Part 1: *Hilbert Matrix: QR, LU and error calculations*

Aaron Andrews, Patrick Tam, Jinsong Han

Math 2605 Calculus 3 for Computer Science
Georgia Tech Institute of Technology

March 31st, 2015

1 Introduction

A Hilbert matrix H is a square matrix of size nxn with unit fractions as entries. For any size n matrix, an individual element $H_{ij} = \frac{1}{i+j-1}$.
The goal of this part of the assignment was to computationally calculate and compare the decomposition methods QR (both Givens and Householder's methods) and LU over a series of Hilbert matrices of increasing size n (from $n = 2$ to $n = 20.$) The methods are compared by plotting the error of the solution as well as the error of the matrices (formed by QR and LU) for each method.
Solution error is calculated by finding the maximum norm of a solution b subtracted from the hilbert matrix H multiplied by solution $\overline{x}$:

$$||H\overline{x} - \overline{b}||_{\infty}$$

Where b is a vector of n rows, with an individual element $\bar{b}_i = 0.1^{\frac{n}{3}}(1, 1, ..., 1)^t$

The error of the decomposition itself is similarly found by taking the maximum norm of the hilbert matrix subtracted from the decomposition:

$$||QR - H||_\infty or ||LU - H||_\infty$$

Graphs

1.1 QR with HouseHolder's Method

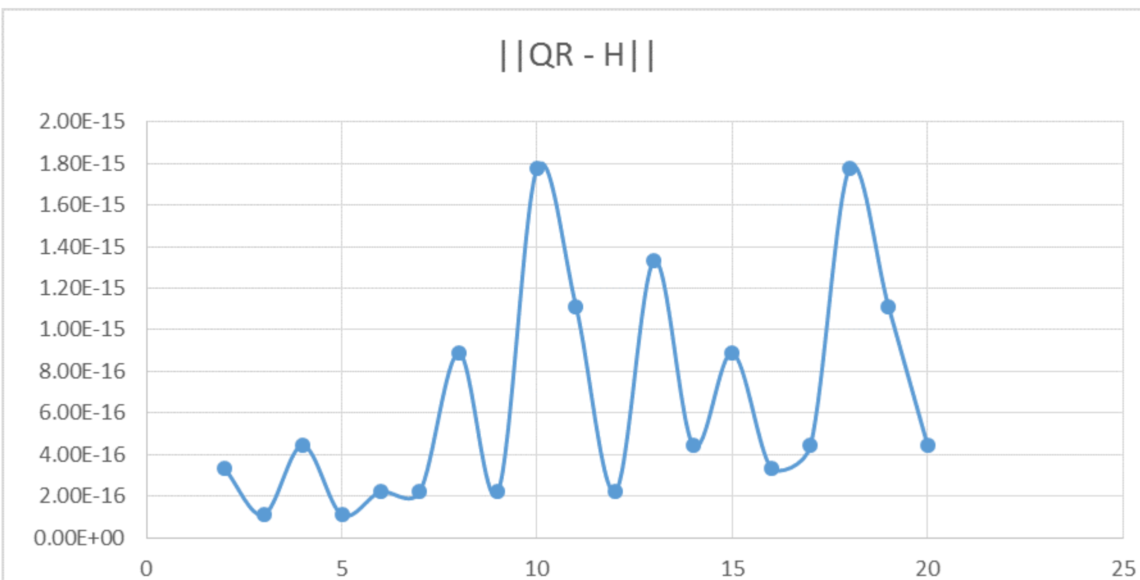

Figure 1: The error of the QR decomposition through Householders' method as a function of the size of the Hilbert matrix

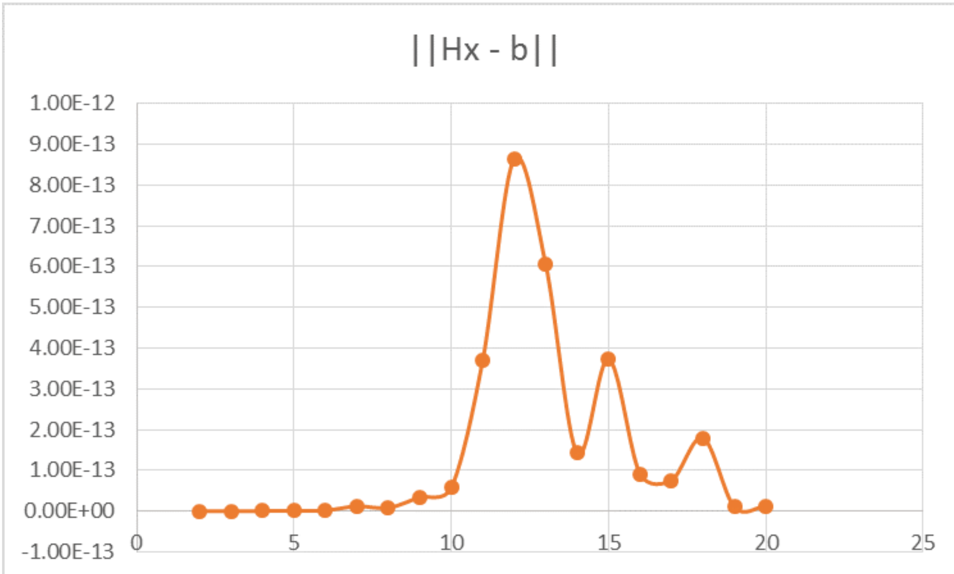

Figure 2: The error of the solution found by using a QR decomposition through Householders' method as a function of the size of the Hilbert Matrix

1.2 QR with Givens Method

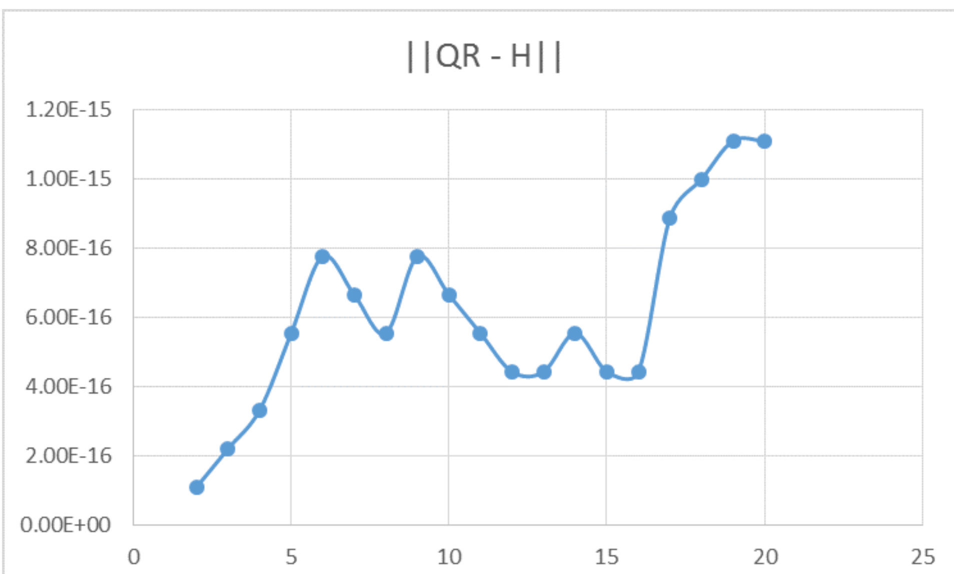

Figure 3: The error of the QR decomposition through Givens method as a function of the size of the Hilbert matrix

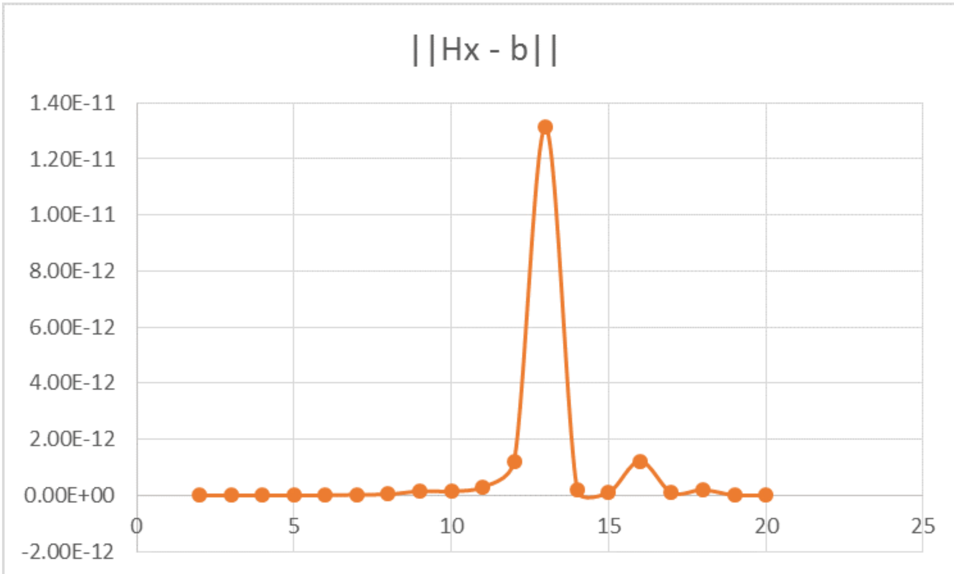

Figure 4: The error of the solution found by using a QR decomposition through Givens method as a function of the size of the Hilbert Matrix

1.3 LU

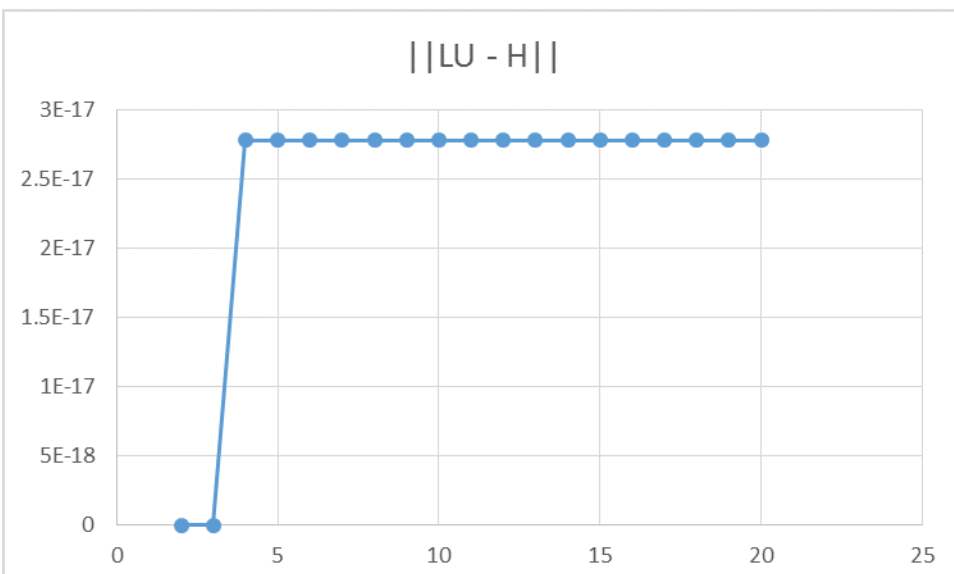

Figure 5: The error of the LU decomposition of the Hilbert Matrix as a function of its size

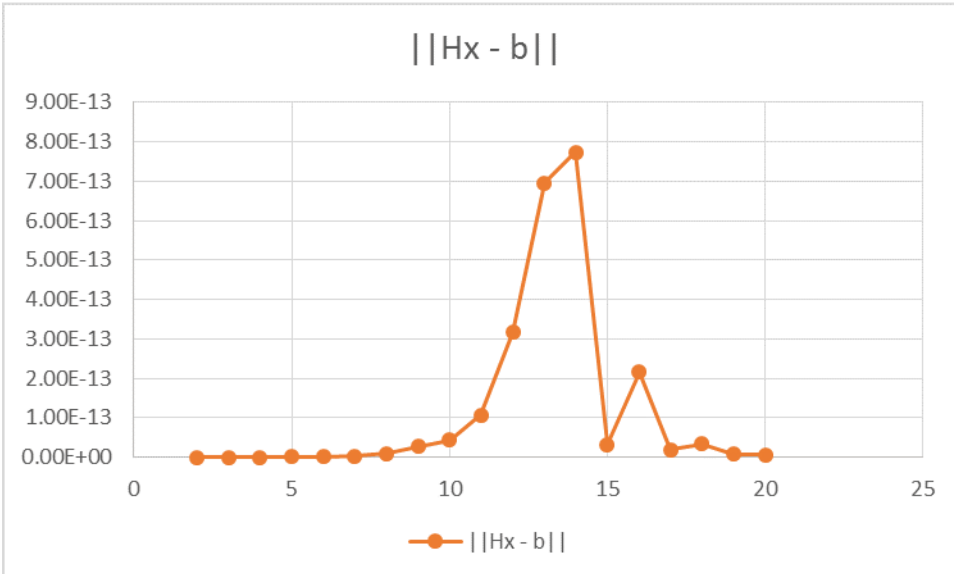

Figure 6: The error of the solution found by using LU decomposition as a function of the size of the Hilbert Matrix

2 Discussion

The benefit and justification of using LU or QR factorizations to solve linear systems of equation is a matter of efficiency and numerical stability, respectively.

2.1 Efficiency

2.1.1 QR

For:

$$A\overline{x} = \overline{b}$$

Since A = QR, we have the equivalent equation:

$$QR\overline{x} = \overline{b}$$

Since Q is made up of orthogonal unit vectors, we can use the property $Q^TQ = I$ to simplify the equation to:

$$R\overline{x} = Q^T\overline{b}$$

From this point on, we could try solving for x by using the inverse matrix algorithm to find R^{-1} and multiply it by $Q^T\overline{b}$, but that is completely unnecessary. That is because the matrix R is defined to be an upper triangular matrix of the form:

$$\begin{pmatrix} X & X & X \\ 0 & X & X \\ 0 & 0 & X \end{pmatrix}$$

This form allows us to simply perform a backward substitution algorithm to find the solution $\overline{x}$. Such an algorithm has only a worst case of $O(n^2)$, because for a matrix of n columns you have to visit n rows. This is relatively much more efficient than the inverse matrix algorithm. Unfortunately, while ultimately solving for $\overline{x}$ is an efficient process with a given R, the slow process of actually finding Q and R (whether using Givens or Householders') sacrifices efficiency for the benefit of numeric stability, and can altogether be even more inefficient than Gaussian Reduction . In terms of speed, QR is less preferable to that of LU and the inverse matrix algorithm.

2.1.2 LU

For:

$$A\overline{x} = \overline{b}$$

Since A = LU, we get the equivalent equations:

$$Ly = b$$

and

$$Ux = y$$

Since L is defined to be a lower-triangular matrix and U, upper-triangular, respectively, finding $\overline{x}$ is as simple as performing forward substitution to find y and then backwards substitution to find $\overline{x}$. Compared to performing a Gaussian reduction to find the inverse, this takes far less operations. Since both forward and backward substitution is required, LU might not seem more efficient as QR, but when taking into to account the process of finding LU, it requires far less operations. Unfortunately, while LU is a very fast algorithm, and generally more stable than using the inverse matrix, it is not as stable as using the QR algorithm.

2.2 Numerical Stability

While Using QR to solve a linear system is less efficient than both LU and the standard inverse matrix algorithm, its true strengths lie in its numeric stability in comparison to both LU decomposition and the inverse matrix formula.

In the case of Linear Algebra, A numerically stable algorithm can be defined as one that for any change in the input $\overline{x}$ (by even a matter of a few floating points), it results in a minimal change in output $\overline{b}$. In other words, numerical stability means minimum error.

We can evidently see that the QR algorithm is more stable than LU by looking at the $||Hx - b||$ graphs in Figure 4 (Givens) and Figure 5, respectively. There is an especially visible difference starting at around $n = 10$ of the graphs where we can see a difference in the shapes of the line. In Fig 6, the slope of LU's solution error seems to change erratically while QR's change in solution error stays far more consistent and changes less times. Looking at the graphs of QR, the numeric stability of the algorithm is not exactly evident, especially at $n = 14$. But is important to remember that Hilbert Matrices are generally ill-conditioned, yielding extremely high condition numbers. Using the QR algorithm can only provide so much numerically stability.