Lecture Notes For: Numerical Methods for Scientific Computing

Ali Fele Paranj
alifele@student.ubc.ca

January 31, 2023

In this document, I have organized different numerical methods that are commonly used for scientific computing.

Chapter 1

System of Linear Equations

1.1 Direct Methods to Solve the System of Equations

1.1.1 LU Decomposition

Will be completed soon!

1.1.2 RQ Decomposition

Will be completed soon!

1.1.3 Guassian Elimination

Will be completed soon!

1.1.4 Tridiagonal Matrix

Will be completed soon!

1.2 Approximate Method to Solve the System of Equations

Suppose that want to solve the following system of equations:

$$\boldsymbol{A}x = b$$

.

Let the matrix $\boldsymbol{A}$ to be: $\boldsymbol{A} = \boldsymbol{S} - \boldsymbol{T}$, in which $\boldsymbol{S}$ and $\boldsymbol{T}$ are the some matrices which are chosed in a smart way!. Let's plug in the new value of $\boldsymbol{A}$ in the system of linear equations:

$$\begin{aligned}
(\boldsymbol{S} - \boldsymbol{T})x &= b \\
\boldsymbol{S}x &= \boldsymbol{T}x + b \\
x &= \boldsymbol{S}^{-1}(\boldsymbol{T}x + b) = \boldsymbol{S}^{-1}\boldsymbol{T}x + \boldsymbol{S}^{-1}b
\end{aligned}$$

So we will have:

$$\boxed{x = \boldsymbol{S}^{-1}\boldsymbol{T}x + \boldsymbol{S}^{-1}b} \tag{1.2.1}$$

Now let's plug in an initial guess x_0 in RHS of the the equation 1.2.1 and name it x_1. Then we can do this repeatedly to get the following equations:

$$\begin{aligned}
x_1 &= \boldsymbol{S}^{-1}\boldsymbol{T}x_0 + \boldsymbol{S}^{-1}b \\
x_2 &= \boldsymbol{S}^{-1}\boldsymbol{T}x_1 + \boldsymbol{S}^{-1}b \\
&\vdots \\
x_n &= \boldsymbol{S}^{-1}\boldsymbol{T}x_{n-1} + \boldsymbol{S}^{-1}b
\end{aligned}$$

So the iterative update equation can be written as:

$$x_{i+1} = \boldsymbol{S}^{-1}\boldsymbol{T}x_i + \boldsymbol{S}^{-1}b \tag{1.2.2}$$

To see if we have get closer to the actual solution of the system of equations, let's asume that the actual solution is x. So let's define the following errors:

$$\begin{aligned}
\epsilon_0 &= x - x_0 \\
\epsilon_1 &= x - x_1 \\
\epsilon_2 &= x - x_2 \\
&\vdots \\
\epsilon_n &= x - x_n
\end{aligned}$$

By pluggin in $x_0 = x - \epsilon_0$ in equation 1.2.1 we will get:

$$\begin{aligned}
x_1 &= \boldsymbol{S}^{-1}\boldsymbol{T}(x - \epsilon_0) + \boldsymbol{S}^{-1}b \\
&= \underbrace{\boldsymbol{S}^{-1}\boldsymbol{T}x + \boldsymbol{S}^{-1}b}_{x} - \boldsymbol{S}^{-1}\boldsymbol{T}\epsilon_0 \\
&= x - \boldsymbol{S}^{-1}\boldsymbol{T}\epsilon_0 = x - \epsilon_1 \\
&\Rightarrow \boxed{\epsilon_1 = \boldsymbol{S}^{-1}\boldsymbol{T}}
\end{aligned}$$

Using the same logic we will get:

$$\epsilon_n = (\boldsymbol{S}^{-1}\boldsymbol{T})^n \epsilon_0 \tag{1.2.3}$$

So using this iterative method to find the approximate solution of the system of the linear equations, we will converge to the actual solution if the largest eigenvalue of the matrix $\boldsymbol{S}^{-1}\boldsymbol{T}$ is smaller than one. Now the only problem is to find the value of $\boldsymbol{S}$ is a clever way such that it meets the convergence criteria and is easy to invert. Note that the time complexity of inverting a matrix is $O(N^3)$. So an inapproporate choice of $\boldsymbol{S}$ will be very costly.

1.2.1 Jacobi Method

One idea for $\boldsymbol{S}$ is a diagonal matrix that contains the diagonal elements of the matrix $\boldsymbol{A}$

$$\boldsymbol{S} = \begin{pmatrix} A_{11} & 0 & \cdots & 0 \\ 0 & A_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{nn} \end{pmatrix} \tag{1.2.4}$$

And for $\boldsymbol{T}$, since $\boldsymbol{A} = \boldsymbol{S} - \boldsymbol{T}$, so we can write:

$$\boldsymbol{T} = \begin{pmatrix} 0 & -A_{12} & \cdots & -A_{1n} \\ -A_{21} & 0 & \cdots & -A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -A_{n1} & -A_{n2} & \cdots & 0 \end{pmatrix} \tag{1.2.5}$$

Note that the conversion criteria (which is $|\lambda_{max}(\boldsymbol{S}^{-1}\boldsymbol{T})| < 1$) still need to be checked. This way of choosing $\boldsymbol{S}$ and $\boldsymbol{T}$ is interesting because calculating the inverse of a diagonal matrix has $O(N)$ time complexity. So calculating the RHS of the update equation (equation 1.2.2) will have a lower time complexity.

1.2.2 Guass Seidel Method

The matrix $\boldsymbol{S}$ can be chosen in a way to be a lower triangular matrix:

$$\boldsymbol{S} = \begin{pmatrix} A_{11} & 0 & \cdots & 0 \\ A_{21} & A_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{pmatrix} \tag{1.2.6}$$

So the matrix $\boldsymbol{T}$ will be:

$$\boldsymbol{T} = \begin{pmatrix} 0 & -A_{12} & \cdots & -A_{1n} \\ 0 & 0 & \cdots & -A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \tag{1.2.7}$$

With choosing $\boldsymbol{S}$ to be a triangular matrix, we can avoid calculating the $\boldsymbol{S}^{-1}$ for equation 1.2.2. Instead we can write the update rule as:

$$\boldsymbol{S}x_{i+1} = \boldsymbol{T}x_i + b \tag{1.2.8}$$

and calculate x_{i+1} via backward or forward substitution which has a $O(N^2)$ time complexity. Note that will this specific choice of $\boldsymbol{S}$ and $\boldsymbol{T}$ we need to verify the conversion criteria to make sure the error will converge to the zero vector.

1.3 Solving Under Determined and Over Determined System of Equations

The under determined and over determined system of equation can be defined as the following:

Definition: Under Determined and Over Determined System of Equations

- **Over determined system of equations:** If a system of linear equations has more equations than the number of variables then we will have an over determined system. An over determined system of equation will generally have *no* solutions.

- **Under determined system of equations:** If a system of linear equations has more variables than the number of equations then we will have an under determined system. An under determined system of equation will generally have *infinite* number of solutions.

Chapter 2

Matrices

2.1 Eigenvalue and Eigenvectors

2.1.1 Power Method

This is to calculate the largest eigenvalue of a matric