Numerical Algorithms 17 (1998) 105–119

The stable $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) algorithm with the CADNA library

Faezeh Toutounian

Department of Mathematics, Ferdowsi University of Mashhad, Mashhad, Iran
E-mail: toutouni@science2.um.ac.ir

Received 4 October 1996; revised 12 January 1998
Communicated by J. Vignes

The major drawback of the s-step iterative methods for nonsymmetric linear systems of equations is that, in the floating-point arithmetic, a quick loss of orthogonality of s-dimensional direction subspaces can occur, and consequently slow convergence and instability in the algorithm may be observed as s gets larger than 5. In [18], Swanson and Chronopoulos have demonstrated that the value of s in the s-step Orthomin(k) algorithm can be increased beyond $s = 5$ by orthogonalizing the s direction vectors in each iteration, and have shown that the $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) is stable for large values of s (up to $s = 16$). The subject of this paper is to show how by using the CADNA library, it is possible to determine a good value of s for $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k), and during the run of its code to detect the numerical instabilities and to stop the process correctly, and to restart the $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) in order to improve the computed solution. Numerical examples are used to show the good numerical properties.

Keywords: iterative methods, s-step methods, $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k), error propagation, CESTAC method, stochastic arithmetic, CADNA library

1. Introduction

Consider the linear system of equations

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

where A is a nonsymmetric matrix of order n. The s-step Orthomin(k) algorithm [8] can be applied to approximate the solution of (1). In the s-step Orthomin(k) iteration s directions $\{r_i, Ar_i, \ldots, A^{s-1}r_i\}$ are formed and are $A^{\mathrm{T}}A$-orthogonalized simultaneously to k of the preceding directions $\{p_j^1, \ldots, p_j^s\}$, $j = j_i, \ldots, i$, where $j_i = \max(0, i - k + 1)$. The norm of the residual $\|r_{i+1}\|_2$ is minimized simultaneously in all s new directions in order to obtain x_{i+1}. This method require less computational work and has better parallel properties than the standard Orthomin(k) algorithm. However, in finite arithmetic, for large $s > 5$ the loss of orthogonality between the direction subspaces leads to instability [8,9,11]. To alleviate the orthog-

onality loss in [18], Swanson and Chronopoulos have developed the $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) algorithm and shown that it is stable for large values of s (up to $s = 16$). The use of this method has to face a difficulty, which is how to choose the value of s. When s has a large value, the method has slow convergence because of the round-off errors propagation. So a reliable and efficient method for evaluating the round-off errors is necessary if one wants to determine a good value of s. In this paper, it is shown that the CESTAC method of La Porte and Vignes [16,17,19,21], which uses a random arithmetic and the CADNA library which implements it, are efficient tools for doing so. In section 2 we briefly describe the $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) algorithm and discuss the problems which exist in the implementation of this algorithm on a computer.

In section 3 we give a brief description of stochastic round-off analysis, the CESTAC method, and the CADNA software [4,22]. Section 4 is devoted to the use of the CESTAC method and CADNA library for determining a good value of s for $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k). Moreover, we will observe that by using the CADNA library and introducing the appropriate stopping criteria, it is possible, during the run of the code of the $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k), to detect the numerical instabilities and to stop correctly the iterative process, and to restart it in order to improve the computed solution. Some numerical results are given to show the good numerical properties.

2. $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) algorithm

In [8], Chronopoulos develops the s-step Orthomin(k) algorithm for nonsymmetric matrices with symmetric part $M = (A + A^{\mathrm{T}})/2$ positive definite or indefinite. In this method the s directions $\{r_i, \ldots, A^{s-1}r_i\}$ are formed and are $A^{\mathrm{T}}A$-orthogonalized simultaneously to k of the preceding directions $\{p_j^1, \ldots, p_j^s\}$, $j = j_i, \ldots, i$, where $j_i = \max(0, i-k+1)$. The norm of the residual $\|r_{i+1}\|_2$ is minimized simultaneously in all s new directions in order to obtain x_{i+1}. More details of the s-step Orthomin(k) algorithm can be found in [8]. The following notation facilitates the description of the algorithm:

- $W_i = [(Ap_i^j, Ap_i^l)]$, where $1 \leqslant j, l \leqslant s$;
- $\underline{a}_i = [a_i^1, \ldots, a_i^s]^{\mathrm{T}}$ (the steplengths in updating x_i);
- $\underline{m}_i = [(r_i, Ap_i^1), \ldots, (r_i, Ap_i^s)]^{\mathrm{T}}$;
- $\underline{c}_j^l = [(A^l r_{i+1}, Ap_j^1), \ldots, (A^l r_{i+1}, Ap_j^s)]^{\mathrm{T}}$;
- $\underline{b}_j^l = \{b_j^{l,m}\}_{m=1}^s$ for $j = j_i, \ldots, i$ and $l = 1, \ldots, s$, where $j_i = \max(0, i-k+1)$ (the coefficients to $A^{\mathrm{T}}A$-orthogonalize to the previous directions);
- $P_i = [p_i^1, \ldots, p_i^s]$ (the direction vectors);
- $R_i = [r_i, Ar_i, \ldots, A^{s-1}r_i]$ (the residuals).

A description of the s-step Orthomin(k) method can be given as follows:

Algorithm 2.1. s-step Orthomin(k)
Select x_0
$P_0 = [r_0 = b - Ax_0, Ar_0, \ldots, A^{s-1}r_0]$
For $i = 0$ **Until** convergence **Do**
Compute $\underline{m}_i$, W_i
Call Scalar1
$x_{i+1} = x_i + P_i\underline{a}_i$
$r_{i+1} = r_i - AP_i\underline{a}_i$
Compute $\underline{c}_j^i$, $j = j_i, \ldots, i$
Call Scalar2
Compute $R_{i+1} = [r_{i+1}, Ar_{i+1}, \ldots, A^{s-1}r_{i+1}]$
$P_{i+1} = R_{i+1} + \sum_{j=j_i}^{i} P_j[\underline{b}_j^l]_{l=1}^{s}$
Compute $AP_{i+1} = AR_{i+1} + \sum_{j=j_i}^{i} AP_j[\underline{b}_j^l]_{l=1}^{s}$
EndFor
Scalar1: Decomposes W_i and solves $W_i\underline{a}_i = \underline{m}_i$.
Scalar2: Solves $W_j\underline{b}_j^l = -\underline{c}_j^l$ for $j = j_i, \ldots, i$ and $l = 1, \ldots, s$, where $j_i = \max(0, i - k + 1)$.

The solution of the linear systems may cause a quick loss of orthogonality of the s-dimensional direction subspaces P_i because the matrix W_i may have a very large condition number. Numerical tests [9–11] have shown that the condition number of W_i is small for $s \leqslant 5$. One way to alleviate the orthogonality loss which can occur for large $s > 5$ is to $A^{\mathrm{T}}A$-orthogonalize the s direction vectors in each iteration. In [18], $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) was developed and shown to be stable for large values of s (up to $s = 16$). In this method the direction vectors within each subspace P_i are $A^{\mathrm{T}}A$-orthogonalized using the Modified Gram–Schmidt method. The linear systems need not be solved at each iteration since the W_i matrix is the identity matrix if P_i is perfectly $A^{\mathrm{T}}A$-orthogonalized. By using the notation $j_i = \max(0, i - k + 1)$ the algorithm can be described as follows:

Algorithm 2.2. $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k)
Select x_0
Compute $r_0 = b - Ax_0$
For $i = 0$ **Until** convergence **Do**
Compute $AP_i = [Ar_i, A^2r_i, \ldots, A^sr_i]$, and set $P_i = [r_i, Ar_i, \ldots, A^{s-1}r_i]$
If $(0 < i)$ **Then**
Compute $\underline{b}_j^l = [-(A^lr_i, Ap_j^1), \ldots, -(A^lr_i, Ap_j^s)]^{\mathrm{T}}$,
for $l = 1, \ldots, s$ and $j = j_{i-1}, \ldots, i - 1$
Compute $P_i = P_i + \sum_{j=j_{(i-1)}}^{i-1} P_j[\underline{b}_j^l]_{l=1}^{s}$
Compute $AP_i = AP_i + \sum_{j=j_{(i-1)}}^{i-1} AP_j[\underline{b}_j^l]_{l=1}^{s}$
EndIf

Apply the Modified Gram–Schmidt method to the matrix AP_i to obtain final AP_i and P_i

Compute $\underline{a}_i = [(r_i, Ap_i^1), \ldots, (r_i, Ap_i^s)]^{\mathrm{T}}$

$x_{i+1} = x_i + P_i \underline{a}_i$

$r_{i+1} = r_i - AP_i \underline{a}_i$

EndFor.

It is necessary to mention that the value of $A^{\mathrm{T}}A$-orthogonalizing the P_i direction vectors is that it allows s to become larger. This means the number of iterations is reduced with more work being done in each iteration, a situation that should make more efficient use of multiple processors. The additional work from the orthogonalization should be compensated by the enhanced parallel performance to obtain a faster algorithm. More details about the $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) method can be found in [18].

The main problem in the use of the $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) method, with floating-point arithmetic, is the choice of s. Let us consider the results of this method with different values of s and k for the examples 1–3 of section 4 (tables 2, 4 and 6, in which the number of iterations to convergence are given). These results clearly show that when s has a small or large value the method has slow convergence, and for each problem and each k there exists an s which minimizes the number of iterations to convergence. However, as mentioned above, the slow convergence of the method with large s values is due to the round-off error propagation. Hence, it is not possible to determine a good value of s without estimating the round-off errors propagation. In section 4, it is shown that by using the CADNA library, which is an efficient tool for doing so, we will be able to determine a good value of s.

Another problem is the choice of the value ε for the stopping criterion $\|r_i\|_2 \leqslant \varepsilon$. When ε is chosen too large, the iterative process is stopped too soon, and consequently the solution obtained has a poor accuracy. On the contrary, when ε is chosen small, it is possible, due to the numerical instabilities, that many useless iterations are performed without improving the accuracy of the solution. How can the iterative processs be stopped correctly, and restarted in order to improve the computed solution? The CADNA library is a precious tool for obtaining an answer to this question. In section 4 we will show that with the CADNA library it is possible, by including simple tests, to stop and to restart correctly the iterative process. The CADNA library, which allows to solve the above numerical problems, is a tool for automatic synchronous implementation of the CESTAC method of Vignes. In the following section we give a brief description of the CESTAC method, which is an efficient method for solving numerical problems such as those described above.

3. The CESTAC method

3.1. Basic ideas of the CESTAC method

Any result R provided by a computer always contains an error resulting from round-off error propagation. It has been proved [2] that a computed result R is modelled to the first order in 2^{-p} by the equation

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

where r is the exact result, α_i is the round-off error, and $u_i(d)$ are quantities depending exclusively on the data. 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.

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 is based on a probabilistic approach of the round-off error propagation, it has been presented in [13–15], the CESTAC method allows to estimate the round-off error on each result and consequently provides the accuracy of this result.

The basic idea of this method consists in performing the same code several times in order to propagate the round-off error differently each time. Several samples of R containing different round-off errors are then obtained. The first digits common to all the samples are significant and the others are not significant and represent the round-off error propagation. The aim is then to obtain these samples of R. They are obtained by the use of random arithmetic.

Indeed, each result r of any floating-point (FP) arithmetical operator is always bounded by two consecutive FP values R^- and R^+. The random arithmetic consists in randomly choosing either R^- or R^+ with a probability 0.5. Then when the same code is executed N times with a computer using this random arithmetic, for each result of any floating-point arithmetic, N different results R_i, $i = 1, \ldots, N$, will be provided. It has been proved [2,6] that, under certain hypotheses, these N results belong to a quasi-Gaussian distribution centered on the exact result r. So, in practice, the use of the CESTAC method consists in:

(i) Running in parallel N times ($N = 2$ or 3) the program with this new arithmetic. Consequently, for each result R of any floating-point arithmetic operation, a set of N computed results R_i, $i = 1, \ldots, N$, is obtained.

(ii) Taking the mean value $\overline{R} = (1/N)\sum_{i=1}^{N} R_i$ of the R_i as the computed result.

(iii) Using the Student distribution to estimate a confidence interval for R, and then compute the number $C_{\overline{R}}$ of significant digits of $\overline{R}$, defined by

$$C_{\overline{R}} = \log_{10}\left(\frac{\sqrt{N}|\overline{R}|}{s\tau_\beta}\right), \quad \text{with } s^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(R_i - \overline{R}\right)^2,$$

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

If $R_i = 0$, $i = 1, \ldots, N$, or if $C_{\overline{R}} \leqslant 0$, then $\overline{R}$ is an informatical zero denoted $\underline{0}$. This concept of informatical zero has been introduced by Vignes [20].

3.2. Stochastic arithmetic

By 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 realisations of a Gaussian random variable centered on the exact result. We can therefore define a new number, called stochastic number, and a new arithmetic, called stochastic arithmetic, applied to these numbers. We present below the main definitions and properties of this arithmetic. For more details see [7].

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 modelling of the concept of informatical zero proposed in [20].

Definition 2. $X \in S$ is a stochastic zero if and only if

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

In accordance with the concept of stochastic zero, two elements X and Y of S will be stochastically equal, denoted $X \,\mathrm{s}{=}\, Y$, if and only if their difference is a stochastic zero. For the order, a stochastic value X will be strictly greater than another stochastic value Y, denoted $X \,\mathrm{s}{>}\, Y$, if and only if it is significantly greater than the other. On the other hand, a stochastic value X will be greater than or equal to another stochastic value Y, denoted $X \,\mathrm{s}{\geqslant}\, Y$, if and only if it is greater than the other or their difference is a stochastic zero.

The stochastic elementary arithmetic operations are defined as operations between Gaussian independent random variables at the first order with respect to σ/μ. Stochastic operations are denoted $(\mathrm{s}+, \mathrm{s}-, \mathrm{s}*, \mathrm{s}/)$. For instance, $X_1 \,\mathrm{s}{-}\, X_2 = (\mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2)$.

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

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

$$a \ \mathrm{s}\leqslant b \ \Rightarrow \ a \ \mathrm{s}= b \text{ or } a \ \mathrm{s}< b;$$
$$a \ \mathrm{s}= b \text{ and } b \ \mathrm{s}< c \ \Rightarrow \ a \ \mathrm{s}\leqslant c;$$
$$a \ \mathrm{s}\leqslant b \text{ and } b \ \mathrm{s}< c \ \Rightarrow \ a \ \mathrm{s}\leqslant c.$$

$\underline{0}$ is absorbent for operation s$*$ and is the neutral element for operation s$+$. Let $x, y \in \mathbb{R}$ and $X, Y \in S$, respectively, be their representative. If $X \ \mathrm{s}< Y \Rightarrow x < y$.

As explained in [5], we recover with these definitions, especially the stochastic equality (the others only depend on it), the coherence between arithmetic operations and order relations that was lost in floating-point arithmetic.

In section 4, we shall see how the use of these new concepts has allowed us, during the run of the code of the $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) algorithm, to estimate the accuracy of any numerical result, to stabilize the code and to reduce numerical instabilities.

3.3. The CADNA library

The abbreviation CADNA [3,4,22] means *Control of Accuracy and Debugging for Numerical Applications*. This library implements the CESTAC method.

The first aim of this library is to enable us to estimate the effect of round-off errors on each result of the scientific codes. In addition, by identifying the notions of the theoretical number of significant digits of a stochastic number and the number of significant digits of an informatical result obtained by the CESTAC method, we can use all the concepts of the stochastic arithmetic on a computer.

Concretely, we assimilate

$$C_{\overline{R}} = \log_{10}\left(\frac{\sqrt{N}|\overline{R}|}{s\tau_\beta}\right)$$

to

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

because, when N is a small value (2 or 3), which is the case in practice, the values obtained with these two equations are very close values. This assimilation allows CADNA to use the definitions of the order relations and the equality relation of stochastic arithmetic. Therefore, CADNA allows to check the branching statements, which constitutes the second aim of this library.

The current version of CADNA has been written in Fortran 90, and is the version that we will describe here. CADNA is a library for programs written in Fortran. The use of CADNA requires compilation of the code with a Fortran 90 compiler and

linking the object code with the CADNA library. CADNA allows the programmers to dispose the new numerical type: *the stochastic type*. It contains the definitions of all the arithmetic operators, the ordering relations, as well as the elementary functions which are defined on the variables of the stochastic type. The control of the round-off errors is uniquely carried out on the variables of the stochastic type. In output, only the significant decimal digits are printed, so it is very easy to see the accuracy of the results. If the result is an informatical zero, the symbol @ appears.

The third aim of this library is to allow one to detect the numerical instabilities of the scientific codes, and to be considered as an efficient *numerical debugger*. We insist on the fact that it is a dynamic debugger which acts not on the correction of the writing of the program, but on the capacity of the computer used for providing the correct results in running the program.

Certainly, CADNA includes all the necessary controls for guaranteeing the reliability of the estimation of round-off errors that are provided by the CESTAC method. These controls, which are imposed by the theoretical study, allow the library self-validation, because it is able to determine, in a few moments, the conditions of the validity of the method which are no longer satisfied, if this is the case, and to warn the user of it.

The numerical debugger and self-validation of CESTAC are translated by the continuous detection of susceptible numerical instabilities which occur during the run of a program. The user is warned of these instabilities by the intermediary of a trace file called Cadna_stability_f90.lst which is managed by the CADNA library. Each time that an instability occurs, a trace numbered, in the form of a message, will be left in this file. These messages are classified into two categories: those which correspond to the self-validation of the CESTAC method, and those which uniquely correspond to the numerical debugger.

These two fundamental instabilities are:

- INSTABLE DIVISION, which means that during a division, the denominator is an informatical zero.
- INSTABLE TEST, which means that in evaluating of $A \leqslant B$, $A - B$ is an informatical zero. By applying the corresponding stochastic definition, the branch corresponding to the equality will be executed.
  But the user is warned of the fact that the mathematical answer of the test may be contrary to the informatical answer.

After each run, the user must consult the trace file and analyse the causes of each message which is left in the file. This can be done very easily with the help of the symbolic debugger. The traces are generated by an internal procedure in CADNA. By placing a “stop” instruction at the beginning of this procedure, under the symbolic debugger, the program will be stopped each time that a trace is written in the file. The statement of the call then provides the line of the source program which is responsible for the trace.

Finally, CADNA allows one to take into account the data errors in estimating the accuracy.

Information and a program for demonstration are available on the Internet site `http://www-masi-chpv.ibp.fr`.

4. Using the CADNA library in $A^{\mathrm{T}}A$ -orthogonal s-step Orthomin(k) method

As we have seen in section 2, in implementations of the $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) method two problems arise. The first one is:

- How to determine a good value of s?

Let us first consider tables 1, 3 and 5 of examples 1–3, respectively. These tables present the minimum number of significant digits of the norm of orthogonal direction vectors of P_0 (the first s-dimensional subspace) which are furnished by the CADNA library for differents values of s. It emerges from these results that, for each problem, this number begins to decrease from a certain s. When it has a small value for some s, a large error exists at the beginning of the iterative process and can lead to serious round-off errors, and then to slow convergence (see the results of tables 2, 4 and 6 of the mentioned examples which represent the number of iterations to convergence for different values of s). By noting this remark, it has been observed in experiments that, for double precision, we can obtain a good value of s by taking the highest value of s for which all the orthogonal direction vectors of P_0 have a norm with at least 10 significant digits. By using the CADNA library and increasing the value of s (for example, 4 by 4), it is very easy to determine such a value of s, because the number of significant digits of the norm of orthogonal direction vectors of P_0, for each s, can be furnished by the `cestac` function which exists in this library, and returns the number of significant digits of every stochastic variable.

Now, we consider the second problem, which is:

- How can the iterative process be stopped correctly?

As we mentioned in section 2, when we use the stopping criterion

$$\|r_i\|_2 \leqslant \varepsilon, \tag{2}$$

it is possible, due to numerical instabilities or/and stationarity, that this stopping criterion is never satisfied. So, we need to use additional termination criteria for stopping the process in the cases:

(i) The algorithm is stationary and can not converge.

(ii) The computer is not able to distinguish the vector r_i from the null vector and to improve the computed solution, because of the round-off error propagation.

As explained in [20,22], the stochastic arithmetic allows the development of two termination criteria for these cases.

In stochastic arithmetic, when the iterative process becomes stationary (before the stopping criterion (2) is satisfied), that is, the difference between two iterates is insignificant, the components of the vector $x_i - x_{i-1}$ are stochastic zeros. So, with the CADNA library which automatically implements the CESTAC method, and using the stopping criterion

$$\|x_i - x_{i-1}\|_1 = \underline{0}, \tag{3}$$

it is possible to stop the iterative process as soon as it becomes stationary.

On the other hand, in stochastic arithmetic, when the computer is unable to distinguish the vector r_i from the null vector (before the stopping criterion (2) is satisfied) and to improve the computed solution, because of the round-off error propagation, the components of r_i are stochastic zeros and a satisfactory informatical solution is available. So, with the CADNA library, and using the stopping criterion

$$\|r_i\|_2 = \underline{0}, \tag{4}$$

it is possible to stop the iterative process as soon as case (ii) occurs and a satisfactory informatical solution is reached.

It is clear that, in the above cases, in which the iterative process is stopped by criterion (3) or (4) before criterion (2) is satisfied, the computed solution will not be a solution with the desired accuracy ($\|r_i\|_2 \leqslant \varepsilon$) and it is necessary to improve it by an increment vector Δx_i. For doing this, we need the classical type value of the residual vector r_i of the computed solution x_i for solving the linear system $A\Delta x_i = r_i$ by restarting the iterative process. Fortunately, with the CADNA library, it suffices for obtaining the classical type value of r_i to use the `old_type` function which exists in this library, and which returns the corresponding classical type value of every stochastic variable.

We observe that, with the CADNA library, criteria (3) and (4) stop the iterative process as soon as cases (i) and (ii) occur, and make it possible to save computation time, because many useless iterations are avoided, to restart the iterative process in order to improve the satisfactory informatical solution which is furnished, and to obtain the solution with the desired accuracy. Consequently, with the CADNA library and using the termination criteria (2)–(4), and including the test for restarting the process in the cases in which the process is stopped by the stopping criterion (3) or (4), we can have a stable and efficient $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) algorithm with the value of s furnished by the method discussed above for solving the linear system and obtaining the desired approximate solution (with $\|r_i\|_2 \leqslant \varepsilon$).

Let us now present the examples and the results which we obtained by the FORTRAN code of the $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) method, with floating-point arithmetic for different values of s, and this code with the CADNA library, and the above tests, for the value of s furnished by the computer. Computations have been performed on a SUN4 computer in double precision. For floating-point arithmetic the stopping criterion was $\|r_i\|_2 \leqslant \varepsilon$ and the maximum number of iterations allowed set to 1000.

Example 1. We consider the constant-coefficient elliptic equation

$$-\Delta u + 2P_1 u_x + 2P_2 u_y = f, \tag{5}$$

which was described in [12], on the unit square $\Omega = \{(x, y) \mid 0 \leqslant x, y \leqslant 1\}$ with Dirichlet boundary conditions. Discretizing (5) on an $n_1 \times n_1$ grid gives rise to a sparse linear system of equations of order $n = n_1^2$. By using the second order centered differences for the first derivatives and the Laplacian, the coefficient matrix has the form

$$A = \begin{bmatrix} a & d & & & e & & \\ b & a & d & & & \ddots & \\ & b & a & & & & e \\ & & & & \ddots & & \\ c & & & \ddots & \ddots & & \\ & \ddots & & & & & d \\ & & & c & & b & a \end{bmatrix}.$$

After scaling the matrix and right hand side by h^2 ($h = 1/(n_1 + 1)$), the matrix entries are given by

$$a = 4, \quad b = -(1 + p_1), \quad c = -(1 + p_2), \quad d = -1 + p_1, \quad e = -1 + p_2,$$

where $p_1 = P_1 h$, $p_2 = P_2 h$. In our test we take $P_1 = 0$, $P_2 = 50$. The grid size is $h = 1/21$, leading to a problem of size 400. The right hand side is determined so that the solution x to the discrete system is 1 everywhere. This allows an easy verification of the results. With $\varepsilon = 10^{-9}$ and $x_0 = [0, \ldots, 0]^{\mathrm{T}}$ the results obtained are presented in tables 1 and 2. Table 1 contains the minimum number of significant digits of the norm of orthogonal direction vectors of P_0 for different values of s. Table 2 contains the number of iterations needed to satisfy the stopping criterion (2). For the CADNA library there are two numbers in this table, the first one, denoted by TN, presents the total number of iterations needed in the different runs of the iterative process. The second number, denoted by NR, presents the number of restarts of the iterative process.

The results presented in table 1 show that the highest value of s for which all the orthogonal direction vectors of P_0 have a norm with at least 10 significant digits is $s = 20$. Table 2 shows that the value $s = 20$ is a good value for this example. With this computed value $s = 20$, and the CADNA library, the solution was reached with

Table 1
The minimum number of significant digits of the norm of orthogonal direction vectors of P_0.

s	4	8	12	16	20	24	28	32	36	40
MIN	14	14	14	13	11	9	7	5	3	3

Table 2
The number of iterations to convergence.

	Double floating-point arithmetic										CADNA library	
s	4	8	12	16	20	24	28	32	36	40	TN	NR
$k=1$	35	27	19	11	7	6	7	9	18	22	5	1
$k=2$	46	23	10	7	6	17	16	19	27	34	5	1
$k=4$	52	12	8	8	8	53	57	65	88	90	5	1

only 5 iterations for all the values $k = 1, 2, 4$, which is less than those needed with floating-point arithmetic for different values of s. It must be noted that the process was stopped by the stopping criterion (4) at the 3rd iteration and restarted for improving the computed solution for which the norm of the residual was $\|r_3\|_2 = 0.484\text{E–}5$, $\|r_3\|_2 = 0.755\text{E–}6$, $\|r_3\|_2 = 0.755\text{E–}6$, for $k = 1, 2, 4$, respectively. We observe that, for this example, by using the CADNA library we could determine a good value of s ($s = 20$), and obtain the desired approximate solution (with $\|r_i\|_2 \leqslant 10^{-9}$) with the minimum number of iterations. So, for this example, the algorithm using the CADNA library is more efficient than that using floating-point arithmetic.

Example 2. We consider the linear system with

$$A = \begin{bmatrix} \alpha & 1 & & & \\ -1 & \alpha & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & & & 1 \\ & & & -1 & \alpha \end{bmatrix}, \qquad b = \begin{bmatrix} 1+\alpha \\ \alpha \\ \vdots \\ \alpha \\ \alpha - 1 \end{bmatrix},$$

which was described in [1], and dimension equal to 400. With $\alpha = 10^{-8}$, $\varepsilon = 10^{-5}$ and the initial vector $x_0 = [0, \ldots, 0]^{\mathrm{T}}$ the results are given in tables 3 and 4.

For this example the computed value of s is 28. From the results presented in table 4 it clearly appears that $s = 28$ is a good value for this example. With the CADNA library, the solution was reached with 18 iterations for $k = 1$, and 17 iterations for $k = 2, 4$. The iterative process was stopped by the stopping citerion (3) at the 16th iteration, and restarted in order to improve the computed solution for which the norm of the residual was $\|r_{16}\|_2 = 0.285\text{E–}3$, $\|r_{16}\|_2 = 0.163\text{E–}3$, $\|r_{16}\|_2 = 0.124\text{E–}3$, for $k = 1, 2, 4$, respectively. We observe that, with the CADNA library for $k = 1$ the total number of iterations to convergence is slightly greater than the smallest which

Table 3
The minimum number of significant digits of the norm of orthogonal direction vectors of P_0.

s	4	8	12	16	20	24	28	32	36	40
MIN	14	14	14	14	14	13	10	7	4	2

Table 4
The number of iterations to convergence.

	Double floating-point arithmetic										CADNA library	
s	4	8	12	16	20	24	28	32	36	40	TN	NR
$k=1$	100	50	36	25	20	19	17	15	24	29	18	1
$k=2$	100	50	37	25	20	20	18	17	23	39	17	1
$k=4$	100	50	39	25	20	22	20	21	37	58	17	1

corresponds to $s = 32$, and for $k = 2, 4$ this number, which is equal to 17, is less than or equal to those needed for different values of s. So, for this example, with the CADNA library the program is able to determine a good value of s ($s = 28$) and to furnish the desired approximate solution (with $\|r_i\|_2 \leqslant 10^{-5}$) with a reasonable number of iterations. Consequently, $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) performed with the CADNA library is an efficient tool for solving the linear system of this example.

Example 3. We consider the linear system with

$$A = \begin{bmatrix} 1 & & & & \alpha \\ & 2 & & & \\ & & \ddots & & \\ & & & n-1 & \\ & & & & n \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix},$$

which was described in [18,23] and has dimension equal to 100. With $\alpha = 2 \times 10^6$, $\varepsilon = 10^{-10}$ and the initial vector $x_0 = [0, \ldots, 0]^{\mathrm{T}}$ the results are listed in tables 5, 6.

For this example, the highest value of s for which all the orthogonal direction vectors of P_0 have a norm with at least 10 significant digits is $s = 8$. The results of

Table 5
The minimum number of significant digits of the norm of orthogonal direction vectors of P_0.

s	4	8	12	16	20	24	28	32	36	40
MIN	15	11	5	2	2	1	0	0	0	0

Table 6
The number of iterations to convergence. $*$ = problem reached iteration count limit.

	Double floating-point arithmetic					CADNA library	
s	4	8	12	16	20	TN	NR
$k=1$	29	14	10	92	*	13	2
$k=2$	27	15	19	85	*	14	3
$k=4$	22	74	68	131	*	15	3

table 6 show that, with the CADNA library, the behavior is similar to that in example 2. It must be noted that the process has been stopped by the stopping criterion (4) two times for $k = 1$, and three times for $k = 2, 4$.

5. Conclusion

In this paper we have seen, in floating-point arithmetic, that the $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) method has to face the two inherent difficulties:

- How to determine a good value of s?
- How can the iterative process be stopped correctly?

We observed that the use of the CADNA library allows us to solve these problems. It has been shown that it is possible, on the one hand, by using the number of significant digits of the norm of orthogonal direction vectors of P_0, furnished by this library, to determine a good value of s, and, on the other hand, by using the appropriate stopping criteria which use the informatical zero, to stop correctly the iterative process and to save computer time, because many useless iterations are not performed. We have seen by restarting the iterative process, that it is possible to improve the computed solution. The numerical experiments show that the total number of iterations taken in the different runs of the iterative process is a reasonable number versus those needed for different values of s with floating-point arithmetic. In short, the two problems encountered in the use of the $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) method have been solved with the CADNA library. Consequently, the $A^{\mathrm{T}}A$-orthogonal s-step Orthomin(k) algorithm with the CADNA library is a robust and efficient tool for solving large nonsymmetric systems of linear equations.

Acknowledgements

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

References

[1] P.N. Brown, A theoretical comparison of the Arnoldi and GMRES algorithms, SIAM J. Sci. Statist. Comput. 12 (1991) 58–78.
[2] J.M. Chesneaux, Study of the computing accuracy by using probabilistic approach, in: *Contribution to Computer Arithmetic and Self Validating Numerical Methods*, ed. C. Ulrich (IMACS, New Brunswick, NJ, 1990) pp. 19–30.
[3] J.M. Chesneaux, CADNA, an ADA tool for round-off error analysis and for numerical debugging, in: *Proc. Congress on ADA in Aerospace*, Barcelona (1990).
[4] J.M. Chesneaux, Descriptif d'utilisation du logiciel CADNA_F, MASI Report, No. 92-32 (1992).
[5] J.M. Chesneaux, The equality relations in scientific computing, Numer. Algorithms 7 (1994) 129–143.

[6] 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.
[7] J.M. Chesneaux and J. Vignes, Les fondements de l'arithmétique stochastique, C. R. Acad. Sci. Paris Sér. I Math. 315 (1992) 1435–1440.
[8] A.T. Chronopoulos, s-step iterative methods for (non)symmetric (in)definite linear systems, SIAM J. Numer. Anal. 28(6) (1991) 1776–1789.
[9] A.T. Chronopoulos and C.W. Gear, s-step iterative methods for symmetric linear systems, J. Comput. Appl. Math. 25 (1989) 153–168.
[10] A.T. Chronopoulos and C.W. Gear, Implementation of preconditioned s-step conjugate gradient methods on a multiprocessor system with memory hierarchy, Parallel Comput. 11 (1989) 37–53.
[11] A.T. Chronopoulos and S.K. Kim, The s-step Orthomin and s-step GMRES implemented on parallel computers, in: *SIAM Conf. on Iterative Methods*, The Copper Mountain, CO (April 1–5, 1990), University of Minnesota, Dept. of Computer Science, Tech. Report 90-15, Minneapolis, MN (1990).
[12] H.C. Elman, A stability analysis of incomplete LU factorizations, J. Math. Comput. 47(175) (1986) 191–217.
[13] A. Feldstein and R. Goodman, Convergence estimates for the distribution of trailing digits, J. ACM 23 (1976) 287–297.
[14] R.W. Hamming, On the distribution of numbers, Bell Syst. Tech. J. 49 (1970) 1609–1625.
[15] T.E. Hull and J.R. Swenson, Test of probabilistic models for propagation of round-off errors, Commun. ACM 9(2) (1966).
[16] M. La Porte and J. Vignes, Evaluation statistique des erreurs numériques dans les calculs sur ordinateur, Numer. Math. 23 (1974) 63–72.
[17] M. La Porte and J. Vignes, *Algorithmes Numériques – Analyse et Mise en Œuvre*, Vol. 1, *Arithmétique des Ordinateurs – Systèmes Linéaires* (Editions Technip, Paris, 1974).
[18] C.D. Swanson and A.T. Chronopoulos, Orthogonal s-step methods for nonsymmetric linear systems of equations, in: *ACM Int. Conf. on Supercomputing* (July 19–23, 1992) pp. 456–464.
[19] J. Vignes, New methods for evaluating the validity of the results of mathematical computations, Math. Comput. Simulation 20(4) (1978) 227–249.
[20] 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.
[21] J. Vignes, Contrôle et estimation stochastique des arrondis de calcul, AFCET/Interfaces 54 (1987) 3–11.
[22] J. Vignes, A stochastic arithmetic for reliable scientific computation, Math. Comput. Simulation 35 (1993) 233–261.
[23] H. Walker, Implementation of the GMRES method using Householder transformations, SIAM J. Sci. Statist. Comput. 9 (1988) 152–163.