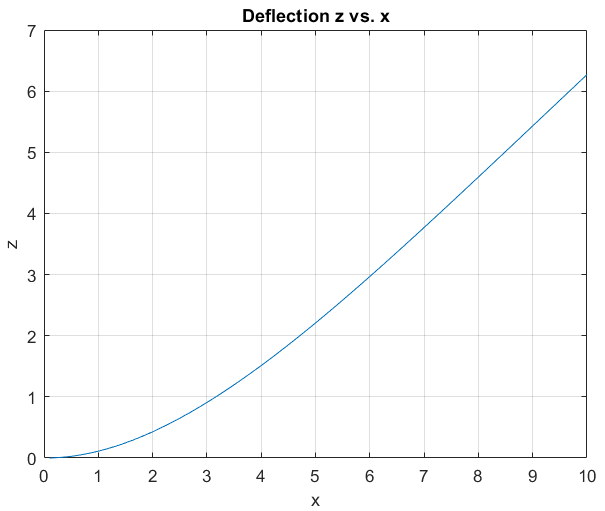
I) For n=100: this method takes roughly 0.004 seconds to compute the answer. For n=1000: this method takes roughly 0.005 seconds to compute the answer.

II) For n=100: this method takes roughly 0.004 seconds to compute the answer. For n=1000: this method takes roughly 0.022 seconds to compute the answer.

III) For n=100: this method takes roughly 0.46 seconds to compute the answer. For n=1000: this method takes roughly 91.3 seconds to compute the answer.

For n=100: I) yielded a norm of 32.097, II) yielded 32.097, and III) yielded 8.7871. For n=1000: I) yielded 100.2668, II) yielded 100.2670, and III) yielded 31.2192. The quickest would be method I) and judging from the norms, this method has a similar answer to II), which is just a bit slower. I) and II) are both good in terms of accuracy. I) is faster because it is calculating the matrix as a sparse matrix, saving time on the zeroes, whereas II) is dense. III) takes very long to compute, and the nature of Gauss-Seidel generates large fractions very quickly as iterations go on, resulting in large errors, hence the large difference in norms between this method and I) and II).

b) n=100: this method takes roughly 0.0056 seconds and has virtually the same 2-norm as I), which is 32.097, so this method is quite accurate. n=1000: the method takes about the same time and has the 2-norm 100.2699, which is nearly the same as part I), just off by ~10-4. From this information, we can conclude this method is accurate and about as quick as part I. p(t)<1 in this case, but there could be error present in this calculation, so we are not completely sure if it will converge.

c) n=100: the condition number of A is 2e8, whereas U is only 2e4, and p(T) = 0.999999957539743. n=1000: the condition number of A is 2e12, whereas U is only 2e6, and p(T) = 0.999999999995867. A’s large condition number would yield greater errors, so the U method is more accurate than any of the methods from a).

d) For n=100: EI = 1.329. For n=1000: EI = 1.333. We can conclude that the bending stiffness is 1.333.

% a)

format long

n = 100;

a = ones(n,1);

Asparse = spdiags([a -4\*a 6\*a -4\*a a], -2:2, n, n);

Asparse(1,1) = 9;

Asparse(n,n) = 1;

Asparse(n,n-1) = -2;

Asparse(n-1,n) = -2;

Asparse(n-1,n-1) = 5;

L = 10;

F = .005;

h = L/n;

b = zeros(n,1);

for i = 1:n

b(i) = F\*h^4;

end

% I)

tic

A1 = Asparse\b;

toc

n1 = norm(A1);

Err1 = condest(Asparse)\*(norm(b-Asparse\*A1)/norm(b));

% II)

Adense = full(Asparse);

tic

A2 = Adense\b;

toc

n2 = norm(A2);

Err2 = condest(Asparse)\*(norm(b-Asparse\*A2)/norm(b));

% III)

z = ones(n,1);

tic

[A3, iter, T] = gs2(Asparse, b, z, 1e-8, 1e5)

toc

n3 = norm(A3);

% b)

U = spdiags([a, -2\*a a], 0:2, n, n);

U(1,1) = 2;

full(U);

tic

z = U\b;

x1 = U'\z;

toc

n4 = norm(x1);

% c)

CAsparse = condest(Asparse);

CU = condest(U);

p = max(abs(eig(full(T))));

% d)

x = 0;

for j = 1:n

x(j) = j\*h;

end

EI = (F\*L^4)./(6\*max(x1))

plot(x, x1)

grid on

title("Deflection z vs. x")

xlabel("x")

ylabel("z")