MA 202: Numerical Methods
Semester–II, Academic Year 2022-23
Tutorial set -3

Instructions

- **Submission deadline:** April 19, 2023.
- Any request for extension will not be entertained. Late submission (even by 1 second) will not be accepted.
- Problem 5 will be solved during the tutorial. The students have to solve the rest of the problems on their own and submit the answers to all problems.
- You have to write computer programs for each problem separately. You may use MATLAB or Python to write the programs.
- Submission will be through google classroom.
- Please put all the programs in a single 'zip' file and upload on google classroom.
- Each program should be named as follows: `Tut<Insert-Number>Prob<Insert-Number>.m/py`.
- The 'zip' file should be named as: `Tut<Insert-Number><Your-Name><Roll-Number>.zip /rar/...`. If additional information is required to run your program, please submit that as a PDF file contained within the same 'zip' file. Handwritten solutions, wherever appropriate must also be included in this same 'zip' file.
- Please make sure that your programs do not contain any error. In case there is a run time error, not credit for that particular problem will be given.
- Please do not submit screenshots of your programs. You must submit executable files.

---

(1) (a) Show by operation counting that the time complexity of the forward elimination in the gauss elimination method, applied to a system of n linear algebraic equations is: $\frac{2}{3}n^3 + O(n^2)$. (b) Similarly show that the time complexity of the Backward substitution is $\frac{1}{2}n^2 + O(n)$. What is the overall time complexity (or, the order of the number of operations) in Gauss-elimination?

(2) Develop a computer program to solve the following set of $2n$ non-linear algebraic equations, for the unknowns $\mathbf{x} = [c_1,\ x_1,\ c_2,\ x_2,\ ...,\ c_n,\ x_n]^T$.

$$f_k(\mathbf{x}) \equiv \sum_{j=1}^{n} c_j x_j^{k-1} - \int_{-1}^{1} t^{k-1} dt = 0, \qquad k = 1,\ 2,\ ...,\ 2n$$

Report your solutions for $n = 2,\ 3$ and 4. Note that you **should not** write separate codes for three separate values of n; rather just one program(s) should give the output and user can give input for n. These equations naturally arise while applying Gauss quadrature to approximately compute integrals. Try the following initial guesses: (a) for $n = 2$, try, $c_1 = c_2 = 1$, $x_1 = -1$ and $x_2 = 1$; (b) for $n = 3$, try: $c_1 = c_3 = 0.3$, $c_2 = 0.5$ and $x_1 = -1,\ x_2 = 0.1$ and $x_3 = 1$; (c) for $n = 4$, try: $[c_1, c_2, c_3, c_4] = [0.5, 0.5, 0.4, 0.4]$

and $[x_1, x_2, x_3, x_4] = [-1, -0.5, 0.5, 1]$. Do you find the convergence to be sensitive to the initial guess? For each value of n report one other initial guess that works and one for which the subsequent iterations diverge. You are not allowed to use in built solvers of MATLAB/Python.

(3) Prove that a sufficient condition for Gauss-Seidel iterations to converge for a set of n linear algebraic equation (written in compact form as: $\mathbf{Ax} = \mathbf{B}$) is that the LHS coefficient matrix be a "Diagonally Dominant Matrix", such that for any j-th row of the coefficient matrix, the following is satisfied:

$$|a_{jj}| \geq \sum_{k=1, k\neq j}^{n} |a_{jk}|, \quad j = 1,\ 2,\ ...,\ n,$$

where a_{ij} is the element in the i-th row and j-th column of the coefficient $n \times n$ matrix $\mathbf{A}$.

(4) Idealized spring-mass systems have numerous applications throughout engineering. For a certain arrangement of four springs in series being depressed with a force of $F = 2000$ N, it is found that at equilibrium, force-balance equations written for the displacement (in m) of each spring take the following form:

$$\begin{aligned} k_2(x_2 - x_1) &= k_1 x_1 \\ k_3(x_3 - x_2) &= k_2(x_2 - x_1) \\ k_4(x_4 - x_3) &= k_3(x_3 - x_2) \\ F &= k_4(x_4 - x_3) \end{aligned}$$

where the k's are spring constants. If k_1 through k_4 are 100, 50, 80, and 200 N/m, respectively, develop a computer code to compute $\mathbf{x} = [x_1, x_2, x_3, x_4]^T$ using (a) Gauss elimination and (b) Gauss-Seidel iterations. Also report the number of iterations taken to converge and the final error.

(5) Use (a) Gauss elimination and (b) Gauss-Seidel iterations to solve the following system of equations:

$$\begin{aligned} 2x_1 - 6x_2 - x_3 &= -38 \\ -3x_1 - x_2 + 7x_3 &= -34 \\ -8x_1 + x_2 - 2x_3 &= -20 \end{aligned}$$

(6) Formulate a method to compute the inverse of an $n \times n$ non-singular square matrix $\mathbf{A}$ using Gauss elimination. Develop a computer program to implement this method. Recall that if $\mathbf{B}$ is the inverse of $\mathbf{A}$, then $\mathbf{AB} = \mathbf{I}$, where $\mathbf{I}$ is the $n \times n$ Identity matrix. Run your code for a 4×4 ($n = 4$) Matrix of your choice.

(7) The specific volume of a super-heated steam is listed in steam tables for various temperatures. For example, at a pressure of 3000 lb/in^2:

T (°F)	v (ft^3/lb$_m$)
700	0.0977
720	0.12184
740	0.14060
760	0.15509
780	0.16643

Use first- through fourth-order lagrange polynomial interpolation to estimate v at $T = 750°\text{F}$.

(8) It is suspected that the high amounts of tannin in mature oak leaves inhibit the growth of the winter moth (*Operophtera bromata L., Geometridae*) larvae that extensively damage these trees in certain years. The following table (Table 1) lists the average weight of two samples of larvae at times in the first 28 days after birth. The first sample was reared on young oak leaves, whereas the second sample was reared on mature leaves from the same tree. Use Lagrange interpolation to approximate the average weight on days 12 and 16 for each sample.

Day	0	6	10	13	17	20	28
Sample 1 average weight (mg)	6.67	17.33	42.67	37.33	30.10	29.31	28.74
Sample 1 average weight (mg)	6.67	16.11	18.89	15.00	10.56	9.44	8.89

Table 1: Average weight of oak leaves (two samples)