MACM 316 – Computing Assignment 5

Due Date: Friday November 20 at 11:00pm.
Submission Instructions: You must upload one .pdf file to Crowdmark that consists of 2 pages: page 1 is your report which should fit all discussions, output and plots onto a single page; and page 2 is a listing of your code. The deadline is **11:00pm** on the due date. The actual due time is set to 11:05pm and if Crowdmark indicates that you submitted late, you will be assigned a grade of 0. Your TA has emailed you a Crowdmark link that you should save since it will allow you to upload your completed assignments.

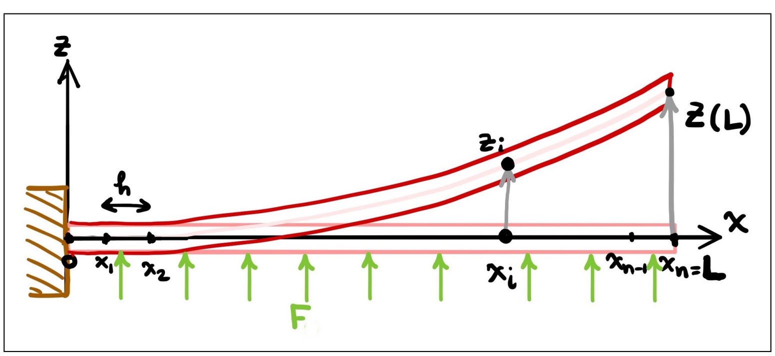
- Please review the **Guidelines for Assignments** carefully.
- Acknowledge any collaborations or assistance from colleagues/TAs/instructor.
- If you have any questions about Matlab or aspects of this assignment, then you are strongly encouraged to attend tutorials and drop-in workshops.

Computing Assignment – Banded Matrices In Solid Mechanics

Consider a horizontal flexible beam of length L that is clamped at one end but free to move vertically along the rest of its length; this is just one example of a *cantilever beam* that is studied extensively in engineering solid mechanics[†]. Referring to the picture below, let x be the horizontal coordinate and $z(x)$ be the vertical deflection of the beam. Divide the the beam into n equal sections of length $h = \frac{L}{n}$, defined by points $x_i = ih$ for $i = 1, 2, \ldots, n$. A discrete model for the forces and solid deformations along the beam yields a system of linear equations $Az = b$, where the $n \times n$ matrix A has the banded form

$$\begin{bmatrix} 9 & -4 & 1 & 0 & \cdots & \cdots & 0 \\ -4 & 6 & -4 & 1 & \ddots & & 0 \\ 1 & -4 & 6 & -4 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -4 & 6 & -4 & 1 \\ \vdots & & \ddots & 1 & -4 & 5 & -2 \\ 0 & \cdots & \cdots & 0 & 1 & -2 & 1 \end{bmatrix}$$

The n-vector $b = [b_i]$ is the given load force acting along the beam (including its own weight) and $z = [z_i]$ is a vector of the corresponding vertical deflections (away from $z = 0$) caused by this load force; both b_i and z_i are measured at location x_i. Take the beam to be *uniformly loaded*, which means that a constant upward force F is distributed along the entire beam length, so that each component of the load vector is a constant $b_i = F \cdot h^4$. For this problem, assume that the values of the physical constants are $L = 1.5\,m$ and $F = 0.4\,N/m$.



[†] For more explanation and lots of examples, see `http://en.wikipedia.org/wiki/Cantilever`.

(a) Solve the linear system $Az = b$ using the following three methods:

I. The GE+PP algorithm for sparse (banded) linear systems, which is the default algorithm used by Matlab's "\" operator when the matrix (call it `Asparse`) is of sparse type. You may find it easiest to set up your `Asparse` matrix using the `spdiags` command.

II. The GE+PP algorithm for dense linear systems, again using "\". Here, you need a dense version of A which you can obtain either with the `diag` command or more simply by typing

```
Adense = full(Asparse)
```

III. The Gauss-Seidel iterative algorithm, also using the matrix `Asparse`. You may make use of the `gs2` code from lectures, setting the parameters `tol = 1e-8` and `maxiter = 1e5`, and taking initial guess $z_0 = (1, 1, \dots, 1)^T$.

For each method, compute the solution with $n = 20, 50, 100$ and 500 points. How do the three methods compare in terms of cost? (use elapsed time from Matlab's `tic` / `toc` as a proxy for cost). How well do your three answers agree with each other? (use the vector 2-norm to compare the differences). Which result is most accurate?

Can you explain what is going on with the Gauss-Seidel solution as n increases?

(b) Next, verify that the matrix A has a UL factorization (yes, that's UL and not LU) of the form $A = UU^T$, where U is the upper triangular matrix

$$\begin{bmatrix} 2 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 1 & -2 & 1 \\ \vdots & & & \ddots & 1 & -2 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

For the same values of n in part (a), solve the linear system using the UU^T factorization via a sequence of two triangular solves (corresponding to forward and backward substitution). You can use the "\" command for both solves, but just make sure that your matrix U is set up as a sparse matrix. Compare your answer to the corresponding solution from Method I (sparse GE+PP) in terms of both accuracy and cost.

(c) Compute the condition numbers of both A and U, as well as the spectral radius $\rho(T)$ of the Gauss-Seidel iteration matrix T. Use this information to help explain your results from parts (a) and (b). The condition number for a large sparse matrix can only be estimated, and so for this purpose you should use the Matlab function `condest`.

(d) Choose what you consider to be your most accurate solution from parts (a)–(b), and use it to generate a plot of the deflection $z(x)$ versus x. In any introductory course in engineering solid mechanics, students learn about the following formula for the maximum deflection of a uniformly load beam which occurs at the free endpoint:

$$z_{max} = z(L) = \frac{FL^4}{6EI}$$

where EI is a material property of the beam called *flexural rigidity* or *bending stiffness* (with units $N \cdot m^2$). Use your solution for z to estimate the value of bending stiffness.

Extra reading: You can find out more about models for beam deflection at

`http://en.wikipedia.org/wiki/Deflection_(engineering)`