Student: Leidong Xu
Journal: Fall 2018
Class: Applied Math I
Instructor: Pietro Poggi-Corradini

Week 1: This week, I attended all the three lecture of Applied Math I and the major topic was going over linear algebra.

References or papers consulted:

- *Convergence Analysis of the Nonlinear Coarse-Mesh Finite Difference Method for One-Dimensional Fixed-Source Neutron Diffusion Problem*
- *A Comparison of Coarse Mesh Rebalance (CMR) and Coarse Mesh Finite Difference (CMFD) Acceleration Methods for the Neutron Transport Calculations*

Things I learned:

Went Over

- Eigenvalue and Eigenvector
- Finite element method
- Matrix Factorization
- Gauss Elimination
- LU factorization Jordan decomposition and Schur decomposition $A = QUQ^{-1}$

New

- Condition number
  We compute $f(x+\delta x) - f(x) \approx |\delta x| \cdot |f'(x)|$ and $|f'(x)|$ is absolute condition number.
  We compute $\frac{|f(x+\delta x)-f(x)|}{|f(x)|} \approx \frac{|\delta x|}{|x|} \cdot \frac{|f'(x)|\cdot|x|}{|f(x)|}$ and $|f'(x)|$ is relative condition number.
- Absolute error and relative error: $|y - \bar{y}|$ and $\frac{|y-\bar{y}|}{|y|}$
- Backward error δx could be defined by $alg(x) = f(x + \delta x)$. This error is used for verify the stability of an algorithm.
- Round-off effects The results could lose its digits by a bad algorithm,etc switch rows before using Gauss elimination.
- Floating Point Algorithm A float number stored in computer included sign, fraction, base and exponent. An IEEE double precision has 64 total bits.
- overflow/underflow threshold are the maximum and minimum value of an certain standard float number. etc, the underflow of IEEE float is 2^{1022} and overflow is 2^1024
- machine epsilon ϵ Machine epsilon could be define as $fl(a) = a(1+\delta)$ $|\delta| \leq \epsilon$ when all the numbers are not beyond the overflow/underflow.

Software Code:

- Create a repository on Github for all the assignments and Journals in the rest of the semester.

Student: Leidong Xu
Journal: Fall 2018
Class: Applied Math I
Instructor: Pietro Poggi-Corradini

Also, only for this week: Personal Intro

Hi, my name is Leidong Xu. And I have been studying mechanical and nuclear engineering in K-state for five years. This academic year will be the second year for my master degree. My thesis is related to using numerical method solving engineering and science problems on Computational Fluid Dynamics and Nuclear systems, which include preconditioner(sparse matrix) and geometric multigrid method.

I did not take any pure mathematic class in the past 3 years, and almost all the courses I took last year were using programming to solve PDEs and ODEs. Things get boring when I continue stayed in finite difference all the time. This also the main reason I come to this class, I hope I can have a better understanding of linear algebra and have a better understanding of those numerical methods in a view of applied math.

Student: Leidong Xu
Journal: Fall 2018
Class: Applied Math I
Instructor: Pietro Poggi-Corradini

Week 2: This week, I studied the rest parts of Chapter 1 which included vector and matrix norm and also a good start of Chapter

References or papers consulted:

- *Linear Independence Oracles and Applications to Rectangular and Low Rank Linear Systems*

Things I learned:

- $\mathbb{R}^{(3x3)}$ means real number matrix and $\mathbb{C}^{(3x3)}$ means complex number matrix.
- some norm axioms
- P-norm could be defined as $||x||_p = (\Sigma_i |x_i|^p)^{1/p}$.
- Euclidean norm: p-norm when p = 2
- taxicab norm: p-norm when p = 1
- $\infty - norm$ or infinity norm: p-norm p = ∞
- Cauchy-Schwartz inequality: $| < x, y > | \leq \sqrt{< x, x > \cdot < y, y >}$
- Inner product: Hard to give a definition, but could be think as a space created by vectors and scalars, related Legendre Polynomial.
- General strategy to show a = 0: a= -a
- Max-norm: $max_{ij}|a_{ij}$ and Frobenius norm :$(\Sigma|a_{ij}|^2)^{1/2}$
- Spectral theory for symmetric matrix
- An operator norm is a matrix norm.
- Matrix norm could be think as the maximum strectch.
- How to compute singular value of an matrix
- residual: $r = A\hat{x} - b$
- condition number of a matrix: $\kappa(A) = ||A^{-1}|| \cdot ||A||$

Software Code:

- Finished programming hw 1. The questions I picked are Problem.32 Most nonzero elements in row and Problem 17. Find all elements less than 0 or greater than 10 and replace them with NaN.
- These problems are pretty easy, and good to know we can define NaN by float('nan').

```
'''
Problem 32. Most nonzero elements in row
Given the matrix a, return the index r of the row with the most nonzero
elements. Assume there will always be exactly one row that matches this
```

```
criterion.
'''

import numpy as np


def mostnonzero(matrix):
    row = []
    max_zero = 0
    for i in range(matrix.shape[0]):
        N_nonzero = 0
        for j in range(matrix.shape[1]):
            if matrix[i, j] != 0.:
                N_nonzero += 1
        if max_zero < N_nonzero:
            row = [i]
            max_zero = N_nonzero
        elif max_zero == N_nonzero:
            row.append(i)
    row = np.array(row) + 1
    print('the' + ' ' + str(row) + 'th row has the maxinum nonzero element')
    return row


'''
Find all elements less than 0 or greater than 10 and replace them with NaN
'''


def replacenan(matrix):
    for i in range(matrix.shape[0]):
        for j in range(matrix.shape[1]):
            if matrix[i, j] > 10 or matrix[i, j] < 0:
                matrix[i, j] = float('nan')
    return matrix
```

Student: Leidong Xu
Journal: Fall 2018
Class: Applied Math I
Instructor: Pietro Poggi-Corradini

Week 3: This week, I went over some knowledge about series and condition number. I was lost those content belong to which chapter on the book.

References or papers consulted:

- *What every computer scientist should know about floating-point arithmetic*

Things I learned:

- Pi notation: $\prod_n^{i=m} x_i = x_m \cdot x_{m+1} \cdot x_{m+2} \cdot \cdots \cdot x_n$
- Vandermonde matrix is a matrix have a geometric series in each row, $V_{i,j} = \alpha_i^{j-1}$ and the determinant of Vandermonde matrix is $\prod_{1 \le i < j \le n}(\alpha_j - \alpha_i)$, for example

```
det (  [[  1,   1,   1,   1],
       [  1,   2,   4,   8],
       [  1,   3,   9,  27],
       [  1,   5,  25, 125]]) = (5-3)*(5-2)*(5-1)*(3-2)*(3-1)*(2-1)
```

- ill-condition: A problem with a low condition number is said to be well-conditioned, while a problem with a high condition number is said to be ill-conditioned.
- geometric series:$1 + x + x^2 + x^3 + x^4 + \cdots x^n = \sum_{k=0}^{\infty} x^k$
- geometric interpretation
- If the vector space X is finite dimensional, all norms are equivalent.

Software Code:

- I have not finished PA 2, but there are two new numpy functions and some latex functions.
- numpy.finfo: Machine limits for floating point types. For example:

```
print(np.finfo(float).eps)
# 2.22044604925e-16

print(np.finfo(np.float32).eps)
# 1.19209e-07
```

- numpy.vander: Generate a Vandermonde matrix. Most common example:

```
>>> x = np.array([1, 2, 3, 5])
>>> np.vander(x, increasing=True)
array([[  1,   1,   1,   1],
       [  1,   2,   4,   8],
       [  1,   3,   9,  27],
       [  1,   5,  25, 125]])
```

- $\mathbb{R}$ by

```
\mathbb{}
```

- $\sqrt{< x, x >}$ by

  ```
  \sqrt{}
  ```

- Pi notation $\prod_{i=1}^{n} a_i$ by

  ```
  \prod_{i = 1}^{n} a_{i}
  ```

Student: Leidong Xu
Journal: Fall 2018
Class: Applied Math I
Instructor: Pietro Poggi-Corradini

Week 4: This week, I went over some knowledge about different matrix factorization.

References or papers consulted:

- *Model order reduction using DMD modes and adjoint DMD modes*

Things I learned:

- SPD: symmetric positive-defined matrix.
- Singular value can be thought dimensional, etc, largest singular value is $||A||_2$, which could be treated as the length of semi-major axis.
- GramSchmidt process
- LU factorization
- leading principal: The determinant of a principal submatrix is called the principal minor of A. The leading principal submatrix of order k of an n n matrix is obtained by deleting the last n k rows and column of the matrix.
- Matrix Minor: the remain matrix from delecting some rows and columns from original matrix.
- Permutation matrix: a matrix have a '1' at every row and column, all others are zero.
- Cholesky factorization(Cholesky decomposition): a decomposition of a Hermitian, positive-definite matrix into the product of a lower triangular matrix and its conjugate transpose.

Software Code:

- N/A

Student: Leidong Xu
Journal: Fall 2018
Class: Applied Math I
Instructor: Pietro Poggi-Corradini

Week 5: This week, I went over some knowledge about Cholesky factorization and finite difference method. I'm pretty familiar with this method but I have never think about adjust the matrix feature to achieve a better performance when using directly method.

References or papers consulted:

- *On Dynamic Mode Decomposition: Theory and Applications*

Things I learned:

- ONB: Orthonormal Basis
- spectral theorem for symmetric matrices
- Two different ways to think matrix multiplication
- Change of Variables
- Gershgorin circle theorem
- Eigenvalue of A is real $\rightarrow$ All the element of A are strictly positive

Software Code:

- The machine epsilon of float 64 (double precision) is around 2.220446049250313e-16, and the smallest nonzero number could be shown is around 1.1102230246251565e-16.
- np.log could directly apply on a array, so do not use for loop.
- machine epsilon could be import by np.finfo(float).eps
- numpy.arange could be a alternative of np.linspace. Instead of input how many points needed, we could define the start point, end point and the space.