Lab Assignment 5 (01/31/2023)
Saichandana V (vsc@iastate.edu)

**Tasks:**

1. Practicing numpy arrays for linear algebra.
2. Using Gaussian-Elimination to solve the linear system given in class and compare to the results obtained by numpy.linalg.solve.
3. Submit source codes and screenshots.
4. Push the code to GitHub.
5. File transfer and Up-to-date in Nova cluster

**Submission Files and Results:**

1. I practiced numpy arrays for linear algebra. I learned that numpy has various libraries to solve linear algebra math. Here is the snapshot of the python prompt of my practiced examples

```
(base) ubuntu@ubuntu-vm:~/Documents/chandanaWorkspace/CPRE 525 Spring 2023/CPRE5
25Spring2023/4. Lab Assignment 5$ python3
Python 3.9.12 (main, Apr  5 2022, 06:56:58)
[GCC 7.5.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> A = np.array([[1, 2], [3, 4]])
>>> B = np.array([[5, 6], [7, 8]])
>>> C = np.dot(A, B)
>>> print('Matrix multiplication of A and B',C)
Matrix multiplication of A and B [[19 22]
 [43 50]]
>>> C_transpose = C.T
>>> print('Transpose of a matrix',C_transpose)
Transpose of a matrix [[19 43]
 [22 50]]
>>> C_inverse = np.linalg.inv(C)
>>> print('Inverse of a matrix', C_inverse)
Inverse of a matrix [[ 12.5   -5.5 ]
 [-10.75   4.75]]
>>> C_det = np.linalg.det(C)
>>> print('Determinant of a matrix', C_det)
Determinant of a matrix 4.000000000000017
>>> eig_values, eig_vectors = np.linalg.eig(C)
>>> print("Eigenvalues:", eig_values)
Eigenvalues: [5.80198014e-02 6.89419802e+01]
```

```
>>> U, s, V = np.linalg.svd(C)
>>> print("Singular value decomposition, U:", U)
Singular value decomposition, U: [[-0.40334516 -0.91504791]
 [-0.91504791  0.40334516]]
>>> print("s:", s)
s: [7.20693896e+01 5.55020657e-02]
>>> print("V:", V)
V: [[-0.65229661 -0.75796381]
 [-0.75796381  0.65229661]]
>>> Q, R = np.linalg.qr(C)
>>> print("QR decomposition, Q:", Q)
QR decomposition, Q: [[-0.40416385 -0.9146866 ]
 [-0.9146866   0.40416385]]
>>> print("QR decomposition, R", R)
QR decomposition, R [[-47.01063709 -54.62593487]
 [  0.           0.08508713]]
>>> A = np.array([[3, 1], [1, 2]])
>>> b = np.array([9, 8])
>>> print('input A: ',A)
input A:  [[3 1]
 [1 2]]
>>> print('input b: ', b)
input b:  [9 8]
>>> x = np.linalg.solve(A, b)
>>> print('Solving the linear equation Ax = b. x:',x)
Solving the linear equation Ax = b. x: [2. 3.]
>>> result = np.dot(A, x)
>>> print('Verification of the solution by computing the dot product of A and x'
, result)
Verification of the solution by computing the dot product of A and x [9. 8.]
>>>
```

2. Prepared python file for the above-practiced commands and submitted the code. Here is the snapshot

```
(base) ubuntu@ubuntu-vm:~/Documents/chandanaWorkspace/CPRE 525 Spring 2023/CPRE5
25Spring2023/4. Lab Assignment 5$ python3 practice_numpyLinearAlgebra.py
Matrix multiplication of A and B [[19 22]
 [43 50]]
Transpose of a matrix [[19 43]
 [22 50]]
Inverse of a matrix [[ 12.5   -5.5 ]
 [-10.75   4.75]]
Determinant of a matrix 4.000000000000017
Eigenvalues: [5.80198014e-02 6.89419802e+01]
Singular value decomposition, U: [[-0.40334516 -0.91504791]
 [-0.91504791  0.40334516]]
s: [7.20693896e+01 5.55020657e-02]
V: [[-0.65229661 -0.75796381]
 [-0.75796381  0.65229661]]
QR decomposition, Q: [[-0.40416385 -0.9146866 ]
 [-0.9146866   0.40416385]]
QR decomposition, R [[-47.01063709 -54.62593487]
 [  0.           0.08508713]]
input A:  [[3 1]
 [1 2]]
input b:  [9 8]
Solving the linear equation Ax = b. x: [2. 3.]
Verification of the solution by computing the dot product of A and x [9. 8.]
```

3. I have created a function using Gaussian-Elimination to solve the linear system given in class and compared the resultswith the numpy.linalg.solve. Submitted the code. Here is the snapshot of the result.

```
(base) ubuntu@ubuntu-vm:~/Documents/chandanaWorkspace/CPRE 525 Spring 2023/CPRE5
25Spring2023/4. Lab Assignment 5$ python guass_elimination_solve.py
The solution x_numpy is:
x1 = -4.0
x2 = -6.000000000000001
x3 = -12.000000000000002

 The solution [x_mine] is:
x1 = -4.0
x2 = -6.0
x3 = -12.000000000000002

The solution x_mine-x_numpy is:
x1 = 0.0
x2 = 8.881784197001252e-16
x3 = 0.0
```

4. Push the code to GitHub
    a. Git status check

```
(base) ubuntu@ubuntu-vm:~/Documents/chandanaWorkspace/CPRE 525 Spring 2023/CPRE5
25Spring2023$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   4. Lab Assignment 5/guass_elimination_solve.py
        new file:   4. Lab Assignment 5/practice_numpyLinearAlgebra.py

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   4. Lab Assignment 5/guass_elimination_solve.py
        modified:   4. Lab Assignment 5/practice_numpyLinearAlgebra.py
```

    b. Git add and git status

```
(base) ubuntu@ubuntu-vm:~/Documents/chandanaWorkspace/CPRE 525 Spring 2023/CPRE5
25Spring2023$ git add .
(base) ubuntu@ubuntu-vm:~/Documents/chandanaWorkspace/CPRE 525 Spring 2023/CPRE5
25Spring2023$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   4. Lab Assignment 5/guass_elimination_solve.py
        new file:   4. Lab Assignment 5/practice_numpyLinearAlgebra.py
```

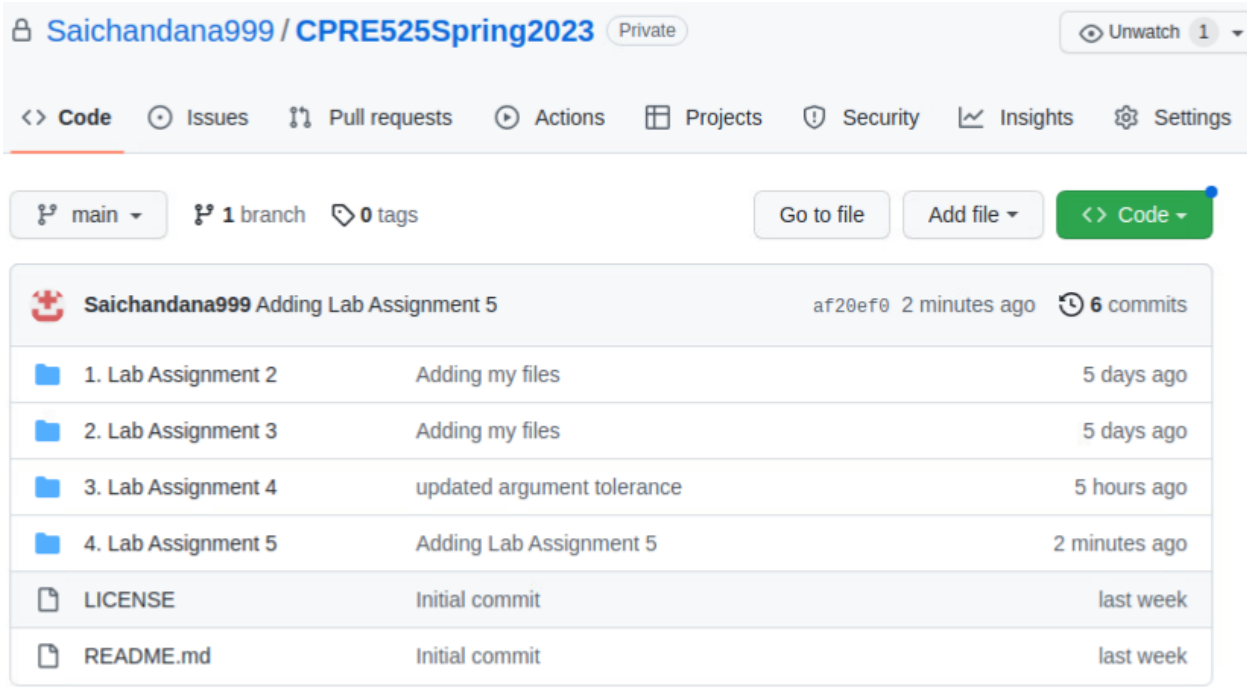
c. Git committed the files

```
(base) ubuntu@ubuntu-vm:~/Documents/chandanaWorkspace/CPRE 525 Spring 2023/CPRE5
25Spring2023$ git commit -m 'Adding Lab Assignment 5'
[main af20ef0] Adding Lab Assignment 5
 2 files changed, 123 insertions(+)
 create mode 100644 4. Lab Assignment 5/guass_elimination_solve.py
 create mode 100644 4. Lab Assignment 5/practice_numpyLinearAlgebra.py
```

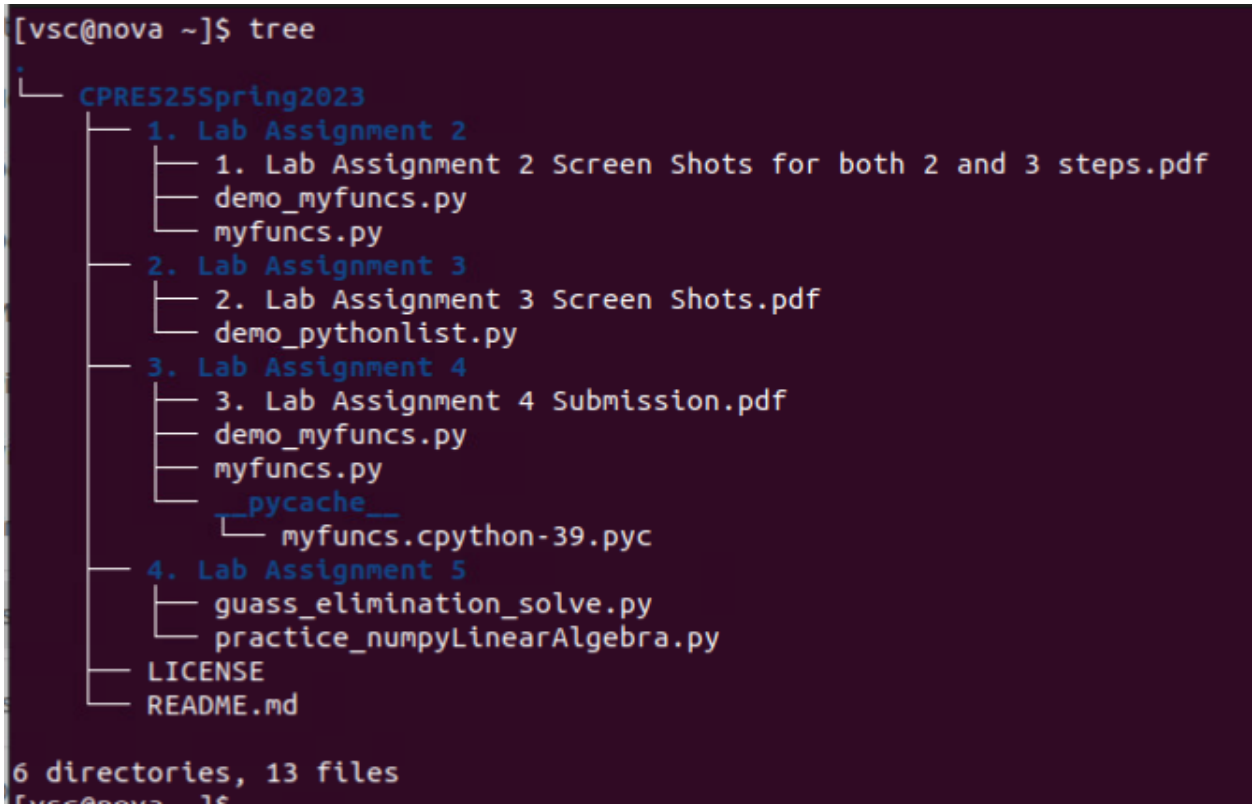
d. Git push

```
(base) ubuntu@ubuntu-vm:~/Documents/chandanaWorkspace/CPRE 525 Spring 2023/CPRE5
25Spring2023$ git push
Username for 'https://github.com': Saichandana999
Password for 'https://Saichandana999@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.65 KiB | 1.65 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Saichandana999/CPRE525Spring2023.git
   57d9329..af20ef0  main -> main
```

e. Final GitHub repository




5. Files transferred and up-to-date in Nova cluster.


[vsc@nova ~]$ tree
.
└── CPRE525Spring2023
    ├── 1. Lab Assignment 2
    │   ├── 1. Lab Assignment 2 Screen Shots for both 2 and 3 steps.pdf
    │   ├── demo_myfuncs.py
    │   └── myfuncs.py
    ├── 2. Lab Assignment 3
    │   ├── 2. Lab Assignment 3 Screen Shots.pdf
    │   └── demo_pythonlist.py
    ├── 3. Lab Assignment 4
    │   ├── 3. Lab Assignment 4 Submission.pdf
    │   ├── demo_myfuncs.py
    │   ├── myfuncs.py
    │   └── __pycache__
    │       └── myfuncs.cpython-39.pyc
    ├── 4. Lab Assignment 5
    │   ├── guass_elimination_solve.py
    │   └── practice_numpyLinearAlgebra.py
    ├── LICENSE
    └── README.md
6 directories, 13 files