Report on Numerical Linear Algebra Coding Assignment

Chapter 2: Error Analysis for Linear Systems

Hongyi Zhang, 1500017736

October 29, 2017

1 L^∞ condition number estimation

1.1 Algorithm

The estimation of the L^∞ condition number here relies on the following procedure for L_1 norm estimation of a given matrix using an optimization method.

Algorithm 1 Estimation of L_1 norm of a matrix $B \in \mathbb{R}^{n\times n}$

Initialize $x \in \mathbb{R}^n$ with $\|x\| \le 1$
while true **do**
  $w = Bx$; $z = B^T \mathrm{sgn}(w)$;
  if $\|z\|_\infty = z^T x$ **then return** $\|w\|_1$
  else $x = e_j$ where $j = \arg\max_j |z_j|$

To estimate $\kappa_\infty(A) = \|A^{-1}\|_\infty \|A\|_\infty$, it is sufficient to first compute $\|A\|_\infty$ directly and then compute $\|A^{-1}\|_\infty = \|A^{-T}\|_1$ by applying the algorithm described above to $B = A^{-T}$. The matrix-vector multiplication step will come down to solving linear systems, i.e.

$$w = Bx \iff A^T w = x,$$
$$z = B^T \mathrm{sgn}(w) \iff Az = \mathrm{sgn}(w).$$

1.2 Numerical results

Now we are going to estimate the L^∞ condition number of the Hilbert matrix $H_n = (\frac{1}{i+j-1})_{n\times n}$ for $5 \le n \le 20$. Estimations are obtained by two methods, one of which employs

Table 1: L^∞ condition number estimation of the Hilbert matrix

n	Algorithm	Definition	n	Algorithm	Definition
5	9.436560e+05	9.436560e+05	13	4.629299e+17	3.728459e+18
6	2.907028e+07	2.907028e+07	14	1.371221e+19	1.359436e+18
7	9.851949e+08	9.851949e+08	15	1.124840e+18	1.764797e+18
8	3.387279e+10	3.387279e+10	16	1.344279e+18	8.916602e+17
9	1.099651e+12	1.099650e+12	17	1.971367e+18	1.763346e+18
10	3.535378e+13	3.535389e+13	18	9.128242e+19	2.692657e+18
11	1.230619e+15	1.231125e+15	19	3.395702e+19	1.317108e+18
12	3.831669e+16	3.947297e+16	20	3.162800e+18	6.323288e+18

the algorithm above to compute $\|A^{-T}\|_1$, while the other directly computes $\|A^{-1}\|_\infty$ by first inverting the matrix and then computing the norm. The two methods are denoted by "algorithm" and "definition" respectively, and the results are displayed in Table 1.

As expected, the Hilbert matrix has an extremely large condition number and is thus highly ill-conditioned. For $n \le 10$, the algorithm method and the definition method achieve similar results, identical up to 5 significant digits, which proves the feasibility of the optimization technique applied to norm estimation. For $n > 10$, however, two methods start to largely disagree with each other. The algorithm method suffers from solving linear systems, while the definition method suffers from inverting the matrix. Therefore, the ill-condition of H_n makes their output both less reliable, and it becomes essentially hard to tell a appropriate condition number for a high-order Hilbert matrix.

2 Accuracy estimation of a numerical solution

2.1 Problem background

Let

$$A_n = \begin{bmatrix} 1 & 0 & \cdots & 0 & 1 \\ -1 & 1 & \cdots & 0 & 1 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ -1 & \cdots & -1 & 1 & 1 \\ -1 & \cdots & -1 & -1 & 1 \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

Pick $x \in \mathbb{R}^n$ randomly and compute $b = A_n x$. Solve this equation using Gaussian elimination with partial pivoting, and estimate the accuracy of the numerical solution $\tilde{x}$ for $5 \le n \le 30$. Compare the estimation with the actual relative error.

2.2 Theoretical foundation

Denote the residual $r = A\tilde{x} - b$ for the numerical solution. Combined with $0 = Ax - b$, we obtain $r = A(\tilde{x} - x)$, which gives

$$\|\tilde{x} - x\| = \|A^{-1}r\| \le \|A^{-1}\|\|r\|.$$

Note that $\|b\| \le \|A\|\|x\|$, and hence

$$\frac{\|\tilde{x} - x\|}{\|x\|} \le \|A^{-1}\|\|A\|\frac{\|r\|}{\|b\|} = \kappa(A)\frac{\|r\|}{\|b\|}. \tag{*}$$

In particular, let $\|\cdot\|$ be the L^∞ norm, then the relative error can be estimated by computing the norm of r and b, along with the L^∞ condition number that can be computed with the methods described in Section 1. Accuracy estimation can be used for an adaptive stopping criteria for an iterative solving routine.

2.3 Numerical results

The accurate solution $x \in \mathbb{R}^n$ is uniformly picked from $[-100, 100]^n$. Let $b = Ax$, and the numerical solution $\tilde{x}$ is obtained by solving it using Gaussian elimination with partial pivoting. The estimated error $\kappa(A)\|r\|/\|b\|$ and the accurate error $\|\tilde{x} - x\|/\|x\|$ are shown in Table 2, for $5 \le n \le 30$.

From Table 2, we can discover that the estimated error, as a reliable upper bound for the accurate error, is larger than the latter up to almost one order of magnitude. Fortunately, the problem is well-conditioned and the actual relative error is fairly small. Therefore, our estimation of the relative error is equal to a proper approximation. On the other hand, the error can be found to increase in magnitude as the size of the matrix n grows. It can be seen as an indication of error accumulation as more and more steps are taken and every step brings some error at each time.

Table 2: Accuracy estimation performance

n	Estimated error	Accurate error	n	Estimated error	Accurate error
5	9.921186e-16	7.658544e-16	18	3.109589e-11	2.348967e-12
6	7.494871e-16	2.137463e-16	19	8.604493e-11	3.223932e-12
7	1.174724e-14	3.808383e-15	20	1.035603e-10	5.350113e-12
8	3.410737e-15	1.195279e-15	21	1.362274e-10	1.421602e-11
9	1.511149e-14	2.282968e-15	22	1.133490e-10	5.181549e-12
10	1.330775e-14	3.893567e-15	23	4.490218e-10	6.180382e-11
11	4.917789e-14	3.066405e-15	24	3.660310e-09	1.780980e-10
12	8.881603e-14	1.511591e-14	25	1.715777e-09	9.322562e-11
13	8.904718e-13	7.956278e-14	26	4.663560e-09	4.920735e-10
14	3.221822e-13	3.288204e-14	27	4.843467e-08	1.780726e-09
15	3.820799e-12	4.626574e-13	28	1.728316e-08	2.639623e-09
16	3.114746e-12	7.985526e-13	29	4.163118e-08	4.924608e-09
17	1.692042e-11	1.031362e-12	30	3.051060e-08	1.504584e-09