**Why is it justified to use the LU or QR-factorizations as opposed of calculating an inverse matrix?**

Calculating the inverse of a matrix without either of these methods is a tedious task which, when using Gauss – Jordan elimination/row reduction, requires O(n4) calculations (since each forward elimination and backward substitution must be done for each system of equations), so using either LU or QR decomposition greatly speeds up calculation time by requiring fewer operations to be done. For LU decomposition, once L and U have been calculated (requires O(n3) calculations), finding the inverse is a simple matter of forward and backward substitution “n” times with the identity matrix (requires O(n2) calculations). The entire process of finding the inverse of a matrix using LU decomposition is then done in O(n3) calculations. For QR decomposition, once Q and R have been calculated (requires O(n3) calculations), the inverse is calculated via R-1 \* Qt, which is a simple calculation since R is an upper triangular matrix (requires O(n2) calculations). So the entire process of finding the inverse of a matrix using QR decomposition is also done in O(n3) calculations.

**What is the benefit of using LU or QR-factorizations in this way?**

Error amplification in finding the inverse of a matrix using Gauss – Jordan elimination/row reduction can be unstable because it can greatly amplify any error already present in the first step. One can use attempt to reduce the error that results from this method by having the largest value of each column at or below the diagonal be a pivot in the matrix (since this way, the row operations have to use some multiplier less than one when row reducing), and this idea can also be applied to LU decomposition. But, QR decomposition provides a much better alternative than either of these methods. QR decomposition gives a huge benefit to calculating the inverse in terms error amplification. Given a matrix A whose inverse is being calculated,

A = Q \* R

A-1 = R-1 \* Q-1

||A-1|| = ||R-1|| \* ||Q-1||

Because Q is an orthogonal matrix, its inverse is equal to Qt. Therefore, the condition number is calculated by the following:

cond(Q) = σmax / σmin = √(λmax(Qt \* Q)) / √(λmin(Qt \* Q)) = 1

Therefore, there is no error added to our calculations involving Q. So in terms of the condition number, cond(A) = cond(R). No error, other than the inherent error in the original problem, is now present in our calculations. And the computation of QR, when using Householder reflections or Givens rotations, doesn’t add any error as well since Q­­n \* … \* Q1 \* A = R, and we just established that the condition number of Qn is equal to 1. This method of calculating an inverse is considered to be very stable.