Numerically Stable Real-Number Codes Based on Random Matrices

Zizhong Chen
Innovative Computing Laboratory
Computer Science Department
University of Tennessee
zchen@cs.utk.edu

Jack Dongarra
Innovative Computing Laboratory
Computer Science Department
University of Tennessee
Computer Science and Mathematics Division
Oak Ridge National Laboratory
dongarra@cs.utk.edu

Abstract — **Error correction codes defined over real-number and complex-number fields have been studied and recognized as useful in many applications. However, most real-number and complex-number codes in literature are quite suspect in their numerical stability. In this paper, we introduce a class of numerically stable real-number and complex-number codes that are based on random generator matrices over real-number and complex-number fields.**

Key Words: Error Correction Codes, Real-Number Codes, Random Matrices, Condition Number, Numerical Stability.

I. Introduction

Error correction codes are often defined over finite fields. However, in some applications, error correction codes defined over finite fields do not work. Instead, codes defined over real-number and complex-number fields have to be used to detect and correct errors. For example, in algorithm-based fault tolerance [4] [12] [13] [15] and fault tolerant dynamic systems [10], to provide fault tolerance in computing, data are first encoded using error correction codes and then algorithms are redesigned to operate (using floating point arithmetic) on the encoded data. Due to the impact of the floating point operation on the binary representation of these encoded data, codes defined over finite fields do not work. But codes defined over real-number and complex-number fields can be used to protect errors in computing by taking advantage of certain relationships, which only exist when real-number (or complex-number) codes are used, among the output data of the redesigned algorithms.

However, most real-number and complex-number codes in literature are quite suspect in their numerical stability. Error correction procedures in most error correction codes involve solving linear system of equations. In computer real-number and complex-number arithmetic where no computation is exact due to round-off errors, it is well known [9] that, in solving a linear system of equations, a condition number of 10^k for the coefficient matrix leads to a loss of accuracy of about k decimal digits in the solution. In the generator matrices of most existing real-number and complex-number codes, there exist ill-conditioned sub-matrices. Therefore, in these codes, when certain error patterns occur, an ill-conditioned linear system of equations has to be solved in the error correction procedure, which can cause the loss of precision of possibly all digits in the recovered numbers.

The numerical issue of the real-number and complex-number codes has been recognized and studied in some literature. In [4], Vandermonde-like matrix for the Chebyshev polynomials was introduced to relieve the numerical instability problem in error correction for algorithm-based fault tolerance. In [7] [8] [11] [14], the numerical properties of the Discrete Fourier Transform codes were analyzed and methods to improve the numerical properties were also proposed. To some extent, these efforts have alleviated the numerical problem of the real-number and complex-number codes. However, they did not obtain the numerical stability we achieve in this paper.

In this paper, we introduce a class of numerically stable real-number and complex-number codes that are based on random generator matrices over real-number and complex-number fields. This paper is organized as follows: Section II specifies the error correction problem we focus on. Section III discusses the impact of round-off errors on error correction. A numerical example illustrating how most existing real-number and complex-number codes may fail to correct errors is given in Section IV. In Section V, we first study the properties of random matrices and then introduce our real-number and complex-number codes. In Section VI, our codes are compared with most existing real-number and complex-number codes.

II. Problem Specification

Let $x = (x_1, x_2, ..., x_N)^T \in \mathcal{C}^N$ denote the original information, and $y = (y_1, y_2, ..., y_M)^T \in \mathcal{C}^M$, where $M = N + K$, denote the encoded information with redundancy. G is a M by N real or complex matrix. The original information x and the encoded information y are related through

$$y = Gx. \tag{1}$$

Our problem is: how to choose the matrix G such that, after any no more than K erasures in the elements of the encoded information y, a good approximation of the original information x can still be reconstructed from y?

We stress here once more that, in some fault tolerant computing applications, error correction codes defined over finite fields

This research was supported in part by the Applied Mathematical Sciences Research Program of the Office of Mathematical, Information, and Computational Sciences, U.S. Department of Energy under contract DE-AC05-00OR22725 with UT-Battelle, LLC

do not work and real-number (or complex-number) codes have to be used to correct errors. In this paper, we are discussing real-number and complex-number codes. Therefore, in this problem specification, all arithmetic operations are over real-number (or complex-number) field.

III. The Impact of Finite Precision Arithmetic on Error Correction in Real and Complex Fields

Assume there are at most K elements of y lost, then there are at least N elements of y available. Let J denote the set of indexes of any N available elements of y. Let y_J denote a sub-vector of y consisting of the N available elements of y whose indexes are in J. Let G_J denote a sub-matrix of G consisting of the N rows whose indexes are in J. Then, from (1), we can get the following relationship between x and y_J:

$$y_J = G_J x. \tag{2}$$

When the matrix G_J is singular, there are infinite number of solutions to (2). But, if the matrix G_J is non-singular, then (2) has one and only one solution, which is the original information vector x. Therefore, in order to be able to recover the original information x, the sub-matrix G_J has to be non-singular.

However, due to the finite precision representation of the real and complex numbers in the computational hardware, there are round-off errors in almost all calculations. Therefore, unlike in finite-field codes, in real-number and complex-number codes, we could only get an approximation $\widehat{x}$ of the original x by solving (2).

The accuracy of the reconstructed $\widehat{x}$ can be measured by the relative error

$$\frac{\| x - \widehat{x} \|_2}{\| x \|_2}.$$

If the relative error is 10^{-i}, then we say the solution $\widehat{x}$ is accurate to i decimal digits [1].

The principal methods for solving (2) are Gaussian elimination with partial pivoting and QR factorization. From [16], we know that the relative error of the computed solution $\widehat{x}$ can be bounded by

$$\frac{\| x - \widehat{x} \|_2}{\| x \|_2} \leq \kappa f(n) \epsilon,$$

where ϵ denotes the machine epsilon.

In practice, the function $f(n)$ is often small and can be ignored. Therefore, in solving (2), a condition number of 10^i for G_J leads to a loss of accuracy of approximately i decimal digits in the solution $\widehat{x}$. Hence, in order to reconstruct a good approximation of the original information x, G_J has to be well-conditioned.

Actually, for any N by N sub-matrix G_J of G, there is a erasure pattern of y which requires to solve a linear system with G_J as the coefficient matrix to reconstruct an approximation of the original x.

Therefore, to guarantee that a reasonably good approximation of x can be reconstructed after any no more than K erasures in y , the generator matrix G must satisfy: *any N by N sub-matrix of G is well-conditioned*.

IV. Existing Codes in Literature

In this section, we briefly review some commonly used codes in literature and give an example to illustrate how these codes may fail to reconstruct a reasonably good approximation of the original information.

In the commonly used real-number and complex-number codes in literature, the generator matrices include: Vandermonde matrix (Vander) [10], Vandermonde-like matrix for the Chebyshev polynomials (Chebvand) [4], Cauchy matrix (Cauchy), Discrete Cosine Transform matrix (DCT), Discrete Fourier Transform matrix (DFT) [8]. These generator matrices all contain ill-conditioned sub-matrices. Therefore, in these codes, when certain error patterns occur, an ill-conditioned linear system has to be solved to reconstruct an approximation of the original information, which can cause the loss of precision of possibly all digits in the recovered numbers.

To better understand the numerical properties of these real-number or complex-number codes, we give an example to demonstrate how inaccurate these codes may reconstruct the original data x in some erasure patterns.

Example 1: Suppose $x = (1, 1, 1, ..., 1)^T$ and the length of x is $N = 100$. G is a 120 by 100 generator matrix. $y = Gx$ is a vector of length 120. Suppose y_i, where $i = 101, 102, ...120$, are lost. We will use y_i, where $i = 1, 2, ...100$, to reconstruct x through solving (2) .

Table 1: The recovery accuracy of the existing codes for the erasure pattern in Example 1.

Name	$\kappa(G_J)$	$\frac{\|x-\widehat{x}\|_2}{\|x\|_2}$	Accurate digits
Vander	3.7e+218	2.4e+153	0
Chebvand	Inf	1.7e+156	0
Cauchy	5.6e+17	1.4e+03	0
DCT	1.5e+17	2.5e+02	0
DFT	2.0e+16	1.6e+00	0

Most generator matrices in literature have parameters. In order to carry on numerical calculations, in this example, we fix some parameters(these parameters do not affect the properties of the matrix we need fundamentally) and list the expression of the generator matrices we used:

- **Vander:** $G = \left((m+1)^{100-n-1}\right)_{120\times 100}$
- **Chebvand:** $G = \left(T_{m-1}(n)\right)_{120\times 100}$, where T_{m-1} is the chebyshev polynomial of degree $n-1$
- **Cauchy :** $G = \left(\frac{1}{m+n}\right)_{120\times 100}$
- **DCT :** $G = \left(\sqrt{\frac{i}{120}}\cos\frac{\pi(2n+1)m}{240}\right)_{120\times 100}$, where if $m = 0, i = 1$, and if $m \neq 0, i = 2$
- **DFT :** $G = \left(e^{-i\frac{2\pi}{120}mn}\right)_{120\times 100}$, where $i = \sqrt{-1}$

In Table 1, we list the name of each generator matrix, the condition number of the resulted G_J from the example erasure pattern, the relative error of the recovered x and how many decimal digits are accurate in the recovered x. All data are calculated using MATLAB. There are about 16 accurate digits in the original representation of x (i.e. the machine precision $\epsilon \approx 10^{-16}$ in MATLAB.)

From Table 1, we can see none of these codes is able to reconstruct the original data x with an accuracy of even only 1 decimal digit. Actually, for any burst erasures of 20 elements in y, none of the above codes could reconstruct an acceptable x.

V. Real and Complex Number Codes Based on Random Matrices

In this section, we propose a class of new codes that are able to reconstruct a very good approximation of the original information with high probability regardless of the erasure patterns in the encoded information. In subsection A, we define some notations and unify some concepts. In subsection B, we briefly review some important properties of the condition number of random matrices. In subsection C, we investigate the properties of the condition number of pseudo random matrices experimentally. In section D, we introduce our real-number and complex-number codes based on random matrices.

A. Definitions and Notions

We will assume that most readers are familiar with the basic terms and ideas from probability and numerical analysis. we then need only a few definitions.

Definition 1 *$N(\mu, \sigma^2)$ denotes the Normal Distribution with mean μ and variance σ^2.*

Definition 2 *$\widetilde{N}(\mu, \sigma^2)$ denotes the distribution of $x+yi$, where x and y are independent and identically distributed $N(\mu, \sigma^2)$.*

Definition 3 *$G(n, n)$ denotes an $n \times n$ matrix, where the n^2 elements are independent and identically distributed $N(0, 1)$.*

Definition 4 *$\widetilde{G}(n, n)$ denotes an $n \times n$ matrix, where the n^2 elements are independent and identically distributed $\widetilde{N}(0, 1)$.*

Definition 5 *Given a square matrix A, the condition number κ of A is defined as $\kappa = \| A \|_2 \times \| A^{-1} \|_2$. The scaled condition number κ_D of A is defined as $\kappa_D = \| A \|_F \times \| A^{-1} \|_2$.*

B. Condition Number of Random Matrices from Standard Normal Distribution

In solving a linear system, the large condition number of the coefficient matrix implies the loss of accuracy, and the logarithm of the condition number is an estimation of how many digits might be lost in the solution. Therefore, in this sub-section, we mainly focus on the probability that the condition number of a random matrix is large and the expectation of the logarithm of the condition number.

Theorem 1 *Let κ denote the condition number of $G(n, n)$, $n > 2$, and $t \geq 1$, then*

$$\frac{0.13n}{t} < P(\kappa > t) < \frac{5.60n}{t}. \tag{3}$$

Moreover,

$$E(\log(\kappa)) = \log(n) + c + \epsilon_n, \tag{4}$$

where $c \approx 1.537$, $lim_{n\to\infty} \epsilon_n = 0$,

Proof The inequality (3) is from Theorem 1 of [2]. The formula (4) can be obtained from Theorem 7.1 of [5].

Lemma 2 *Let $\widetilde{\kappa}$ denote the condition number of $\widetilde{G}(n, n)$, and $\widetilde{\kappa}_D$ denote the scaled condition number of $\widetilde{G}(n, n)$, then*

$$\frac{\widetilde{\kappa}_D}{\sqrt{n}} \leq \widetilde{\kappa} \leq \widetilde{\kappa}_D. \tag{5}$$

Proof The inequality (5) can be found in standard textbooks such as [9].

Theorem 3 *Let $\widetilde{\kappa}$ denote the condition number of $\widetilde{G}(n, n)$, and $t \geq \sqrt{n}$, then*

$$1 - \left(1 - \frac{1}{t^2}\right)^{n^2-1} \leq P(\widetilde{\kappa} > t) \leq 1 - \left(1 - \frac{n}{t^2}\right)^{n^2-1}. \tag{6}$$

Moreover,

$$E(\log(\widetilde{\kappa})) = \log(n) + c + \epsilon_n, \tag{7}$$

where $c \approx 0.982$, $lim_{n\to\infty} \epsilon_n = 0$,

Proof From Lemma 2, we get

$$P(\frac{\widetilde{\kappa}_D}{\sqrt{n}} > t) \leq P(\widetilde{\kappa} > t) \leq P(\widetilde{\kappa}_D > t). \tag{8}$$

From Corollary 3.2 in [6], we have

$$P(\widetilde{\kappa}_D > t) = 1 - \left(1 - \frac{n}{t^2}\right)^{n^2-1}. \tag{9}$$

Therefore,

$$P(\frac{\widetilde{\kappa}_D}{\sqrt{n}} > t) = P(\widetilde{\kappa}_D > \sqrt{n}t) = 1 - \left(1 - \frac{1}{t^2}\right)^{n^2-1}. \tag{10}$$

The inequality (6) can be obtained from (8), (9) and (10). The formula (4) can be obtained from Theorem 7.2 of [5].

C. Experiment on Pseudo Random Matrices

In sub-section B, we have proved some bounds and asymptotic properties for random matrices. However, in error correction practice, all random numbers used are pseudo random numbers, which have to be generated through a random number generator.

In this sub-section, we investigate experimentally the properties of the condition number of pseudo random matrices. All

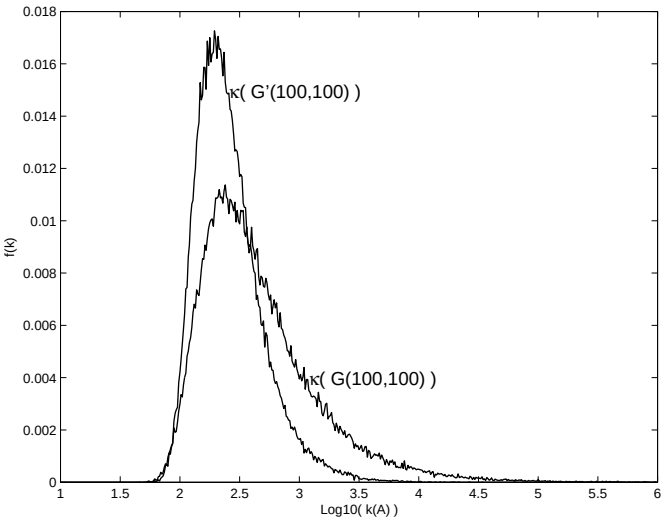

Figure 1: The density functions of the condition numbers of $G(100, 100)$ and $\widetilde{G}(100, 100)$.

random numbers we used are generated from the MATLAB pseudo random number generators. An $N \times N$ real pseudo random matrix from standard normal distribution is generated by $G(N, N)$= *randn(N,N)*. An $N \times N$ complex pseudo random matrix from standard normal distribution is generated by $\widetilde{G}(N, N)$= *randn(N,N)* + $\sqrt{-1}$ * *randn(N,N)*.

Figure 1 shows the empirical probability density functions of the condition numbers of $G(100, 100)$ and $\widetilde{G}(100, 100)$. From these density functions, we know that most pseudo random matrices have very small condition numbers. And, for the same matrix size, the tail of the condition number for a complex random matrix is thinner than that of a real one.

In Table 2, we give the proportion(can be explained as probability) of the 100 by 100 MATLAB pseudo random matrices whose condition numbers are large. The data are based on 1,000,000 sample pseudo random matrices from the standard normal distribution.

Table 2: The proportion of the 100 by 100 MATLAB pseudo random standard normal matrices whose condition number is large.

	$G(100, 100)$	$\widetilde{G}(100, 100)$
$P(\kappa \geq 10^2)$	0.97869	0.97788
$P(\kappa \geq 10^3)$	0.19211	0.03758
$P(\kappa \geq 10^4)$	0.01955	0.00041
$P(\kappa \geq 10^5)$	0.00201	0.00000
$P(\kappa \geq 10^6)$	0.00021	0.00000
$P(\kappa \geq 10^7)$	0.00003	0.00000
$P(\kappa \geq 10^8)$	0.00000	0.00000

We have also tested some other random matrices. Experiments show a lot of other random matrices, for example, uniformly distributed pseudo random matrices, also have small condition numbers with high probability. For random matrices of non-normal distribution, we will report our experiments and some analytical proofs of their condition number properties in a further coming paper.

D. Real and Complex Number Codes Based on Random Matrices

In this sub-section, we propose a class of new codes that are able to reconstruct a very good approximation of the original information with very high probability regardless of the erasure patterns in the encoded information.

In the real number case, we propose to use $G(M, N)$ or uniformly distributed M by N matrices with mean 0 (denote as $U(M, N)$) as our generator matrices G. In the complex number case, we propose to use $\widetilde{G}(M, N)$ or uniformly distributed M by N complex matrices with mean 0 (denote as $\widetilde{U}(M, N)$) as our generator matrices G.

Take the real-number codes based on random matrix $G(M, N)$ as an example. Since each element of the generator matrix $G(M, N)$ is a random number from the standard normal distribution, so each element of any $N \times N$ sub-matrix $(G_J)_{N\times N}$ of $G(M, N)$ is also a random number from the standard normal distribution. According to the condition number results in sub-section B and C, the probability that the condition number of $(G_J)_{N\times N}$ is large is very small. Hence, any N by N sub-matrix $(G_J)_{N\times N}$ of G is well-conditioned with very high probability. Therefore, no mater what erasure patterns occur, the error correction procedure is numerically stable with high probability.

We admit that our real-number and complex-number codes are not perfect. Due to the probability approach we used, the drawback of our codes is that, no matter how small the probability is, there is a probability that a erasure pattern may not be able to be recovered accurately.

However, compared with the existing codes in literature, the probability that our codes fail to recover a good approximation of the original information is negligible (see section VI for detail). Moreover, in the error correction practice, we may first generate a set of pseudo random generator matrices and then test each generator matrix until we find a satisfied one.

VI. Comparison with Existing Codes

To demonstrate that our codes are able to reconstruct a very good approximation of the original information x, we use our codes to recover the example erasure in section IV and compare the accuracy of the recovered x with that in section IV. The generator matrices of our codes are constructed using MATLAB pseudo random number generator. Table 3 shows how each of our generator matrices is generated. Table 4 compares the recovery accuracy of our codes for the example erasure in section IV with that of the existing codes.

Table 4 shows our codes are able to reconstruct the original information x with much higher accuracy than the existing

Table 3: The generator matrices G of our codes in Example 1

Name	MATLAB command to generate G
RandN	randn(120,100)
RandN-C	randn(120,100) + i * randn(120,100)
RandU	rand(120,100) - 0.5
RandU-C	rand(120,100) - 0.5 + i * (rand(120,100) - 0.5)

Table 4: The recovery accuracy of different codes for the erasure pattern in Example 1

Name	$\kappa(G_J)$	$\frac{\|x-\widehat{x}\|_2}{\|x\|_2}$	Accurate digits
Vander	3.7e+218	2.4e+153	0
Chebvand	Inf	1.7e+156	0
Cauchy	5.6e+17	1.4e+03	0
DCT	1.5e+17	2.5e+02	0
DFT	2.0e+16	1.6e+00	0
RandN	7.5e+2	3.8e-14	14
RandN-C	4.5e+2	6.8e-14	14
RandU	8.6e+2	3.7e-14	14
RandU-C	5.7e+2	2.6e-14	14

codes. The reconstructed x from all existing codes we tested in Example 1 lost all of their 16 effective digits. However, the reconstructed x from the codes we proposed in last section lost only about 2 effective digits.

The condition number of a sub-matrix is directly related with the accuracy of recovery, In Table 5, we compare the proportion of 100 by 100 sub-matrices whose condition number is larger than 10^i, where $i = 4, 6, 8$, and 10, for different kind of generator matrices of size 150 by 100. Our generator matrices are defined in Table 3. Other generator matrices are defined in Section IV. All results in Table 5 are calculated using MATLAB based on 1,000,000 randomly (uniformly) selected sub-matrices.

From Table 5, we can see, of the 1,000,000 randomly selected sub-matrices from any of our random generator matrices, there are 0.000% sub-matrices whose condition number is larger than 10^8. However, for all existing codes in literature that we have tested, there are at least 21.644% sub-matrices whose condition number is larger than 10^8. Therefore, the numerical properties of our codes are much more stable than the existing codes we have tested.

VII. Conclusion

In this paper, we have introduced a class of numerically stable error correction codes defined over real-number and complex-number fields. Our new codes are based on random generator matrices over real-number and complex-number fields. We have demonstrated that our codes are numerically very stable compared with the existing codes in literature.

Table 5: Percentage of 100 by 100 sub-matrices (of a 150 by 100 generator matrix) whose condition number is larger than 10^i.

Name	$\kappa \geq 10^4$	$\kappa \geq 10^6$	$\kappa \geq 10^8$	$\kappa \geq 10^{10}$
Vander	100.000%	100.000%	100.000%	100.000%
Chebvand	100.000%	100.000%	100.000%	100.000%
Cauchy	100.000%	100.000%	100.000%	100.000%
DCT	96.187%	75.837%	48.943%	28.027%
DFT	92.853%	56.913%	21.644%	5.414%
RandN	1.994%	0.023%	0.000%	0.000%
RandN-C	0.033%	0.000%	0.000%	0.000%
RandU	1.990%	0.018%	0.000%	0.000%
RandU-C	0.036%	0.000%	0.000%	0.000%

References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd Ed, Society for Industrial and Applied Mathematics, 1999.

[2] J. M. Azais and M.Wschebor, "Upper and lower bounds for the tails of the distribution of the condition number of a gaussian matrix", submitted for publication, 2003

[3] R. E. Blahut, *Algebraic Methods for Signal Processing and Communication Coding*, New York: Springer-Verlag, 1992.

[4] D. L. Boley, R. P. Brent, G. H. Golub and F. T. Luk, "Algorithmic Fault Tolerance Using the Lanczos Method," *SIAM Journal on Matrix Analysis and Applications*, vol. 13, (1992), pp. 312-332.

[5] A. Edelman, *Eigenvalues and Condition Numbers of Random Matrices*, Ph.D. thesis, Dept. of Math., M.I.T., 1989.

[6] A. Edelman, "On the distribution of a scaled condition number," *Mathematics of Computation*,vol. 58, (1992), pp. 185-190.

[7] P. Ferreira, "Stability issues in error control coding in complex field, interpolation, and frame bounds", *IEEE Signal Processing Letters*, vol.7 No.3,(2000) pp.57-59.

[8] P. Ferreira, J. Vieira, "Stable DFT codes and frames", *IEEE Signal Processing Letters*, vol.10 No.2,(2003) pp.50-53.

[9] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd Ed., The John Hopkins University Press, 1989.

[10] C. N. Hadjicostis and G. C. Verghese, "Coding approaches to fault tolerance in linear dynamic systems," Submitted to *IEEE Transactions on Information Theory*.

[11] W. Henkel, "Multiple error correction with analog codes," *Proceedings of AAECC*, Springer-Verlag, (1989), pp. 239-249.

[12] H. Huang and J. A. Abraham, "Algorithm-based fault tolerance for matrix operations," *IEEE Transactions on Computers*, vol. C-39, (1984) pp.300-304.

[13] F. T. Luk and H.Park, "An analysis of algorithm-based fault tolerance techniques," *Journal of Parallel and Distributed Computing*, vol. 5 (1988), pp. 1434-1438.

[14] F. Marvasti, M. Hasan, M. Echhart, S. Talebi, "Efficient algorithms for burst error recovery using FFT and other transform kernels," *IEEE Transactions on Signal Processing*, vol.47, No.4, (1999), pp. 1065-1075.

[15] S. S. Nair and J. A. Abraham, "Real-number codes for fault-tolerant matrix operations on processor arrays," *IEEE Transactions on Computers*, vol. C-39,(1990) pp.300-304.

[16] J. H. Wilkinson, "Error analysis revisited," *Bulletin of the Institute of Mathematics and its Application*, vol. 22, (1986), pp. 192-200.