ELSEVIER

Applied Numerical Mathematics 23 (1997) 275–289

APPLIED NUMERICAL MATHEMATICS

The use of the CADNA library for validating the numerical results of the hybrid GMRES algorithm

Faezeh Toutounian [1]

Department of Mathematics, Ferdowsi University of Mashhad, Mashhad, Iran

Abstract

Over the past several years a number of hybrid algorithms have been proposed for solving large sparse systems of linear algebraic equations. In this paper we consider the hybrid GMRES algorithm by Nachtigal, Reichel and Trefethen (1992) and show that in the floating-point arithmetic there exist some cases in which the properties of this algorithm are lost, e.g., the result is false, or the coefficients of the GMRES residual polynomial are non-significant and lead to serious round-off errors. The subject of this paper is to show how by using the CADNA library, it is possible during the run of the hybrid GMRES code to detect the numerical instabilities, to stop correctly the process, and to evaluate the accuracy of the results provided by the computer. Numerical examples are used to show the good numerical properties. © 1997 Elsevier Science B.V.

Keywords: Iterative methods; GMRES method; Hybrid methods; Error propagation; CESTAC method; Stochastic arithmetic; CADNA library

1. Introduction

In recent years there has been significant progress in the development of iterative methods for solving sparse real linear systems of the form

$$Ax = b, \tag{1}$$

where A is a nonsymmetric matrix of order n. The GMRES method by Saad and Schultz [17] is one of the most popular iterative methods for solving such systems. In this method the residual norm is explicitly minimized over the Krylov subspace in every iteration. It is clear that we face the same practical difficulties with the GMRES method as with many other Krylov subspace iterations. The number of operations and the storage requirements grow with the iteration number. To remedy this difficulty it was proposed in [17] to restart the algorithm after a cycle of iterations. This obviously fixes the maximum amount of storage and operations needed in an iteration. The convergence of

[1] E-mail: toutouni@science2.um.ac.ir.

PII S0168-9274(96)00054-2

cyclic variant GMRES, however, cannot be proved. It is often observed that due to the restarts the super-linear convergence behaviour of GMRES is lost, or even that no convergence occurred at all. As an alternative several authors have proposed to combine GMRES with another, more simple, iterative method [6,18,19]. The hybrid methods they proposed perform only a limited number of GMRES iterations and estimate eigenvalues and then apply this knowledge in further iterations. However, in [15] Nachtigal, Reichel and Trefethen have presented a hybrid method in which a few steps of GMRES are followed by a Richardson iteration [16] or Horner iteration [7] based on the polynomial implicitly constructed by GMRES. Unlike other hybrid algorithms, this one never estimates any eigenvalues and it is simpler than other hybrid iterations. In Section 2 we briefly describe GMRES and these hybrid GMRES algorithms, and then we show that in scientific computation, due to finite precision there exist some cases in which the properties of this algorithm are lost, e.g., the result is false, the coefficients of the residual polynomial are non-significant and lead to serious round-off errors. In the floating-point arithmetic each operation generally causes a round-off error. Computed results are irremediably affected by round-off error propagation. The results may sometimes become non-significant. This means that there is no common significant digit between the computed result and the corresponding exact result. As a consequence, it is necessary to control the accuracy of the computed results. The CESTAC method of La Porte and Vignes [12,21,23] is an efficient tool for estimating the accuracy of the results provided by a computer, to detect the numerical instabilities during the running of a program, and to improve the numerical algorithm. The basic idea of the CESTAC method is to replace the usual floating-point arithmetic with a random arithmetic. Consequently, each result appears as a random variable. This approach leads toward two concepts: stochastic numbers and stochastic arithmetic. In Section 3 we give a brief description of this arithmetic and the CADNA software [3,24]. CADNA is a library for programs written in FORTRAN 77, FORTRAN 90, or in ADA which allows computation using stochastic arithmetic by automatically implementing the synchronous CESTAC method.

The aim of this paper is to use the CADNA library to try to solve the problems which exist in running the hybrid GMRES program. In Section 4 we describe the hybrid GMRES algorithm using the CADNA library, pointing out all the advantages. Some numerical results are given to show the good numerical properties.

2. GMRES and hybrid GMRES algorithms

In this section we recall some fundamental properties of the GMRES and hybrid GMRES methods [15,17], which are iterative methods for solving linear systems with a nonsymmetric matrix. Then we discuss the problems which exist in the implementation of the hybrid GMRES algorithm on a computer using the floating-point arithmetic.

2.1. GMRES algorithm

Consider the linear system (1) with $\boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^n$ and with a nonsingular matrix $A \in \mathbb{R}^{n\times n}$. The Krylov subspace $K^m(A; \boldsymbol{r}_0)$ is defined by

$$K^m(A; \boldsymbol{r}_0) = \operatorname{span}\{\boldsymbol{r}_0, A\boldsymbol{r}_0, \ldots, A^{m-1}\boldsymbol{r}_0\}.$$

In GMRES, Arnoldi's method is used for the construction of an orthonormal basis $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ for $K^m(A; \boldsymbol{r}_0)$. The modified Gram–Schmidt version of Arnoldi's method can be described as follows [17,20]:

1. Start: Choose $\boldsymbol{x}_0$ and compute $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$ and $\boldsymbol{v}_1 = \boldsymbol{r}_0/\|\boldsymbol{r}_0\|_2$.
2. Iterate: For $j = 1, 2, \ldots, m$ do:

$$\boldsymbol{v}_{j+1} = A\boldsymbol{v}_j,$$
$$\text{for } i = 1, \ldots, j \text{ do:}$$
$$h_{ij} = \boldsymbol{v}_{j+1}^{\mathrm{T}}\boldsymbol{v}_i, \quad \boldsymbol{v}_{j+1} = \boldsymbol{v}_{j+1} - h_{ij}\boldsymbol{v}_i,$$
$$h_{j+1,j} = \|\boldsymbol{v}_{j+1}\|_2, \quad \boldsymbol{v}_{j+1} = \boldsymbol{v}_{j+1}/h_{j+1,j}.$$

(The nondefined h_{ij} are assumed to be zero.)

With the $n \times m$ matrix $V_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m]$ we have that $H_m = V_m^{\mathrm{T}} A V_m$ is an upper $m \times m$ Hessenberg matrix whose entries are the scalars h_{ij}. Formally Step 2 can be described by

$$AV_m = V_{m+1}\bar{H}_m, \tag{2}$$

where the $(m+1) \times m$ matrix $\bar{H}_m$ is the same as H_m except for an additional row whose only nonzero element is $h_{m+1,m}$ in the $(m+1, m)$ position.

The idea of GMRES is to construct an approximate solution of the form $\boldsymbol{x}_m = \boldsymbol{x}_0 + \boldsymbol{z}_m$ where $\boldsymbol{z}_m$ is an element of $K^m(A; \boldsymbol{r}_0)$ with the following property:

$$\|\boldsymbol{r}_m\|_2 = \|\boldsymbol{b} - A\boldsymbol{x}_m\|_2 = \min_{\boldsymbol{z} \in K^m(A;\boldsymbol{r}_0)} \|\boldsymbol{r}_0 - A\boldsymbol{z}\|_2. \tag{3}$$

If we set $\boldsymbol{z} = V_m\boldsymbol{y}$, by using (2) and the fact that V_{m+1} is orthonormal, the last expression in (3) is equal to

$$\min_{\boldsymbol{y} \in \mathbb{R}^m} \|\boldsymbol{r}_0 - AV_m\boldsymbol{y}\|_2 = \min_{\boldsymbol{y} \in \mathbb{R}^m} \|\boldsymbol{r}_0 - V_{m+1}\bar{H}_m\boldsymbol{y}\|_2 = \min_{\boldsymbol{y} \in \mathbb{R}^m} \|\beta\boldsymbol{e}_1 - \bar{H}_m\boldsymbol{y}\|_2, \tag{4}$$

with $\beta = \|r_0\|_2$ and $\boldsymbol{e}_1$ the first unit vector in $\mathbb{R}^{m+1}$. Hence, the GMRES iterate is given by $\boldsymbol{x}_m = \boldsymbol{x}_0 + V_m\boldsymbol{y}_m$, where $\boldsymbol{y}_m$ is the solution to the upper-Hessenberg least-squares problem on the right-hand side of (4). By factoring $\bar{H}_m$ as $\bar{H}_m = Q_m^{\mathrm{T}} R_m$, in which $Q_m \in \mathbb{R}^{(m+1)\times(m+1)}$ is a product of Givens rotations and $R_m \in \mathbb{R}^{(m+1)\times m}$ is an upper triangular matrix, this least-squares problem is equivalent to

$$\|\beta\boldsymbol{e}_1 - \bar{H}_m\boldsymbol{y}_m\|_2 = \min_{\boldsymbol{y} \in \mathbb{R}^m} \|Q_m\beta\boldsymbol{e}_1 - R_m\boldsymbol{y}\|_2.$$

Since the last row of R_m is zero, $\boldsymbol{y}_m$ is the solution of the linear system with the leading $m \times m$ submatrix R_m as the matrix and the first m elements of $Q_m\beta\boldsymbol{e}_1$ as the right-hand side. By construction of $\boldsymbol{y}_m$ the residual norm of the approximate solution $\boldsymbol{x}_m$ is equal to the absolute value of the $(m+1)$st element of $Q_m\beta\boldsymbol{e}_1$. We cite that the residual norm is nonincreasing, because it is minimized at every step of the GMRES method, and for an $n \times n$ problem GMRES terminates in at most n steps. More details about the GMRES method can be found in [17].

2.2. Hybrid GMRES algorithm

The GMRES iteration constructs a sequence of residual polynomials that minimize the norm of the residual

$$\|\boldsymbol{r}_m\|_2 = \|p_m(A)\boldsymbol{r}_0\|_2 = \min_{p\in P_m,\ p(0)=1} \|p(A)\boldsymbol{r}_0\|_2, \quad m = 1, 2, \ldots .$$

With these GMRES polynomials the following hybrid GMRES is proposed in [15]:

Start with a random initial guess $\boldsymbol{x}_0$.
Phase I: Run GMRES until $\|\boldsymbol{r}_m\|_2$ drops by a suitable amount. Set $\nu = m$.
Phase II: Re-apply the GMRES residual polynomial $p_\nu(z)$ cyclically until convergence.

The main idea of this algorithm is to suppose that at the νth GMRES step the relation

$$\frac{\|\boldsymbol{r}_\nu\|_2}{\|\boldsymbol{r}_0\|_2} = \frac{\|p_\nu(A)\boldsymbol{r}_0\|_2}{\|\boldsymbol{r}_0\|_2} = \tau$$

holds for some $\tau < 1$, and moreover that we have

$$\|p_\nu(A)\|_2 \simeq \tau.$$

So by re-applying the GMRES polynomial $p_\nu(z)$ cyclically we can reduce the residual norm. Of course, these assumptions do not always hold and we must modify the algorithm in order to cope with failure. By assumption that storage is not limited, we propose the following safeguarding procedure which differs from Nachtigal, Reichel and Trefethen algorithm [15] in Step 2:

(1) If any cycle of ν steps of Phase II reduces $\|\boldsymbol{r}_m\|_2$ by a factor less than $\sqrt{\tau}$—that is, if the convergence is more than twice as slow as expected—return to Phase I.
(2) Carry out additional GMRES steps $\nu+1, \nu+2, \ldots, \nu'$ of Phase I until $\|\boldsymbol{r}_{\nu'}\|_2/\|\boldsymbol{r}_\nu\|_2 < 1/2$, and calculate a new polynomial $p_{\nu'}(z)$.
(3) Begin a new Phase II iteration with the new polynomial $p_{\nu'}(z)$, starting from the previous best value $\boldsymbol{x}_m$, which will come either from the previous Phase II if the convergence there was slow but positive, or from the new Phase I if there was actual divergence in the previous Phase II.

As shown in [15], we can calculate the coefficients of $p_\nu(z)$ in the following way. Let K_m denote $n \times m$ matrix of Krylov vectors

$$K_m = [\boldsymbol{r}_0, A\boldsymbol{r}_0, \ldots, A^{m-1}\boldsymbol{r}_0].$$

Since the columns of V_m and K_m span the same space for each m, we have

$$V_m = K_m C_m$$

for some upper triangular matrix

$$C_m = \begin{bmatrix} c_{11} & \cdots & c_{1m} \\ & \ddots & \vdots \\ & & c_{mm} \end{bmatrix}.$$

By (2) we have

$$\boldsymbol{v}_{m+1} = h_{m+1,m}^{-1}(A\boldsymbol{v}_m - V_m\boldsymbol{h}_m), \quad \boldsymbol{h}_m = [h_{1m}, \ldots, h_{mm}]^{\mathrm{T}},$$

thus we obtain the formula

$$\begin{bmatrix} c_{1,m+1} \\ \vdots \\ c_{m+1,m+1} \end{bmatrix} = h_{m+1,m}^{-1} \begin{bmatrix} 0 \\ c_{1,m} \\ \vdots \\ c_{m,m} \end{bmatrix} - h_{m+1,m}^{-1} \begin{bmatrix} C_m \boldsymbol{h}_m \\ 0 \end{bmatrix},$$

for generating the elements of C_m column by column as the GMRES iteration proceeds. As previously seen, at step $m = \nu$, GMRES produces an iterate $\boldsymbol{x}_\nu$ of the form

$$\boldsymbol{x}_\nu = \boldsymbol{x}_0 + V_\nu \boldsymbol{y}_\nu, \quad \boldsymbol{y}_\nu \in \mathbb{R}^\nu.$$

From $V_\nu \boldsymbol{y}_\nu = K_\nu C_\nu \boldsymbol{y}_\nu$, we get

$$\boldsymbol{x}_\nu = \boldsymbol{x}_0 + q_{\nu-1}(A)\boldsymbol{r}_0,$$

where

$$q_{\nu-1}(z) = \alpha_0 + \alpha_1 z + \cdots + \alpha_{\nu-1} z^{\nu-1}, \quad [\alpha_0, \ldots, \alpha_{\nu-1}]^{\mathrm{T}} = C_\nu \boldsymbol{y}_\nu.$$

Since $p_\nu(z) = 1 - z q_{\nu-1}(z)$, by computing the vector $C_\nu \boldsymbol{y}_\nu$ we can explicitly obtain the coefficients of the residual polynomial $p_\nu(z)$. More details about this hybrid GMRES method can be found in [15].

2.3. *Numerical problems encountered in the method*

When the hybrid GMRES is implemented on a computer, at each iteration q of this algorithm we have one approximate solution from Phase I, denoted $\boldsymbol{x}_{0\nu}$, and some approximate solutions from Phase II, denoted $\boldsymbol{x}_{k\nu}$, $k = 1, 2, \ldots$. During the run of the program if any of these approximate solutions is a satisfactory solution, in the sense that its residual norm is reduced by a factor ε (where ε is an arbitrary positive value), then the program can be stopped. So, for stopping the process at iteration q with $\nu < n$, $k = 0, 1, \ldots$, we can use the following termination criterion:

$$\text{if} \quad \|\boldsymbol{r}_{k\nu}\|_2 \leqslant \varepsilon \|\boldsymbol{r}_0\|_2 \quad \text{then stop.}$$

In the case $\nu = n$, the program must be stopped because the GMRES method converges, in the absence of rounding errors, in at most n iterations.

Note that in the above termination criterion ε is an arbitrary value, the results of Example 1, Section 4 (Table 2, in which only the decimal significant digits are printed) show that when ε is chosen too large ($\varepsilon = 10^{-6}$) the process is broken off too early and consequently the solution obtained has poor accuracy. On the contrary when ε is chosen too small ($\varepsilon = 10^{-16}$) the iterative process is stopped too late and many useless iterations are performed, without improving the accuracy of the solution obtained with $\varepsilon = 10^{-15}$. In practice it is absolutely impossible to choose correctly the value of the convergence tolerance ε. But as explained in Section 4 with the CADNA library it is possible to break off the iteration of an iterative process as soon as a satisfactory computed solution is reached, and this without using any arbitrary ε.

Let us now consider the linear system

$$\begin{bmatrix} 21 & 130 & 0 & 2.1 \\ 13 & 80 & 4.74\text{E+8} & 752 \\ 0 & -0.4 & 3.9816\text{E+8} & 4.2 \\ 0 & 0 & 1.7 & 9\text{E}-9 \end{bmatrix} x = \begin{bmatrix} 153.1 \\ 849.74 \\ 7.7816 \\ 2.6\text{E}-8 \end{bmatrix}, \tag{5}$$

which was described in [24]. The exact solution is $\boldsymbol{x} = [1.0, 1.0, 10^{-8}, 1.0]^{\mathrm{T}}$. The approximate solution obtained with the initial guess $\boldsymbol{x}_0 = [0, 0, \ldots, 0]^T$ and the FORTRAN code of the hybrid GMRES algorithm, performed on a SUN4 computer, in double precision is as follows:

$$\boldsymbol{x}(1) = -89.8760751972095591, \qquad \boldsymbol{x}(2) = 15.6799813793030225,$$
$$\boldsymbol{x}(3) = 2.4747823346160658\text{E}-08, \qquad \boldsymbol{x}(4) = 1.0000000003115719.$$

It is necessary to say that this approximate solution has been obtained by Phase I with $\nu = 4$. This solution is false and the hybrid GMRES algorithm is not able, because of the propagation of the round-off errors, to provide a satisfactory solution for this example. However, nothing in the software nor in the solution allows the user to be aware that the computed solution is false. In Section 3 we will show that the CADNA library is able to estimate the propagation of round-off errors during the run of the code and to furnish the accuracy of the computed solution.

Let us consider another linear system with the $n \times n$ block diagonal matrix

$$A = \begin{bmatrix} M_1 & & & \\ & M_2 & & \\ & & \ddots & \\ & & & M_{n/2} \end{bmatrix}, \quad \text{where } M_j = \begin{bmatrix} 1 & j-1+\alpha \\ 0 & -1 \end{bmatrix}, \tag{6}$$

which with $\alpha = 0$ corresponds to the example $B_{\pm 1}$ of [14], and the second member $\boldsymbol{b} = [5, -3, 4, -4, 1, \ldots, 1]^{\mathrm{T}}$. The exact solution is given by $\boldsymbol{x}_{2j-1} = \boldsymbol{b}_{2j-1} + (j-1+\alpha)\boldsymbol{b}_{2j}$ and $\boldsymbol{x}_{2j} = -\boldsymbol{b}_{2j}$. With $\alpha = 1.11$, $n = 150$, and the initial guess $\boldsymbol{x}_0 = [0, 0, \ldots, 0]^{\mathrm{T}}$ the computed coefficients α_i of the GMRES polynomial $p_6(z)$ with single and double floating-point arithmetic rounded to nearest mode are presented in Table 1. In Section 4, thanks to the CADNA library, we consider that these coefficients are non-significant. It is clear that Phase II is not able here to improve the approximate solution which has been obtained by Phase I. In general, when ν has a large value, it is possible to have an unstable solution $\boldsymbol{x}_{k\nu}$, $k = 1, 2, \ldots$, and also a GMRES polynomial with non-significant coefficients. In this situation the use of Phase II has obviously no sense. So, in order to avoid the performance of many useless operations of Phase II and to prevent an overflow which may occur, it is better that program performs only Phase I as soon as such an instability occurs during the run. We now face the question of how we can detect this kind of instabilities? The CADNA library is a precious tool for obtaining an answer to this question. In Section 4 we show that with the CADNA library it is possible, by including a simple test to detect it.

The CADNA library which allows to solve the above numerical problems is a tool for automatic synchronous implementation of the CESTAC method of J. Vignes. In the following section we give

Table 1

	Single precision	Double precision
α_0	−2.9381578E+05	1.6409978460272019E+14
α_1	3.0272866E+05	−1.6409978460272156E+14
α_2	5.8763362E+05	−3.2819956920543981E+14
α_3	−6.0545688E+05	3.2819956920544550E+14
α_4	−2.9381784E+05	1.6409978460271962E+14
α_5	3.0272922E+05	−1.6409978460272297E+14

a brief description of the CESTAC method which is an efficient method for solving the numerical problems such as those described above.

3. The CESTAC method

3.1. Basic ideas of the CESTAC method

When some numerical algorithm is performed on a computer, each result thus provided always contains an error resulting from round-off error propagation. The CESTAC method (Contrôle et Estimation Stochastique des Arrondis de Calculs) was developed by La Porte and Vignes, and was then generalized by the latter. It allows to estimate the round-off error on each result and consequently provides the accuracy of this result.

The CESTAC method is based on a probabilistic approach of the round-off error propagation [8,10,11]. It replaces ordinary floating-point arithmetic by a random arithmetic which consists in randomly perturbing the lowest weight bit of the mantissa of the result of each arithmetic operation. By running the same program N times ($N = 2$ or 3), for each result of any floating-point arithmetic operation, a set of N computed results R_i, $i = 1, \ldots, N$, is obtained. It has been proved [1,4] that, under hypotheses which generally hold in real problems, these N results belong to a quasi-Gaussian distribution centered on the exact mathematical result. The mean value of these N results must be provided and the accuracy of this mean value may be estimated from these N results by the use of the Student's law. More precisely, it has been proved [1] that every result R obtained with random arithmetic can be given by

$$R = r + \sum_{i=1}^{n} u_i(d) 2^{-p} (\alpha_i - h_i) + \mathrm{O}\left(2^{-2p}\right),$$

where r is the exact result, the $u_i(d)$ are quantities depending on the data and on the computer program, but independent of the usual relative round-off error α_i and of the random perturbations h_i. The integer n is the number of arithmetical operations involved in the computation of R, and the integer p is the number of bits in the mantissa. It has also been proved [1] that, under the same hypotheses, N computations of R with the random arithmetic provide R_i, $i = 1, \ldots, N$, results belonging to a quasi-Gaussian distribution centered on the exact result r. So, in practice, from N realizations R_i, the

informatical results that must be provided are the mean value $\bar{R} = (1/N)\sum_{i=1}^{N} R_i$ of the R_i and the number of significant digits of $\bar{R}$, which can be estimated with the equation $C_{\bar{R}} = \log_{10}(\sqrt{N}|\bar{R}|/\tau_\beta\sigma)$, in which

$$\sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}(R_i - \bar{R})^2,$$

and τ_β is the value of the Student's distribution for $N-1$ degrees of freedom and a probability level $1-\beta$.

3.2. *Stochastic arithmetic*

When using the CESTAC method so that the N runs of the computer program take place in parallel, the N results of each arithmetic operation can be considered as realizations of Gaussian random variables centered on the exact result. This implementation is called synchronous implementation of the CESTAC method. Stochastic arithmetic models the synchronous implementation of the CESTAC method.

We can therefore define a new number, called stochastic number, and a new arithmetic, called stochastic arithmetic, applied to these numbers. Below, we present the main definitions and properties of this arithmetic. For more details see [5].

Definition 1. We define the set S of stochastic numbers as the set of Gaussian random variables. We denote an element $X \in S$ by $X = (\mu, \sigma^2)$, where μ is the mean value of X and σ its standard deviation. If $X \in S$ and $X = (\mu, \sigma^2)$, there exists λ_β, depending only on β, such that

$$P\big(X \in [\mu - \lambda_\beta\sigma,\ \mu + \lambda_\beta\sigma]\big) = 1 - \beta.$$

$I_{\beta,X} = [\mu - \lambda_\beta\sigma,\ \mu + \lambda_\beta\sigma]$ is a confidence interval of μ at $(1-\beta)$. An upper bound to the number of significant digits common to μ and each element of $I_{\beta,X}$ is

$$C_{\beta,X} = \log_{10}\left(\frac{|\mu|}{\lambda_\beta\sigma}\right).$$

The following definition is the modeling of the concept of informatical zero proposed in [22].

Definition 2. $X \in S$ is a stochastic zero, denoted $\underline{0}$, if and only if

$$C_{\beta,X} \leqslant 0 \quad \text{or} \quad X = (0,0).$$

Definition 3. Let $X_1 = (\mu_1, \sigma_1^2)$ and $X_2 = (\mu_2, \sigma_2^2)$ be two elements of S. We define the four elementary stochastic operations denoted $(s+, s-, s*, s/)$ on the stochastic numbers by

$$X_1\ s+\ X_2 \overset{\text{def}}{=} (\mu_1 + \mu_2,\ \sigma_1^2 + \sigma_2^2),$$
$$X_1\ s-\ X_2 \overset{\text{def}}{=} (\mu_1 - \mu_2,\ \sigma_1^2 + \sigma_2^2),$$
$$X_1\ s*\ X_2 \overset{\text{def}}{=} (\mu_1 * \mu_2,\ \mu_2^2\sigma_1^2 + \mu_1^2\sigma_2^2),$$
$$X_1\ s/\ X_2 \overset{\text{def}}{=} (\mu_1/\mu_2,\ (\sigma_1/\mu_2)^2 + (\mu_1\sigma_2/\mu_2^2)^2), \quad \text{with } \mu_2 \neq 0.$$

Definition 4. Let X_1 and X_2 be two elements of S, X_1 is stochastically equal to X_2, denoted $X_1 s{=}X_2$, if and only if

$$X_1 \; s- \; X_2 = \underline{0}.$$

Definition 5. Let $X_1 = (\mu_1, \sigma_1^2)$ and $X_2 = (\mu_2, \sigma_2^2)$ be two elements of S. X_1 is stochastically strictly greater than X_2, denoted $X_1 \; s{>} \; X_2$, if and only if

$$\mu_1 - \mu_2 > \lambda_\beta \sqrt{\sigma_1^2 + \sigma_2^2}.$$

Definition 6. Let $X_1 = (\mu_1, \sigma_1^2)$, $X_2 = (\mu_2, \sigma_2^2)$ be elements of S. X_1 is stochastically greater than X_2, denoted $X_1 \; s{\geqslant} \; X_2$, if and only if

$$X_1 \; s{>} \; X_2 \quad \text{or} \quad X_1 \; s{=} \; X_2.$$

Based on these definitions, the following properties of stochastic arithmetic have been proved:

- $s{=}$ is reflexive and symmetric but not transitive,
- $s{>}$ is transitive,
- $s{\geqslant}$ is reflexive, anti-symmetric, but not transitive:

$$a \; s{\leqslant} \; b \quad \Rightarrow \quad a \; s{=} \; b \quad \text{or} \quad a \; s{<} \; b,$$

$$a \; s{=} \; b \quad \text{and} \quad b \; s{<} \; c \quad \Rightarrow \quad a \; s{\leqslant} \; c,$$

$$a \; s{\leqslant} \; b \quad \text{and} \quad b \; s{<} \; c \quad \Rightarrow \quad a \; s{\leqslant} \; c.$$

$\underline{0}$ is absorbent for operation $s*$ and is the neutral element for operation $s+$. Let $x, y \in \mathbb{R}$ and let $X, Y \in S$ respectively be their representatives. If $X \; s{<} \; Y \Rightarrow x < y$. Thus the stochastic arithmetic retrieves properties of the exact arithmetic, which are lost by the usual floating-point arithmetic such as associativity, distributivity, the concept of remarkable identities. The stochastic arithmetic described above can be used in scientific codes to serve:

(1) during the run of a scientific code, to estimate the accuracy of any numerical result, to detect the numerical instabilities, and to check the branchings;
(2) to eliminate the programming expedients that are absolutely unfounded, such as those used, for example, in the termination criteria of iterative methods, and replace them by criteria that directly reflect the mathematical condition that must be satisfied at the solution.

3.3. The CADNA library

CADNA (Control of Accuracy and Debugging for Numerical Applications) is a library for programs written in FORTRAN 77, FORTRAN 90, or in ADA which allows the computation using stochastic arithmetic by automatically implementing the CESTAC method [2,3]. CADNA is able to estimate the accuracy of the computed results, and to detect numerical instabilities occurring during the run. To use the CADNA library, it suffices to place the instruction USE CADNA at the top of the initial FORTRAN or ADA source code and to replace the declarations of the real type by the stochastic type and to change some statements such as printing statements.

During the run, as soon as a numerical anomaly (for example, appearance of the informatical zero in a computation or a criterion) occurs, a message is written in a special file called Cadna_stability_f90.lst.

The user must consult this file after the program has run. If it is empty, this means the program has been run without any problem, that it has accordingly been validated, and that the results have been given with their associated accuracy. If it contains messages, the user, using the debugger associated with the compiler, will find the instructions that are the cause of these numerical anomalies, and must reflect in order to correct them if necessary. The program execution time using the CADNA library is only multiplied by a factor 3, which is perfectly acceptable in view of the major advantage offered, i.e., the validation of programs. CADNA is also able to estimate the influence of data errors on the result provided by the computer.

4. Using the CADNA library in the hybrid GMRES method

As we have seen in Section 2.3, in the implementation of the hybrid GMRES method three difficulties arise. Let us first consider the third problem which concerns the numerical instabilities which may occur in Phase II. By remarking that a GMRES polynomial with non-significant coefficients and also an unstable solution $\boldsymbol{x}_{k\nu}$, $k = 1, 2, \ldots$, always provides an non-significant residual norm with large magnitude, we can define with the CADNA library the test

$$\text{if} \quad \|\boldsymbol{r}_{k\nu}\|_2 = \underline{0} \quad \text{and} \quad \|\boldsymbol{r}_{k\nu}\|_2 \geqslant 1.0 \quad \text{then} \quad \textit{index} = 1, \tag{7}$$

with initial value $\textit{index} = 0$, which allows us by checking the value of parameter *index* to detect the presence of an unstable solution. As a consequence, the program, during the run, can decide to perform both Phases I and II, or only Phase I according to the value of this parameter *index*.

Now, we consider the first problem which concerns choosing a stopping criterion. As explained in [22], from a mathematical point of view, once we know a solution $\boldsymbol{x}_m$, we can validate its validity by checking the value of the residual norm

$$\|\boldsymbol{r}_m\|_2 = \|\boldsymbol{b} - A\boldsymbol{x}_m\|_2 = 0.$$

Obviously with usual floating-point arithmetic this equality is never satisfied even when the solution $\boldsymbol{x}_m$ is the exact solution, because of the round-off error propagation. However, with the CADNA library in view of its properties, the result will be

$$\|\boldsymbol{r}_m\|_2 = \|\boldsymbol{b} - A\boldsymbol{x}_m\|_2 = \underline{0}.$$

So, as soon as the residual norm of a stable computed solution is equal to the informatical zero, a satisfactory informatical solution is reached and the iterative process must be stopped. Now, by noting that it is possible, in Phase II, to have an unstable solution $\boldsymbol{x}_{k\nu}$, $k = 1, 2, \ldots$, that we can detect by the test (7), we define the termination criterion

$$\text{if} \quad \|\boldsymbol{r}_{k\nu}\|_2 = \underline{0} \quad \text{then stop}, \tag{8}$$

for checking the value of the residual norm of a stable solution $\boldsymbol{x}_{k\nu}$, $k = 0, 1, \ldots$. This termination criterion stops the iterative hybrid GMRES process as soon as a satisfactory solution is reached either by Phase I or by Phase II. It is necessary to say that the instability of the solution $\boldsymbol{x}_{k\nu}$, $k = 1, 2, \ldots$, must be checked before using the termination criterion (8).

Finally the second problem, which concerns the accuracy of the computed solution, will be solved by using the CADNA library. Since, as it was explained in Section 3.3, the CADNA library is able to estimate the accuracy of the computed results and to furnish the results with their exact decimal figures.

Let us now present the examples and the results which we obtained by the FORTRAN code of the hybrid GMRES method, combined with the CADNA library and the above tests. The computation has been performed on a SUN4 computer in double or simple precision with the stochastic arithmetic using the CADNA library.

Example 1. We consider the linear system with

$$A = \begin{bmatrix} 2 & 1 & & & & & \\ 0 & 2 & 1 & & & & \\ 1 & 0 & 2 & 1 & & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & 1 & 0 & 2 & 1 & \\ & & & 1 & 0 & 2 & \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} 3 \\ 3 \\ 4 \\ \vdots \\ 4 \\ 3 \end{bmatrix},$$

which was described in [9], with dimension equal to 400. The exact solution is given by $\boldsymbol{x} = [1, 1, \ldots, 1]^{\mathrm{T}}$. The results obtained by using the floating-point arithmetic with $\varepsilon = 10^{-6}$, 10^{-15}, 10^{-16}, and the CADNA library are presented in Table 2. Only the decimal significant digits are printed. At the last line of this table we can find the corresponding value of indices ν and k of the presented solution $\boldsymbol{x}_{k\nu}$.

For this example the initial guess has been $\boldsymbol{x}_0 = [0, 0, \ldots, 0]^{\mathrm{T}}$. From the results of Table 2, it appears that when ε is too large, the iterative process is stopped too soon (at $\nu = 4$, $k = 11$), and the solution furnished is not the best that the computer may provide. When ε is chosen too small ($\varepsilon = 10^{-16}$) the iterative process is stopped too late (at $\nu = 6$, $k = 1$), and many useless iterations

Table 2

	$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-15}$	$\varepsilon = 10^{-16}$	CADNA library
$\boldsymbol{x}(1)$	1.0000000	1.0000000000000000	1.0000000000000000	0.100000000000000E+1
$\boldsymbol{x}(2)$	0.999999	1.0000000000000000	1.0000000000000000	0.999999999999999E+0
$\boldsymbol{x}(3)$	1.000000	1.000000000000000	1.000000000000000	0.100000000000000E+1
$\boldsymbol{x}(4)$	1.000000	0.9999999999999999	0.9999999999999999	0.100000000000000E+1
$\boldsymbol{x}(5)$	0.999999	1.0000000000000000	1.0000000000000000	0.999999999999999E+0
⋮	⋮	⋮	⋮	⋮
$\boldsymbol{x}(396)$	1.000000	1.0000000000000000	1.0000000000000000	0.999999999999999E+0
$\boldsymbol{x}(397)$	0.9999999	1.0000000000000000	1.0000000000000000	0.100000000000000E+1
$\boldsymbol{x}(398)$	0.9999999	1.000000000000000	1.000000000000000	0.100000000000000E+1
$\boldsymbol{x}(399)$	1.000000	1.0000000000000000	1.0000000000000000	0.999999999999999E+0
$\boldsymbol{x}(400)$	0.9999999	1.0000000000000000	1.0000000000000000	0.100000000000000E+1
	$\nu = 4$	$\nu = 4$	$\nu = 6$	$\nu = 4$
	$k = 11$	$k = 46$	$k = 1$	$k = 48$

are performed without improving the accuracy of the solution with $\varepsilon = 10^{-15}$. By using the CADNA library, the optimal termination criterion (8) has stopped the iterative process at $\nu = 4$, $k = 48$, and the solution is reached with about 15 exact significant digits on all the elements.

Remark. For controlling the quality of a computed solution $\boldsymbol{X}$ of a linear system $A\boldsymbol{x} = \boldsymbol{b}$ we can use the normalized residual test. As explained in [12,13], this test consists in computing the normalized residuals

$$\rho_i^* = \frac{|\rho_i|}{2^{-p}\sqrt{m_i^q(\sum_{j=1}^n (A_{ij}X_j)^2 + B_i^2)}}, \quad i = 1, \ldots, n,$$

where

$$\rho_i = B_i - \sum_{j=1}^{n} A_{ij}X_j$$

and A_{ij} and B_i are the normalized floating-point representations of a_{ij} and b_i, respectively, X_j is the jth element of computed solution $\boldsymbol{X}$, and $q = 1$ for rounding to the nearest mode, and $q = 2$ for other rounding modes. The integer m_i is the number of nonzero elements of row i, and the integer p is the number of bits in the mantissa.

The three following cases can occur:

Case I. All the n normalized residuals are of the order of magnitude 1:

$$\rho_i^* \sim 1, \quad \forall i \in [1, 2, \ldots, n],$$

thus the computed solution $\boldsymbol{X}$ is a satisfactory informatical solution.

Case II. At least one of the normalized residuals is strictly greater than one, but strictly smaller than 2^p:

$$1 \ll \rho_i^* \ll 2^p,$$

thus the computed solution $\boldsymbol{X}$ is not a satisfactory informatical solution, but it is possible to improve it by an incremental vector $\Delta\boldsymbol{X}$ which may be obtained by solving the linear system $A\Delta\boldsymbol{X} = \boldsymbol{R}$, where $\boldsymbol{R}$ is the residual vector with ith element ρ_i.

Case III. At least one of the normalized residuals is of the order of magnitude 2^p:

$$\rho_i^* \sim 2^p,$$

thus $\boldsymbol{X}$ is a bad solution and in general, we cannot improve it. If this situation occurs, this means that the used method is not adapted to the proposed system.

By applying this test to the computed solutions of the above example, we discover that all the above computed solutions are the satisfactory informatical solutions except the one which is obtained by the floating-point arithmetic with $\varepsilon = 10^{-6}$. By improving this solution twice, we could obtain a satisfactory informatical solution. We observe that without using the CADNA library, it is very difficult to obtain a satisfactory informatical solution.

Example 2. Let us again consider the linear system (5). The solution obtained with the CADNA library is as follows:

$$\boldsymbol{x}(1) = \underline{0}, \qquad \boldsymbol{x}(2) = \underline{0},$$
$$\boldsymbol{x}(3) = \underline{0}, \qquad \boldsymbol{x}(4) = 0.9999999\text{E+}00.$$

Table 3

	$\varepsilon = 10^{-5}$	$\varepsilon = 10^{-6}$	CADNA library
$x(1)$	1.6700622	1.6700689	0.167001E+01
$x(2)$	2.9999490	2.9998803	0.299997E+01
$x(3)$	−4.4399438	−4.4398289	−0.443998E+01
$x(4)$	3.9999347	3.9998538	0.399999E+01
$x(5)$	4.1099434	4.1098661	0.41099E+01
⋮	⋮	⋮	⋮
$x(146)$	−0.9999837	−0.9999635	−0.99999E+00
$x(147)$	75.1088562	75.1074753	0.75109E+02
$x(148)$	−0.9999837	−0.9999635	−0.99999E+00
$x(149)$	76.1087952	76.1073151	0.76109E+02
$x(150)$	−0.9999837	−0.9999635	−0.99999E+00
	$\nu = 7$	$\nu = 11$	$\nu = 12$
	$k = 0$	$k = 0$	$k = 0$

Table 4

	$\varepsilon = 10^{-5}$	$\varepsilon = 10^{-6}$
$x(1)$	1.6699998	1.6700000
$x(2)$	3.0000000	3.0000000
$x(3)$	−4.4400001	−4.4400010
$x(4)$	4.0000000	4.0000000
$x(5)$	4.1100001	4.1100001
⋮	⋮	⋮
$x(146)$	−1.0000000	−1.0000000
$x(147)$	75.1100006	75.1100006
$x(148)$	−1.0000000	−1.0000000
$x(149)$	76.1100006	76.1100006
$x(150)$	−1.0000000	−1.0000000
	$\nu = 9$	$\nu = 5$
	$k = 0$	$k = 0$

These results show that the first three elements of the computed solution are non-significant and the last one has seven significant digits. We observe that only by using the CADNA library it is possible to conclude that the results obtained for the first three elements of the computed solution must be due to round-off error propagation.

Example 3. Let us again consider the linear system (6). The solutions furnished by using single floating-point arithmetic with $\varepsilon = 10^{-5}$, $\varepsilon = 10^{-6}$, and the CADNA library are presented in Table 3. With $\varepsilon = 10^{-7}$ no solution has been obtained, because an overflow occurred during the run of code. The CADNA library detected the numerical instabilities and showed that all the coefficients of the GMRES polynomial presented in Table 1 are $\underline{0}$, i.e., they have no significant digit. The test of normalized residuals showed that the solution obtained by the CADNA library is a satisfactory informatical solution, but those which are obtained by the floating-point arithmetic are not. By solving the corresponding linear systems $A\Delta\boldsymbol{X} = \boldsymbol{R}$ we could improve the solutions obtained with $\varepsilon = 10^{-5}$, $\varepsilon = 10^{-6}$ and obtain the satisfactory informatical solutions which are presented in Table 4. Finally, with many difficulties, we obtained the satisfactory informatical solutions, but what is the accuracy of each element of these solutions? We observe that the CADNA library not only has obtained a satisfactory informatical solution, but also has furnished the elements of the computed solution with their exact digits.

5. Conclusion

In this paper we have seen that in the implementation of the hybrid GMRES method the following problems arise:

- How can the iterative process be stopped correctly?
- What is the accuracy of the computed solution given by computer?
- How can we detect the numerical instabilities which may occur in Phase II of the program?

We observed that the use of the CADNA library allows us to solve these problems. It has been shown that the CADNA library with the optimal termination criterion and the appropriate test is able to stop the program as soon as a satisfactory solution is reached, to estimate the accuracy of the solution, to detect the numerical instabilities, to prevent an overflow which may occur, and to save computer time, because many useless operations and iterations are not performed. Consequently, the hybrid GMRES with the CADNA library is a robust and efficient tool for solving large nonsymmetric systems of linear equations.

Acknowledgements

We would like to thank Professor J. Vignes and Dr. J.M. Chesneaux for advise on many aspects of this work.

References

[1] J.M. Chesneaux, Study of the computing accuracy by using probabilistic approach, in: C. Ulrich, ed., *Contribution to Computer Arithmetic and Self Validating Numerical Methods* (IMACS, New Brunswick, NJ, 1990) 19–30.

[2] J.M. Chesneaux, CADNA, An ADA tool for round-off error analysis and for numerical debugging, in: *Proceedings Congress on ADA in Aerospace*, Barcelona (1990).

[3] J.M. Chesneaux, Descriptif d'utilisation du logiciel CADNA_F, MASI Report No. 92-32 (1992).

[4] J.M. Chesneaux and J. Vignes, Sur la robustesse de la méthode CESTAC, *C. R. Acad. Sci. Paris Sér. I Math.* 307 (1988) 855–860.
[5] J.M. Chesneaux and J. Vignes, Les fondements de l'arithmétique stochastique, *C. R. Acad. Sci. Paris Sér. I Math.* 315 (1992) 1435–1440.
[6] H. Elman, Y. Saad and P.E. Saylor, A hybrid Chebychev Krylov subspace algorithm for solving nonsymmetric systems of linear equations, *SIAM J. Sci. Statist. Comput.* 7 (1986) 840–855.
[7] H. Elman and R. Streit, Polynomial iteration for nonsymmetric indefinite linear systems, in: J.P. Hennert, ed., *Numerical Analysis*, Lecture Notes in Mathematics 1230 (Springer, Berlin, 1986).
[8] A. Feldstein and R. Goodman, Convergence estimates for the distribution of trailing digits, *J. ACM* 23 (1976).
[9] M.H. Gutknecht, Variants of BICGSTAB for matrices with complex spectrum, to appear.
[10] R.W. Hamming, On the distribution of numbers, *The Bell System Technical Journal* (1970).
[11] T.E. Hull and J.R. Swenson, Test of probabilistic models for propagation of round-off errors, *ACM Comm.* 9 (2) (1966).
[12] M. La Porte and J. Vignes, Evaluation statistique des erreurs numèriques dans les calculs sur ordinateur, *Numer. Math.* 23 (1974) 63–72.
[13] M. La Porte and J. Vignes, *Algorithmes Numériques—Analyse et Mise en Œuvre, Vol. 1, Arithmétique des Ordinateurs—Systèmes Linéaires* (Editions Technip, Paris, 1974).
[14] N.M. Nachtigal, S.C. Reddy and L.N. Trefethen, How fast are nonsymmetric matrix iterations?, *SIAM J. Matrix Anal. Appl.* 13 (3) (1992) 778–795.
[15] N.M. Nachtigal, L. Reichel and L.N. Trefethen, A hybrid GMRES algorithm for nonsymmetric linear systems, *SIAM J. Matrix Anal. Appl.* 13 (3) (1992) 796–825.
[16] L. Reichel, The application of Leja points to Richardson iteration and polynomial preconditioning, *Linear Algebra Appl.* 154–156 (1991) 389–414.
[17] Y. Saad and M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* 7 (3) (1986) 856–869.
[18] Y. Saad, Least squares polynomials in the complex plane and their use for solving nonsymmetric linear systems, *SIAM J. Numer. Anal.* 24 (1) (1987) 155–169.
[19] D.C. Smolarski and P. Saylor, An optimum iterative method for solving any linear system with a square matrix, *BIT* 28 (1988) 163–178.
[20] G.W. Stewart, *Introduction to Matrix Computations* (Academic Press, New York, 1973).
[21] J. Vignes, New methods for evaluating the validity of the results of mathematical computations, *Math. Comput. Simulation* 20 (4) (1978) 227–249.
[22] J. Vignes, Zéro mathématique et zéro informatique, *C. R. Acad. Sci. Paris Sér. I Math.* 303 (1986) 997–1000; also: *La Vie des Sciences* 4 (1) (1987) 1–13.
[23] J. Vignes, Contrôle et estimation stochastique des arrondis de calcul, *AFCET/Interfaces* 54 (1987) 3–11.
[24] J. Vignes, A stochastic arithmetic for reliable scientific computation, *Math. Comput. Simulation* 35 (1993) 233–261.