Petter Frisvåg

The Hilbert matrix $H \in \mathbb{R}^{n \times n}$ is a dense, square matrix with indices given by

$$H_{ij} = \frac{1}{i+j-1}, \quad i, j = 1, \ldots, n.$$

It is notorious for its large condition number, resulting in instabilities and slow convergence in many numerical algorithms. It naturally arises in regression, where the Hilbert space constructed from the monomial basis on the line segment $(0, 1)$ results in the inner product

$$\langle x^{i-1}, x^{j-1} \rangle = \int_0^1 x^{i-1} x^{j-1} \mathrm{d}x = \frac{1}{i+j-1} = H_{ij}$$

of the basis vectors. The Hilbert matrix is a good example of how run-off-the-mill matrices from real-world applications can be ill-conditioned, even though they are invertible. In other words, these kind of matrices are interesting not only from a theoretical point of view.

The program `cg.cpp` uses the conjugate gradient method with CBLAS to solve the system

$$H\mathbf{x} = \mathbf{b},$$

where $\mathbf{b}^T = (1, \ldots, 1) \in \mathbb{R}^n$, for some provided n.

Although the conjugate gradient method in theory should find the exact solution in no more than n steps, the numerical round-off errors due to the ill-conditioning of the Hilbert matrix results in the algorithm to use significantly more steps in order to reach the desired accuracy, especially when n increases. For the exact version of the CG method used, see Algorithm 5.2 in [2].

References

[1] Hilbert matrix. `https://en.wikipedia.org/wiki/Hilbert_matrix`.

[2] Jorge Nocedal and Stephen Wright. *Numerical Optimization (Springer Series in Operations Research and Financial Engineering)*. Springer, 2006.