Project 3, due on 04/11.

Problem 1. For undergraduate students, the assignment is to implementing QR factorization for Upper Hessenberg matrix using Givens rotation.

Step 1: Generate a upper Hessenberg matrix. Do this by modifying the code my_io.c at:

/afs/crc.nd.edu/user/z/zxu2/Public/ACMS40212-S12/col_decomp_mat_vec_multi/data_gen

The current my_io.c code generates a matrix with randomly filled entries. To generate a $M \times M$ Hessenberg matrix, entries below the first lower diagonal need to be zero, i.e., the following piece of code needs to be modified:

```
for(i = 0; i < M; i++)
{
   for(j =0; j < M; j++)
   {
      mat[i][j]=(double)rand()/RAND_MAX;
   }
}
```

Step 2: Use the code

/afs/crc.nd.edu/user/z/zxu2/Public/ACMS40212-S12/col_decomp_mat_vec_multi/data_gen/Givens_QR.c

as the framework to implement your QR algorithm using Givens rotation.

Givens_QR.c has implemented to read in the matrix data from the data file. The read-in matrix data is saved in matA.

Hand-In.

1. The hardcopy of your source code (Also send the source code to me by email. Please use the email title: Project 3: your name).

2. A report which contains results of validation and a description of your algorithm using the pseudo code language.

3 To validate, test if you get back to $H = Q^T R$. H is the original Hessenberg matrix. R is the upper triangular matrix and Q consists of orthonormal columns. Q and R are obtained by your factorization algorithm.

Problem 2. For Graduate Students, the assignment is to implement HouseHolder Arnoldi algorithm.

Step 1: Generate a strictly diagonally dominant matrix. Do this by modifying the code my_io.c at:

/afs/crc.nd.edu/user/z/zxu2/Public/ACMS40212-S12/col_decomp_mat_vec_multi/data_gen

The current my_io.c code generates a matrix with randomly filled entries.

Step 2: Use the code

/afs/crc.nd.edu/user/z/zxu2/Public/ACMS40212-S12/col_decomp_mat_vec_multi/data_gen/Householder_Arnoldi.c

as the framework to implement your QR algorithm using Givens rotation. You can simply use unit vector $\boldsymbol{e}_1$ as $\boldsymbol{v}_1$.

Let the dimension m of the Krylov subspace be much less than the dimension of the matrix. For instance, you can choose a matrix of size 40×40 and let $m = 10$.

Reference: H.F. Walker. Implementation of the GMRES method using Householder transformations. *SIAM. J. on Sci. Comput.* 9:152-163, 1988.

Hand-In.

1. The hardcopy of your source code (Also send the source code to me by email. Please use the email title: Project 3: your name).

2. A report which contains validation of results and a description of your algorithm using the pseudo code language.

3. Validation. Since now you have the orthonormal basis Q of the Krylov subspace $span\{\boldsymbol{r}^{(0)}, A\boldsymbol{r}^{(0)}, A^2\boldsymbol{r}^{(0)}, \dots, A^{m-1}\boldsymbol{r}^{(0)}\}$, generate a vector $\boldsymbol{b}$ in this subspace by doing a linear combination of $\boldsymbol{r}^{(0)}, A\boldsymbol{r}^{(0)}, A^2\boldsymbol{r}^{(0)}, \dots, A^{m-1}\boldsymbol{r}^{(0)}$, then solve $Q\boldsymbol{x} = \boldsymbol{b}$ for $\boldsymbol{x}$ to see if $\boldsymbol{x}$ agrees with coefficients of the linear combination to generate $\boldsymbol{b}$.

.