main

Contents

Basic terminology and definitions

An important aspect of computation is the error that shows up when we try to model a system. Error often propagates through our calculations.
One example of a source of error is in the limited capacity of a computer memory to hold a number. Ex, a floating point number in IEEE 754 standard uses 52 bits for its mantissa in 64 bit machines. Thus, accuracy can be lost

Machine Epsilon

A machine epsilon can be defined as the smallest positive number that can be added to 1 and still have the result be greater than 1
Any postive number smaller than the machine epsilon that's added to 1 will result in the sum be treated as 1. This is due to the finite precision of floating point numbers

in computers. 1+number_less_than_epsilon is identical to 1 in the computer's memory and ALU.
The following is a python script for calculating machine epsilon:

```python
import math
def main():
    espsilon = 1
    while(1+espsilon>1):
        espsilon = espsilon*0.999999
    print(espsilon)
    print(math.log2(1/espsilon)) # ~53 for 64 bit machines
    return 0

if __name__ == '__main__':
    main()
```


Condition number

One method of identifying the amount of error is by the condition number.
The Condition number of a continuous and differentiable function f at the point c is defined as:

$$|\frac{f'(c)}{f(c)}c|$$

It tells us the error in the output as a ratio of the error in the input.

Ill-conditioned and well-conditioned

The process of evaluating a continuously differentiable function f at a point $x = c$ is said to be Well-Conditioned if the Condition number is small. Else it is Ill-Conditioned

Stable and unstable computation

The process of evaluating a function f, involving n steps is said to have instability if at-least one of the steps is Ill-conditioned . If all steps are Well-Conditioned , then the process is stable

Methods of solving Linear Systems

Direct Methods

Given a system of n linear equations, in n variables, we can employ various methods to solve them.

Gaussian Elimination

We could simply try the Gaussian elimination method and solve for x_n and then apply backward-substitution to solve for $x_i \quad i \in 1, 2 \dots n-1$

However floating point errors may result in a on-zero pivot point to exist where it should have been zero, thus causing large errors in the solution to the equations.

To rectify this, a popular approach is partial-pivoting

Modified Gaussian Elimination with Partial Pivoting

In this version of the Gaussian elimination, rows are interchanged before the elimination step such that a_{ii} is the largest(magnitude wise) element in $\{a_{ik} \quad : \quad k \in i, i+1, i+2 \dots n\}$

Thomas method for Tri-diagonal Systems

Thomas method is a special case of Guassian elimination where the latter can be simplified to increase efficiency. It can only be used in tri-diagonal systems of linear equations.

LU factorization

It is often useful to factorize a matrix into simpler, more easily solvable forms.
One such form is the LU form, where L is a lower triangle matrix and U is an upper triangle matrix.
Three popular methods of factorizing a matrix are:

Dolittle's factorization

A matrix A is said to have a Doolittle factorization if there exists a lower triangular matrix L with all diagonal elements as 1, and an upper triangular matrix U such that:

$$A = LU$$

Ex:

$$\begin{bmatrix} 1 & 1 & -1 \\ 1 & 2 & -2 \\ -2 & 1 & 1 \end{bmatrix}$$

has a Doolittle factorization of:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -2 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 2 \end{bmatrix}$$

Crout's Factorization

A matrix A is said to have a Crout factorization is there exists an up triangular matrix U with all diagonal elements as 1, and a lower triangular matrix L such that:

$$A = LU$$

Cholesky's Factorization

A matrix A is said to have a Cholesky Factorization if there exists a lower triangular matrix L such that

$$A = LL^T$$

Note, this implies A is symmetric
Additionally, we know that every symmetric matrix, positive definite has an LL^T factorization(with L having positive diagonal elements) and hence a Cholesky Factorization

> ⓘ Note
>
> A matrix A is said to be positive definite if
>
> $$x^T Ax > 0 \qquad \forall x \in \mathbb{R}^n$$

Norms

Vector Norm

A norm of a vector is a generalization of the | | operator.
It has the following properties:

$$||x|| \geq 0 \ \ \forall x \in \mathbb{R}^n$$

$$||x|| = 0 \ \text{ if and only if } x = 0$$

$$||\alpha x|| = |\alpha| ||x|| \text{ for all } x \in \mathbb{R}^n \text{ and for all } \alpha \in \mathbb{R}$$

$$||x + y|| \leq ||x|| + ||y|| \quad \forall x, y \in \mathbb{R}^n$$

An l_n norm is defined as follows:

$$||v||_n = \left(\sum_{j=0}^{k} |v_j|^n \right)^{\left(\frac{1}{n}\right)}$$

The ∞ norm is simply $max(|v_1|, |v_2|, \dots |v_k|)$
While 1st norm is $v_1 + v_2 + \cdots + v_k$

Matrix Norm

A matrix norm on the vector space is a function $||.|| : M_n(R) \to [0, \infty)$ having similar properties to the matrix norm (the four properties)

Matrix Norm subordinate to a vector norm

Let ||.|| be a vector norm on $\mathbb{R}^n$ and let $A \in M_n(\mathbb{R})$. The matrix norm of A subordinate to the vector norm || . || is defined by

$$||A|| := \sup \{ \| Ax \| \quad : \ x \in \mathbb{R}^n, \| x \| = 1 \}$$

Note

$$||A||_\infty = \max_{1 \le i \le n} \sum_{i=1}^{n} |a_{ij}|$$

$$||A||_1 = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}|$$

$$||A||_2 = \sqrt{\max_{1 \le i \le n} |\lambda_i|}$$

$\lambda_1, \lambda_2, \dots, \lambda_n$ are the eigenvalues of the matrix $A^T A$. This norm is also known as the spectral norm of A

Condition number for a matrix

We can now define a condition number for a matrix.
The condition number (often denoted as $\kappa(A)$) is defined as:

$$\kappa(A) := ||A||||A^{-1}||$$

The condition number gives a measure of how "close" a matrix is to a singular matrix.
Let $A \in M_n(\mathbb{R})$ be non-singular. Then, for any singular n x n matrix B, we have:

$$\frac{1}{\kappa(A)} \leq \frac{||A - B||}{||A||}$$

Iterative methods

1. Jacobi method. Step wise calculation of next state step from present state. State is update after the calculation is complete. Implemented here: https://sam-martis.github.io/ChaosTheoryAttractors/
2. Guass-Seidel method. Similar to Jacobi, but state is updated every every substep calculation. Faster convergence.
   Both methods converge is the system of linear equations in the form of matrix A is diagonally dominant.

> ⓘ Diagonally Dominant Matrices
> A matrix A is said to be diagonally dominant if it satisfies the inequality
>
> $$\left(\sum_{j=1, j \neq i}^{n} |a_{ij}| \right) < |a_{ii}|, \quad i = 1, 2, \ldots, n$$

Stationary Iterative methods

An iterative method is called stationary if it can be written in the form

$$x^{(k+1)} = Bx^{(k)} + c$$

for some $B \in M_n(\mathbb{R})$ and $c \in \mathbb{R}^n$

Spectral radius

$$p(A) = \max_{j=1,2,\ldots,n} |\lambda_j|$$

If the spectral radius is less than 1, the matrix compresses the space. Thus, it would make sense that $lim_{n\to\infty} A^n x = 0$
No matter the vector, it gets shrunk.