Worksheet 5 (March 31st, 2021)

1 Pseudo-inverse for ill-conditioned systems

We will revisit here square linear systems $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. A classic example of a very ill-conditioned $n \times n$ matrix $\boldsymbol{A}$ is the Hilbert matrix, defined by

$$a_{ij} = \frac{1}{i+j-1}.$$

Note: *This problem is* **not** *about the Hilbert matrix per se, but about ill-conditioned matrices in general, so do not focus on this specific matrix (one reason we are using it is that we will encounter it again in a later homework on polynomial approximation). We will later talk about the Vandermonde matrix (related to polynomial interpolation) which is another example of an ill-conditioned matrix that you could use equally well.*

1.1 Conditioning numbers

Form the Hilbert matrix in MATLAB and compute the conditioning number $\kappa_2(\boldsymbol{A}) = \|\boldsymbol{A}\|_2 \|\boldsymbol{A}^{-1}\|_2$ for increasing size of the matrix $n = 10, 12, 14, 15, 16$ using MATLAB's $cond$ function (which by default uses the L_2 norm), and compare to the built-in function $rcond$, which estimates the *inverse* of the conditioning number in the L_1 norm in a rapid but approximate way. For what n does the Hilbert matrix become too ill-conditioned for double precision floating-point arithmetic?

From now on set $n = 15$.

1.2 Solving ill-conditioned systems

Compute a right-hand side (rhs) vector $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}$ so that the exact solution is $\boldsymbol{x} = [1, 1, \ldots, 1]$ (all unit entries). Solve the linear system using MATLAB's built-in solver and report the error δx in the approximate solution $\hat{\boldsymbol{x}}$ (for example, in the Euclidian norm, $\delta x = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2$). How many digits of accuracy do you get in the solution $\boldsymbol{x}$? Do your results conform to the theoretical expectation discussed in class?

Also compute and report the relative norm of the residual $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\| / \|\boldsymbol{b}\|$ and comment on your observations.
Note: A method is called *backward stable* if it computes the *exact* solution to a nearby problem, i.e., if the residual is small.

1.3 Solving systems using the SVD

For this section start by using the built-in function $pinv$ if you cannot quickly write your own version, and then come back at the end to writing your own pseudo inverse function (validate it against the built-in function). However, it will be crucial to read the documentation $pinv$ to set properly the "tolerance" parameter (see below)!

Compute the SVD decomposition of $\boldsymbol{A}$. Look at the singular values of $\boldsymbol{A}$ and compute the conditioning number of $\boldsymbol{A}$ based on this [*Hint: The MATLAB function $diag$ can be used to extract the diagonal of a matrix or to construct a diagonal matrix*].

Construct the matrix pseudo-inverse $\boldsymbol{A}^{\dagger}$ from the SVD or using $pinv$. Use the pseudo-inverse to compute the solution $\hat{\boldsymbol{x}} = \boldsymbol{A}^{\dagger}\boldsymbol{b}$, and see if this is any more accurate than the previous direct solution.

1.3.1 Regularized Pseudo-Inverse

For a given relative tolerance $\varepsilon \ll 1$, a modified or regularized pseudo-inverse $\hat{\boldsymbol{A}}^{\dagger}$ is obtained by first setting to zero all singular values that are smaller than $\varepsilon\sigma_1$, where σ_1 is the largest singular value. This can be obtained in MATLAB using the built-in function $pinv$ as $\hat{\boldsymbol{A}}^{\dagger} = pinv(A, \, \varepsilon\sigma_1)$.

For several logarithmically-spaced tolerances (for example, $\varepsilon = 10^{-i}$ for $i = 1, 2, \ldots, 16$), compute the modified pseudo-inverse and then a solution $\hat{\boldsymbol{x}} = \hat{\boldsymbol{A}}^{\dagger}\boldsymbol{b}$. Plot the relative error in the modified solution versus the tolerance on a log-log scale. You should see a clear minimum error for some $\varepsilon = \tilde{\varepsilon}$. Report this optimal $\tilde{\varepsilon}$ and the smallest error that you can get.