Japan J. Indust. Appl. Math., 26 (2009), 249–277 Area ⟨2⟩

Inversion of Extremely Ill-Conditioned Matrices in Floating-Point

Siegfried M. Rump

Institute for Reliable Computing, Hamburg University of Technology
Schwarzenbergstraße 95, Hamburg 21071, Germany, and
Visiting Professor at Waseda University, Faculty of Science and Engineering
3–4–1 Okubo, Shinjuku-ku, Tokyo 169–8555, Japan
E-mail: rump@tu-harburg.de

Received March 17, 2008
Revised December 5, 2008

Let an $n \times n$ matrix A of floating-point numbers in some format be given. Denote the relative rounding error unit of the given format by $\texttt{eps}$. Assume A to be extremely ill-conditioned, that is $\mathrm{cond}(A) \gg \texttt{eps}^{-1}$. In about 1984 I developed an algorithm to calculate an approximate inverse of A solely using the given floating-point format. The key is a multiplicative correction rather than a Newton-type additive correction. I did not publish it because of lack of analysis. Recently, in [9] a modification of the algorithm was analyzed. The present paper has two purposes. The first is to present reasoning how and why the original algorithm works. The second is to discuss a quite unexpected feature of floating-point computations, namely, that an approximate inverse of an extraordinary ill-conditioned matrix still contains a lot of useful information. We will demonstrate this by inverting a matrix with condition number beyond 10^{300} solely using double precision. This is a workout of the invited talk at the SCAN meeting 2006 in Duisburg.

Key words: extremely ill-conditioned matrix, condition number, multiplicative correction, accurate dot product, accurate summation, error-free transformations

1. Introduction and previous work

Consider a set of floating-point numbers $\mathbb{F}$, for instance double precision floating-point numbers according to the IEEE 754 standard [3]. Let a matrix $A \in \mathbb{F}^{n \times n}$ be given. The only requirement for the following algorithm are floating-point operations in the given format. For convenience, assume this format to be double precision in the following.

First we will show how to compute the dot product $x^{\mathrm{T}}y$ of two vectors $x, y \in \mathbb{F}$ in K-fold precision with storing the result in one or in K floating-point numbers. This algorithm to be described in Section 2 uses solely double precision floating-point arithmetic and is based on so-called error-free transformations [7, 13, 12]. The analysis will show that the result is of a quality "as if" computed in K-fold precision.

The relative rounding error unit in IEEE 754 double precision in rounding to nearest is $\texttt{eps} = 2^{-53}$. Throughout the paper we assume that no over- or underflow occurs. Then every single floating-point operation produces a result with relative error not larger than $\texttt{eps}$.

This research was partially supported by Grant-in-Aid for Specially Promoted Research (No. 17002012: Establishment of Verified Numerical Computation) from the Ministry of Education, Science, Sports and Culture of Japan.