This project consists of two parts. The first coding part is worth 56 points, the second writing part is worth 48 points. Extra 6 points will be considered for quality of discussion.

You may work in teams of **up to three** persons. Code for your project should be submitted in either Java, Python, or MATLAB (or its free version FreeMat). (Choose only one of these languages to use for the whole project—do not mix and match.) Your final deliverable should be submitted in a single `.zip` archive file. The archive file should be uploaded to the Dropbox on T-Square of *one* of your team members by **11:59 p.m.** on **April 18**. The file should contain:

- A `team_members.txt` file listing the name of each person in the project team.
- A `.doc`, `.docx`, or `.pdf` file for each part of the project you submit, containing the written component of the project.
- A `readme.txt` file for each part of the project you submit, explaining how to execute that part of the project, use your code to read the test files (in the form *.dat), and input the variables. Further, it should tell graders how to read the output results and find where are they.
- For Java or Python users, do not submit a project that has to be run on an IDE. The graders should be able to at least initialize your project on a command line interface, either Linux or Windows.
- The input matrices required in several parts of the project will be `.dat` files in ASCII or text format, that contain double floating point numbers as the matrix entries in the same row separated by spaces or commas; and different rows in the matrix are different rows in the file.

Required Codes

Your code will be inspected manually and run against a test case.

(a) (8 pts) `lu_fact`. A test file `test.dat` will consist of $n \times n$ matrix A of floating-point numbers with linearly independent columns, these numbers contains positive, zero, and negative real numbers. No points will be awarded if the function does not implement the LU-decomposition algorithm.

(b) (16 pts) `qr_fact_house` and `qr_fact_givens`. A test file `test.dat` will consist of $n \times n$ matrix A of floating-point numbers with linearly independent columns, these numbers contains positive, zero, and negative real numbers. No points will be awarded if the function does not implement the specified QR-decomposition algorithm.

(c) `solve_lu`(4 pts) and `solve_qr_house`(2 pts), and `solve_qr_givens`(2 pts) . A test file `test.dat` will consist of $n \times (n+1)$ argument matrix Ab of positive, zero, and negative real floating-point numbers. No points will be awarded if the function use of an inverse matrix algorithm, or if the solution vector is incorrect.

(d) (16 pts) `jacobi_iter` and `gs_iter`. The test case includes a file `test.dat` of $n\times(n+1)$ argument matrix Ab, a file `test_u.dat` of $n \times 1$ vector u as the initial guess, a small positive tolerance number, and a positive integer as maximum iterations. Set $\vec{0}$ as the initial vector. (Please indicate in your documentation what the expected failure value is.) No points will be awarded if the function does not implement the iterative method.

(e) (8 pts) `power_method`. The test case includes a file `test.dat` of $n \times n$ matrix A, a file `test_u.dat` of $n \times 1$ vector u as the initial vector, a file `test_w.dat` of $n \times 1$ vector w as the auxiliary vector, a small positive tolerance number, and a positive integer as maximum iterations. (Please indicate in your documentation what the expected failure value is.) No points will be awarded if the function does not implement the power method.

Writing reports

1 The Hilbert Matrix

- (8 pts) Plot data. Solution and errors for the Hilbert matrix of dimension $n \times n$, $n = 2, 3, \ldots, 20$ and $\vec{b}$ as described in the project. This should be a routine (or set of routines) that calls the appropriate programs and returns a readable outputs as described in the projects for the input n. You should have 3 plots for the errors of $\vec{x}_{sol}$, each obtained with LU and both QR decomposition; one for $||LU - P||_\infty$, and two more for $||QR - P||_\infty$ in each QR-factorization. Your plots will probably look better in a Log scale, and can be grouped in a way that makes comparisons possible.
- (8 pts) Discussion. Writing component should include plots of the errors obtained for the Hilbert matrix as a function of n. The discussion should address the questions posted in the description of the project.

2 Convergence of the iterative methods

I. (12pts)

   – Average these 100 solutions $\vec{x}_N$ to obtain an approximation solution $\vec{x}_{approx}$ and compute its error with exact solution $\|\vec{x}_{approx} - \vec{x}_{exact}\|_\infty$. Average these 100 ratios $N_{Jacobi}/N_{Gauss-Seidel}$ to obtain the multiple of the steps between two methods. Plot 100 points on a colored scatterplot, the x-axis is $\|\vec{x}_0 - \vec{x}_{exact}\|_\infty$ and y-axis is N. Use black scatters for Jacobi results, and blue for Gauss-Seidel results. All display specifications from the project description should be met (2 scatterplots, correct axes, colored data points, etc.)

- Discussion: The accuracy, completeness, and clarity of your writing will be considered in assigning credit. Be sure to discuss: 1. the effect of initial vector position, 2. the steps needs in both methods. [Hint: the ratio of $N_{Jacobi}/N_{Gauss-Seidel}$ seems to be a fixed number, it's related to eigenvalues of the iteration matrix $S^{-1}T$]. 3. the graphs.

II. (4 pts)

- Plot two colored scatterplots in a graph, the x-axis is M and y-axis is the corresponding errors. Use black scatters for Jacobi results, and blue for Gauss-Seidel results. Your plots will probably look better in a Log scale, and can be grouped in a way that makes comparisons possible. All display specifications from the project description should be met (2 scatterplots, correct axes, colored data points, etc.)
- Discussion: Interpret how the error be amplified, and why both method does't work for this matrix. (it should be related to the maximum eigenvalue of the matrix $S^{-1}T$ in the view of absolute value.)

3 Convergence of the Power Method

I. (12 pts)

- All display specifications from the project description are met (2 scatterplots, correct axes, colored data points, etc.) The data shown is generated in the manner set out.
- Discussion: 1. The general shape of the scatterplots. It would be good to talk the mathematical reasons behind how the two plots are related. [Hint: Show that trace(A^{-1}) = trace(A) / determinant(A). This fact may be useful for talking about the relationship between the two plots.] 2. The relationship between the position of a matrix on the plot and the number of iterations needed in the power method. [Hint: Let λ_1 be the larger eigenvalue of A in magnitude and λ_2 the smaller in magnitude. Consider the ratio $\|\lambda_1/\lambda_2\|$: how is it related to the power method?]

II. (4pts) Dissuasion: Interpret your results why we picked these two special p_1 and p_2, and why the largest eigenvalue of $(A-p_2I)^{-1}$ converges to $(3-p_2)^{-1}$. (it should related to the distant between p_2 and eigenvalues.)