M340L Coding Project

Abe Mathew Mankavil

November 2022

1 # of FLOPs - (Theoretical Problem)

Given a m x n matrix, we attempt to calculate the computational complexity of partial pivoting by counting the theoretical number of FLOPs. For each iteration of j in line 18 of *M340LCodingProj.py*, the Gaussian Elimination method does $(m-j)$ FLOPs for lines 18-20, $2(m-j)$ FLOPs for lines 22-23, and $(n-i-1)(m-i)$ FLOPs for lines 24-25. Summing the FLOPs in these segments of code give us the number of FLOPs for one iteration of the for-loop on line 18 of Gaussian elimination. This for-loop iterates over these FLOPs for whichever is less: the number of rows, m, or the number of columns, n. That is $min(m, n)$. Thus we say the total number of FLOPs to perform Gaussian Elimination with partial pivoting, where $m \leq n$, is:

$$f(m,n) = \sum_{i=1}^{m}[m-i] + 2\sum_{i=1}^{m}[m-i] + 2\sum_{i=1}^{m}[(n-i-1)(m-i)] \tag{1}$$

Using the rules of summations found in the Appendix we can simplify Eq. 1 to be:

$$= 2m^2n - nm(m+1) + \frac{m(m+1)(2m+1)}{3} - m^2(m+1) - \frac{m(m+1)}{2} + m^2 \tag{2}$$

The first term in Eq. 2 is the dominant term, because it grows the fastest as $m, n \longrightarrow \infty$. If we were to recount the number of FLOPs when $m \geq n$. then the leading term would be $2mn^2$ instead. Thus ignoring constants we can say that the partial pivoting algorithm for Gaussian elimination on an m x n matrix has a computational complexity of $\mathcal{O}(mn\ \min(m, n))$

2 Jacobi Method - (Extension Problem)

Iterative methods of solving systems of linear equations are seldom used on matrices of small dimensions because the time taken to complete the computations are often longer than solving the system using direct methods like Gaussian elimination/LU decomposition. The use cases of iterative methods are usually

found in systems that are very large and sparse, that is matrices with a large fraction of zero coefficient entries. Given a matrix that adheres to the aforementioned properties, these iterative techniques can prove to be especially efficient as it pertains to computational time. An additional advantage of iterative techniques is that they eliminate many instances of round off error, often found in Gaussian elimination, as they are not given a chance to accumulate. Iterative techniques such as the Jacobi method solve the original system of equations by starting with an initial guess, $\overrightarrow{x}^{(0)}$, that is used to generate a sequence of approximate solutions $\{\overrightarrow{x}^{(1)}, \overrightarrow{x}^{(2)},..., \overrightarrow{x}^{(n)}\}$ with each subsequent solution based on the previous approximation. To begin we solve each row of the coefficient matrix for the x_i in the main diagonal position (Solve the 1^{st} equation for x_1, solve the 2^{nd} equation for x_2, and so on). Then for each $\overrightarrow{x}^{(k)}$ we substitute in the elements of $\overrightarrow{x}^{(k-1)}$, that is $\{x_1^{(k-1)}, x_2^{(k-1)}, ... x_n^{(k-1)}\}$. Common convention is to begin by setting all elements of $\overrightarrow{x}^{(0)}$ to zero and use those values in the calculation of $\overrightarrow{x}^{(1)}$. The iterative method is described formally below for $k \geq 1$ where a represents elements of the coefficient matrix and b are the elements of the matrix's constant vector. This iterative process transforms the original system $\mathbf{A}\vec{x} = \vec{b}$ to an equivalent system of the form $\vec{x}^{(k)} = \mathbf{T}\vec{x}^{(k-1)} + \vec{c}$, where $\mathbf{T}$ is the coefficient matrix where each row is solved in terms of the x_i in the main diagonal position of the coefficient matrix $\mathbf{A}$.

$$x_i^{(k)} = \frac{1}{a_{ii}} \left[\sum_{\substack{j=1 \\ j \neq i}}^{n} [(-a_{ij} x_j^{(k-1)}) + b_i] \right], where\ i = 1, 2, 3, ...n \tag{3}$$

As k increases, the approximations $\overrightarrow{x}^{(k)}$ will begin to converge towards an accurate solution, $\overrightarrow{x}$, of the original matrix equation, as long as the spectral radius of the iterative matrix is strictly less than one ($\rho(\mathbf{T}) < 1$). We define the spectral radius $\rho(\mathbf{A})$, for an arbitrary matrix $\mathbf{A}$, as the maximum of the absolute values of $\mathbf{A}$'s eigenvalues. A common eye test to confirm convergence is that a diagonally dominant coefficient matrix ensures the Jacobi method will converge, however convergence of the method does not necessarily imply diagonal dominance. Below we see an example of a coefficient matrix that wouldn't converge using the Jacobi method.

$$\mathbf{A} = \begin{bmatrix} 1 & 0.75 & 0.75 \\ 0.75 & 1 & 0.75 \\ 0.75 & 0.75 & 1 \end{bmatrix} \quad \mathbf{T} = \begin{bmatrix} 0 & -0.75 & -0.75 \\ -0.75 & 0 & -0.75 \\ -0.75 & -0.75 & 0 \end{bmatrix}$$

$$\rho(\mathbf{T}) = 1.5 > 1 \longrightarrow \therefore \text{Jacobi method does not converge for } \mathbf{A}$$

The spectral radius of $\mathbf{T}$ is greater than 1 because its eigenvalues are $\lambda = -\frac{3}{2}, \frac{3}{4}$ and the maximum absolute values of its eigenvalues is clearly $\frac{3}{2}$ or 1.5.

3 Appendix

3.1 Summation Rules

$$\sum_{i=1}^{n} c = cn \tag{4}$$

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} \tag{5}$$

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \tag{6}$$