

Report for Home work 11:

1&2.

- The main part of the code (*matvector.c*) is written to read the matrix from matrix.output in Aij format and then convert it to CCS (Column Compressed Storage) format as it was described in the class. At the same time, the reordering is done in a loop using index given in problem.
- Matrix-Vector multiplication was done in the routine *matvec()*. (I have run the code in hydra21.eecs.utk.edu)
- To test the code, a vector was considered whose elements are all 2, and the norm was calculated and was compared with the one obtained by MATLAB:

Table 1. Norm comparison.

| Method | Norm |
|---------------------------------------|------------------------|
| MATLAB | 6.991543109915370e+007 |
| matvec() with original matrix | 69885621.393082 |
| matvec() with reordered matrix | 69885621.393082 |

There is a slight difference between the two but they are close.

- MFLOPS using PAPI:

| Method | MFLOPS |
|---------------------------------------|------------|
| matvec() with original matrix | 253.814011 |
| matvec() with reordered matrix | 258.313385 |

The multiplication with reordered matrix should be more efficient (and it is somehow reflected in Table 2) which was expected based on the explanations in the class and the fact that the locality of the data (more dense data) would benefit us in multiplication.

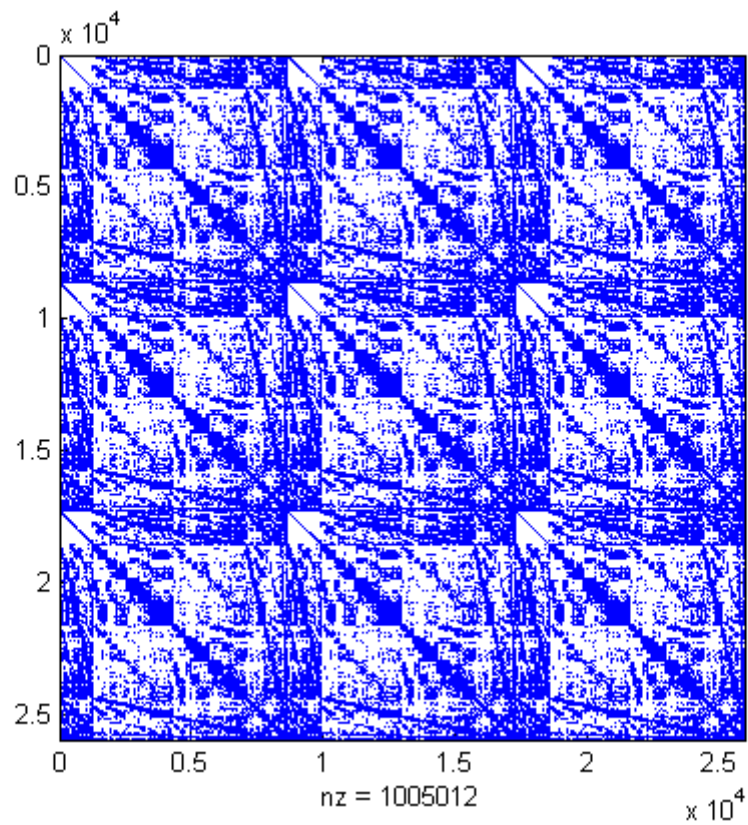
- Plotting the nonzero of the matrices:

Results:

Using the commands:

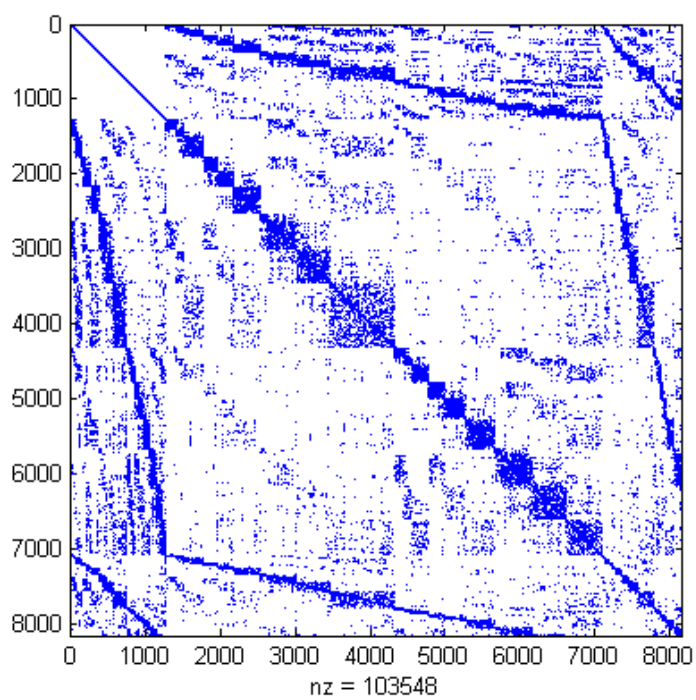
```
load matrix.output
S = spconvert(matrix)
spy(S)
```

Figure 1. Original matrix sparse view.



```
spy(S(1:8166,1:8166));
```

Figure 2. Original matrix sparse second view.



For Reordered Matrix

```
load newmatrix.output  
S = spconvert(newmatrix)  
spy(S)
```

Figure 3. Reordered matrix.

