CS206 Assignment #4 - The Dreaded Hilbert Matrix

We are finally going to use Matlab in this assignment. You are welcome to purchase a student copy ($50) for yourself, but you can also use Matlab on the openlab machines for free—you may even be able to get a free copy of Matlab from the UCI NACS/OIT website.

In this assignment, we're going to generate a bunch of random matrices of size $n \times n$ for various n, and try to solve them, then compute the residuals.

1. Let $n = 5$, and let H_5 be the *Hilbert Matrix* of order 5 (that is, the 5×5 Hilbert matrix). You should read about it a bit on Wikipedia, but Matlab has a built-in function that provides the Hilbert matrix, called `hilb(`n`)`. Interestingly, although the Hilbert matrix is hard to represent exactly for large n, its *inverse* has an entirely integer representation, and thus can be computed and represented *exactly* up to about $n = 15$ or so; in Matlab you get this matrix by calling `invhilb(`n`)`. Compute the condition number of H_5. Then, compute $H_5^{-1} \times H_5$, where H_5^{-1} is the exact inverse Hilbert matrix returned by `invhilb(5)`. Since the exact value of $H^{-1}H$ should be the $n \times n$ identity matrix I_n, we can check how close we are to the correct answer by computing $(H_5^{-1} \times H_5 - I_n)$, and then taking its norm (use the 2-norm, which is the default). Try that, and then use the `inv` function in Matlab to compute the inverse of H_5 instead of `invhilb(`n`)`, and perform the same norm computation. How do they compare?

2. Now, rather than just $n = 5$, do the same thing for n going from 1 to 30 **using single precision arithmetic**. Create a table showing n, the condition number of H_n, the norm of $(H_n^{-1}H_n - I_n)$ (where H_n^{-1} is the exact inverse Hilbert matrix), and the norm of (`inv`$(H_n) * H_n - I_n$). Comment on how the error increases in the two cases. Hint: the former is a half-decent estimate on the actual error in the multiplication, and the latter is merely the residual. Does the behavior change significatly if you reverse the order (ie., ($H_n *$`inv`$(H_n) - I_n$) instead of (`inv`$(H_n) * H_n - I_n$))?

3. Repeat the above in double precision.

Bonus (worth at most 10% of the full assignment) Repeat the single-precision version using one or more iterative methods of your choice to perform the matrix inversion. Do any of them do better?

4. The Hilbert matrix is a pathologically bad matrix, with a very bad condition number. Most matrices are not really that bad. For each n going from 1 to 30, generate 1,000 random matrices with entries in the range [0,1] (again, there is a built-in matlab function to generate random matrices, just use it), and do something similar to the above, except create the table with the following columns: n; the mean, min, and max condition number observed across your 1,000 samples; and the mean, min, and max 2-norm of $(M * inv(M) - I_n)$ and $(inv(M) * M - I_n)$. If the table is too wide, you can print it in landscape mode.

You should submit a PDF write-up electronically.