Optimal Iterate of the Power and Inverse Iteration Methods

Davod Khojasteh Salkuyeh[†] and Faezeh Toutounian[‡]

[†]Department of Mathematics, University of Mohaghegh Ardabili,
P.O. Box. 179, Ardabil, Iran
E-mail: khojaste@uma.ac.ir

[‡]Department of Mathematics, School of Mathematical Sciences, Ferdowsi University of Mashhad,
P.O. Box 1159-91775, Mashhad, Iran
E-mail: toutouni@math.um.ac.ir

Abstract

The power method is an algorithm for computing the largest eigenvalue of matrix A in absolute value. To find the other eigenvalues one can apply the power method to the matrix $(A - \sigma I)^{-1}$ for some shift σ. This scheme is called the inverse iteration method. Both of these two methods produce a convergence sequence and the limit is approximated by one of the iterates. In the chosen iterate, it may be difficult to estimate the global error, consisting of the truncation error and the round-off error. In this paper, by using the CESTAC method and the CADNA library, we propose a method for computing the optimal iterate, the iterate for which the global error is minimal. In the proposed method the accuracy of the computed eigenvalue may also be estimated. Some numerical examples are given to show the efficiency of the method.

AMS Subject Classification : 65F15, 65G50.
Keywords: power method, inverse iteration method, round-off error, common significant digits, CESTAC method, CADNA library.

1. Introduction

Many of the problems in scientific computing involve the computation of convergence sequences and the limit of the sequence is approximated by one of the iterates. Since the finite precision computations affect the stability of iterative algorithms and the accuracy of computed solutions, it is difficult to estimate the global error, consisting of the truncation error and the round off error, in the chosen iterate. In [11, 16], the authors showed that, by using the CESTAC method [17, 19] and the CADNA library [2, 6, 10], the optimal iterate, i.e., the approximation for which the global error is minimized, can be dynamically computed. In this paper we show how, by using this library, the optimal iterate of the power method and inverse iteration method can be dynamically computed and its accuracy can be estimated.

The power method is an iterative method for computing the largest eigenvalue of a matrix in absolute value. Let A be an $n \times n$ diagonalizable matrix and its eigenvalues satisfy $|\lambda_1| >$

$|\lambda_2| \geq \ldots \geq |\lambda_n|$. Let also x_i be the eigenvector of A corresponding to the eigenvalue λ_i. Then the power method is an iterative method which computes an approximation of the eigenpair (λ_1, x_1) of A. This algorithm may run as Algorithm 1 [8, 9, 22].

Algorithm 1. Power method

1. Choose ϵ and vector v_0 such that $\|v_0\|_2 = 1$
2. $\tilde{\lambda}_0 = v_0^T A v_0$
3. For $m = 1, 2, \ldots$ Do
4. $\quad w_m = A v_{m-1}$
5. $\quad v_m = w_m / \|w_m\|_2$ (approximate eigenvector)
6. $\quad \tilde{\lambda}_m = v_m^T A v_m$ (approximate eigenvalue)
7. $\quad$ If $|\tilde{\lambda}_m - \tilde{\lambda}_{m-1}| \leq \epsilon$, then stop
8. EndDo

In the inverse iteration method the power method is applied to $(A - \sigma I)^{-1}$, where σ is called a shift and I is the identity matrix. In this case the method converges to the eigenvalue closest to σ, rather than just λ_1. This algorithm may run as Algorithm 2 [8, 9, 22].

Algorithm 2. Inverse iteration method

1. Choose ϵ, σ and vector v_0 such that $\|v_0\|_2 = 1$
2. $\tilde{\lambda}_0 = v_0^T A v_0$
3. For $m = 1, 2, \ldots$ Do
4. $\quad w_m = (A - \sigma I)^{-1} v_{m-1}$
5. $\quad v_m = w_m / \|w_m\|_2$ (approximate eigenvector)
6. $\quad \tilde{\lambda}_m = v_m^T A v_m$ (approximate eigenvalue)
7. $\quad$ If $|\tilde{\lambda}_m - \tilde{\lambda}_{m-1}| \leq \epsilon$, then stop
8. EndDo

In step 4 of this algorithm one can solve $(A - \sigma I) w_m = v_{m-1}$ for w_m, instead of computing $(A - \sigma I)^{-1}$. For solving $(A - \sigma I) w_m = v_{m-1}$ one may use the LU factorization of $A - \sigma I$.

As we observe, in both algorithms, a stopping criterion in step 7 is used. If the parameter ϵ is chosen too large (e.g. $\epsilon = 10^{-5}$) then the iterative method is broken off too early and the computed solution has poor accuracy. On the contrary when ϵ is chosen too small (e.g. $\epsilon = 10^{-16}$) it is possible, due to the numerical instabilities, that many useless iterations are performed without improving the accuracy of the iterate. In practice it is absolutely impossible to choose correctly the value of the parameter ϵ which minimizes the global error.

Assume that each of the algorithms converges to the eigenvalue λ (for Algorithm 1, $\lambda = \lambda_1$) of matrix A. We state a relation between the number of common significant digits of two successive approximations $\widetilde{\lambda}_m$ and $\widetilde{\lambda}_{m+1}$ produced by one of the algorithms and the common significant digits between $\widetilde{\lambda}_m$ and λ. Then with the aid of the CESTAC method and the

CADNA library we propose an optimal stopping criterion which is able to stop correctly the iterative process, and to minimize the global error.

This paper is organized as follows. In section 2, the main results are given. Section 3 is devoted to a brief description of the stochastic round-off error analysis, the CESTAC method and the CADNA library. In section 4, the optimal stopping criterion is presented. Some numerical examples are given in section 4. Section 5 is devoted to some concluding remarks.

2. Theoretical description

We begin this section with the following lemma.

Lemma 1. *Assume that the matrix A has real eigenvalues $\lambda_i,\ i = 1, 2, \ldots, n$, such that $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|$ and a complete set of orthonormal eigenvectors $x_1, x_2, \ldots, x_n$. Let $v_0 \in \mathbb{R}^n$ be the starting vector and $\alpha_1 = v_0^T x_1 \neq 0$. Then the approximate eigenvalue computed by the power method at iteration m, $\widetilde{\lambda}_m$, satisfies*

$$\widetilde{\lambda}_m - \lambda_1 = \sum_{i=2}^{n} (\lambda_i - \lambda_1)\theta_i^2 (\frac{\lambda_i}{\lambda_1})^{2m} + \mathcal{O}((\frac{\lambda_2}{\lambda_1})^{4m-1}), \tag{1}$$

where $\theta_i \in \mathbb{R}$, $i = 2, \ldots, n$.

Proof. Let $\alpha_i = v_0^T x_i,\ i = 1, 2, \ldots, n$. Then the Fourier expansion of v_0 takes the form (see [13], page 299)

$$v_0 = \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n. \tag{2}$$

It can be easily verified that vector v_m can be written as

$$v_m = \frac{A^m v_0}{\|A^m v_0\|_2}.$$

From (2), we have

$$A^m v_0 = \sum_{i=1}^{n} \alpha_i \lambda_i^m x_i.$$

Using the orthonormality of the x_i's, we conclude

$$\begin{aligned}
\widetilde{\lambda}_m &= v_m^T A v_m \\
&= \frac{\sum_{i=1}^n \alpha_i \lambda_i^m x_i^T \sum_{i=1}^n \alpha_i \lambda_i^{m+1} x_i}{\sum_{i=1}^n \alpha_i^2 \lambda_i^{2m}} \\
&= \frac{\sum_{i=1}^n \alpha_i^2 \lambda_i^{2m+1}}{\sum_{i=1}^n \alpha_i^2 \lambda_i^{2m}} \\
&= \lambda_1 \frac{1 + \sum_{i=2}^n \theta_i^2 (\frac{\lambda_i}{\lambda_1})^{2m+1}}{1 + \sum_{i=2}^n \theta_i^2 (\frac{\lambda_i}{\lambda_1})^{2m}}, \qquad \text{where } \theta_i = \alpha_i/\alpha_1
\end{aligned}$$

$$= \lambda_1(1+\sum_{i=2}^{n}\theta_i^2(\frac{\lambda_i}{\lambda_1})^{2m+1})(1-\sum_{j=1}^{\infty}(-1)^{j+1}(\sum_{i=2}^{n}\theta_i^2(\frac{\lambda_i}{\lambda_1})^{2m})^j)$$
$$= \lambda_1+\sum_{i=2}^{n}(\lambda_i-\lambda_1)\theta_i^2(\frac{\lambda_i}{\lambda_1})^{2m}+\mathcal{O}((\frac{\lambda_2}{\lambda_1})^{4m-1}).$$

From this the desired relation is easily obtained. □

A similar lemma can be established for the inverse iteration method as following.

Lemma 2. *Assume that the matrix A has real eigenvalues λ_i, $i = 1, 2, \ldots, n$, such that $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$ and a complete set of orthonormal eigenvectors $x_1, x_2, \ldots, x_n$. Let $v_0 \in \mathbb{R}^n$ be the starting vector and λ_J be the closest eigenvalue to the shift σ and λ_K be the second closest one, that is, $0 \neq |\lambda_J - \sigma| < |\lambda_K - \sigma| \leq |\lambda_j - \sigma|$ for each $j \neq J$. Moreover suppose that $v_0^T x_J \neq 0$. Then the approximate eigenvalue computed by the inverse iteration method at iteration m, $\widetilde{\lambda}_m$, satisfies*

$$\widetilde{\lambda}_m - \lambda_J = \sum_{i=1, i\neq J}^{n}(\lambda_i-\lambda_J)\gamma_i^2(\frac{\lambda_J-\sigma}{\lambda_i-\sigma})^{2m}+\mathcal{O}((\frac{\lambda_J-\sigma}{\lambda_K-\sigma})^{4m-1}), \tag{3}$$

where $\gamma_i \in \mathbb{R}$, $i \neq J$.

Proof. Obviously the vector v_m computed by the inverse iteration method can be written as

$$v_m = \frac{(A-\sigma I)^{-m}v_0}{\|(A-\sigma I)^{-m}v_0\|_2}.$$

From (2), it follows that

$$(A-\sigma I)^{-m}v_0 = \sum_{i=1}^{n}\alpha_i(\lambda_i-\sigma)^{-m}x_i.$$

Using the orthonormality of the x_i's, we have

$$\begin{aligned}
\widetilde{\lambda}_m &= v_m^T A v_m \\
&= \frac{\sum_{i=1}^n \alpha_i(\lambda_i-\sigma)^{-m}x_i^T \sum_{i=1}^n \alpha_i\lambda_i(\lambda_i-\sigma)^{-m}x_i}{\sum_{i=1}^n \alpha_i^2(\lambda_i-\sigma)^{-2m}} \\
&= \frac{\sum_{i=1}^n \alpha_i^2\lambda_i(\lambda_i-\sigma)^{-2m}}{\sum_{i=1}^n \alpha_i^2(\lambda_i-\sigma)^{-2m}} \\
&= \lambda_J\frac{1+\sum_{i=1,i\neq J}^n \gamma_i^2\frac{\lambda_i}{\lambda_J}(\frac{\lambda_i-\sigma}{\lambda_J-\sigma})^{-2m}}{1+\sum_{i=1,i\neq J}^n \gamma_i^2(\frac{\lambda_i-\sigma}{\lambda_J-\sigma})^{-2m}}, \qquad \text{where } \gamma_i = \alpha_i/\alpha_J \\
&= \lambda_J(1+\sum_{i=1,i\neq J}^{n}\gamma_i^2\frac{\lambda_i}{\lambda_J}(\frac{\lambda_J-\sigma}{\lambda_i-\sigma})^{2m})(1-\sum_{j=1}^{\infty}(-1)^{j+1}(\sum_{i=1,i\neq J}^{n}\gamma_i^2(\frac{\lambda_J-\sigma}{\lambda_i-\sigma})^{2m})^j)
\end{aligned}$$

$$= \lambda_J + \sum_{i=1, i\neq J}^{n} (\lambda_i - \lambda_J)\gamma_i^2 (\frac{\lambda_J - \sigma}{\lambda_i - \sigma})^{2m} + \mathcal{O}((\frac{\lambda_J - \sigma}{\lambda_K - \sigma})^{4m-1}).$$

This completes the proof. $\square$

For later use we recall the following definition [12, 1].

Definition 1. *Let p and q be two real numbers, the number of exact significant digits that are common to p and q can be defined in $(-\infty, +\infty)$ by*

1. *for $p \neq q$,*
$$C_{p,q} = \log_{10} \mid \frac{p+q}{2(p-q)} \mid .$$
2. *$\forall p \in \mathbb{R},\ C_{p,p} = +\infty$.*

Theorem 1. *Suppose that all of the assumptions of Lemma 1 hold. Then*

$$C_{\widetilde{\lambda}_m, \widetilde{\lambda}_{m+1}} = C_{\widetilde{\lambda}_m, \lambda_1} + \log_{10} \frac{1}{1 - (\frac{\lambda_2}{\lambda_1})^2} + \mathcal{E}_m, \tag{4}$$

where $\mathcal{E}_m \to 0$ as $m \to \infty$.

Proof. Obviously there exists a natural number r with $2 \leq r \leq n$ such that

$$|\lambda_2| = |\lambda_3| = \ldots = |\lambda_r| > |\lambda_{r+1}| \geq \ldots \geq |\lambda_n|.$$

Then the relation (1) can be written as

$$\widetilde{\lambda}_m - \lambda_1 = S(\frac{\lambda_2}{\lambda_1})^{2m} + \mathcal{W}_m, \tag{5}$$

in which

$$\begin{aligned} S &= \sum_{i=2}^{r} (\lambda_i - \lambda_1)\theta_i^2, \\ \mathcal{W}_m &= \sum_{i=r+1}^{n} (\lambda_i - \lambda_1)\theta_i^2 (\frac{\lambda_i}{\lambda_1})^{2m} + \mathcal{O}((\frac{\lambda_2}{\lambda_1})^{4m-1}). \end{aligned}$$

Using Eq. (5), we obtain

$$\begin{aligned} \widetilde{\lambda}_m + \lambda_1 &= 2\lambda_1 + S(\frac{\lambda_2}{\lambda_1})^{2m} + \mathcal{W}_m, \\ \widetilde{\lambda}_m - \widetilde{\lambda}_{m+1} &= (1 - (\frac{\lambda_2}{\lambda_1})^2)(S(\frac{\lambda_2}{\lambda_1})^{2m} + \mathcal{V}_m), \end{aligned} \tag{6}$$

$$\widetilde{\lambda}_m + \widetilde{\lambda}_{m+1} = 2\lambda_1 + (1 + (\frac{\lambda_2}{\lambda_1})^2)(S(\frac{\lambda_2}{\lambda_1})^{2m} + \mathcal{U}_m),$$

with

$$\begin{aligned}
\mathcal{V}_m &= \sum_{i=r+1}^{n} \frac{\lambda_1^2 - \lambda_i^2}{\lambda_1^2 - \lambda_2^2}(\lambda_i - \lambda_1)\theta_i^2(\frac{\lambda_i}{\lambda_1})^{2m} + \mathcal{O}((\frac{\lambda_2}{\lambda_1})^{4m-1}), \\
\mathcal{U}_m &= \sum_{i=r+1}^{n} \frac{\lambda_1^2 + \lambda_i^2}{\lambda_1^2 + \lambda_2^2}(\lambda_i - \lambda_1)\theta_i^2(\frac{\lambda_i}{\lambda_1})^{2m} + \mathcal{O}((\frac{\lambda_2}{\lambda_1})^{4m-1}).
\end{aligned}$$

According to the latter relations and Eq. (5) we see that

$$\log_{10}|\frac{\widetilde{\lambda}_m + \widetilde{\lambda}_{m+1}}{\widetilde{\lambda}_m + \lambda_1}| = \log_{10}|\frac{2\lambda_1 + (1 + (\frac{\lambda_2}{\lambda_1})^2)(S(\frac{\lambda_2}{\lambda_1})^{2m} + \mathcal{U}_m)}{2\lambda_1 + S(\frac{\lambda_2}{\lambda_1})^{2m} + \mathcal{W}_m}|.$$

Obviously, this expression tends to zero as $m \to \infty$, since $\mathcal{W}_m, \mathcal{U}_m \to 0$. On the other hand

$$\begin{aligned}
\log_{10}|\frac{\widetilde{\lambda}_m - \lambda_1}{\widetilde{\lambda}_m - \widetilde{\lambda}_{m+1}}| &= \log_{10}|\frac{S(\frac{\lambda_2}{\lambda_1})^{2m} + \mathcal{W}_m}{(1 - (\frac{\lambda_2}{\lambda_1})^2)(S(\frac{\lambda_2}{\lambda_1})^{2m} + \mathcal{V}_m)}| \\
&= \log_{10}\frac{1}{1 - (\frac{\lambda_2}{\lambda_1})^2} + \log_{10}|\frac{S + (\frac{\lambda_1}{\lambda_2})^{2m}\mathcal{W}_m}{S + (\frac{\lambda_1}{\lambda_2})^{2m}\mathcal{V}_m}|.
\end{aligned}$$

The second term of the right hand side of the latter equation tends to zero as $m \to \infty$. Since

$$\begin{aligned}
(\frac{\lambda_1}{\lambda_2})^{2m}\mathcal{W}_m &= \sum_{i=r+1}^{n} (\lambda_i - \lambda_1)\theta_i^2(\frac{\lambda_i}{\lambda_2})^{2m} + \mathcal{O}((\frac{\lambda_2}{\lambda_1})^{2m-1}) \to 0, \\
(\frac{\lambda_1}{\lambda_2})^{2m}\mathcal{V}_m &= \sum_{i=r+1}^{n} \frac{\lambda_1^2 - \lambda_i^2}{\lambda_1^2 - \lambda_2^2}(\lambda_i - \lambda_1)\theta_i^2(\frac{\lambda_i}{\lambda_2})^{2m} + \mathcal{O}((\frac{\lambda_2}{\lambda_1})^{2m-1}) \to 0,
\end{aligned}$$

as $m \to \infty$, regarding that $|\lambda_i/\lambda_2| < 1,\ i = r+1, \ldots, n$. Hence from Definition 1 we have

$$\begin{aligned}
C_{\widetilde{\lambda}_m, \widetilde{\lambda}_{m+1}} - C_{\widetilde{\lambda}_m, \lambda_1} &= \log_{10}|\frac{\widetilde{\lambda}_m + \widetilde{\lambda}_{m+1}}{\widetilde{\lambda}_m + \lambda_1}| + \log_{10}|\frac{\widetilde{\lambda}_m - \lambda_1}{\widetilde{\lambda}_m - \widetilde{\lambda}_{m+1}}| \\
&= \log_{10}\frac{1}{1 - (\frac{\lambda_2}{\lambda_1})^2} + \mathcal{E}_m,
\end{aligned}$$

where $\mathcal{E}_m \to 0$ as $m \to \infty$. $\square$

Theorem 2. *Suppose that all of the assumptions of Lemma 2 hold. Then*

$$C_{\widetilde{\lambda}_m, \widetilde{\lambda}_{m+1}} = C_{\widetilde{\lambda}_m, \lambda_J} + \log_{10}\frac{1}{1 - (\frac{\lambda_J - \sigma}{\lambda_K - \sigma})^2} + \mathcal{E}_m, \tag{7}$$

where $\mathcal{E}_m \to 0$ *as* $m \to \infty$.

Proof. The proof of this theorem is quite similar to that of Theorem 1 and is omitted. □

Let us define the scalars $\alpha = (\frac{\lambda_2}{\lambda_1})^2$ and $\alpha = (\frac{\lambda_J - \sigma}{\lambda_K - \sigma})^2$ for the power and the inverse iteration methods, respectively. In both methods we have $0 \leq \alpha < 1$. Eqs. (4) and (7) show that, if the convergence zone is reached, i.e., $|\mathcal{E}_m| \ll 1$, then the last term in these equations becomes negligible. In this case, from the significant digits in common between $\widetilde{\lambda}_m$ and $\widetilde{\lambda}_{m+1}$, we can deduce the significant digits in common between $\widetilde{\lambda}_m$ and λ_1 in the power method (λ_J in the inverse iteration method). If $\alpha = 0$ and the convergence zone is reached, then the significant digits in common between $\widetilde{\lambda}_m$ and $\widetilde{\lambda}_{m+1}$ are also in common with λ_1 in the power method (λ_J in the inverse iteration method). If $0 < \alpha < 1$, as mentioned in [11], there exists a natural number k such that $0 < \alpha \leq 1 - 10^{-k}$. Therefore, we have $0 < \log_{10} \frac{1}{1-\alpha} \leq k$. Hence, if the convergence zone is reached, then the significant digits in common between $\widetilde{\lambda}_m$ and $\widetilde{\lambda}_{m+1}$ are also in common with λ_1 in the power method (λ_J in the inverse iteration method) , up to k digits. As a remark, if $0 < \alpha \leq \frac{1}{2}$, then $0 < \log_{10} \frac{1}{1-\alpha} \leq 1$. In this case, if the convergence zone is reached, the significant digits in common between $\widetilde{\lambda}_m$ and $\widetilde{\lambda}_{m+1}$ are also in common with λ_1 in the power method (λ_J in the inverse iteration method), up to one.

From the above discussion, we observe that for estimating the accuracy of the computed eigenvalue $\widetilde{\lambda}_m$, an approximation of $\log_{10} \frac{1}{1-\alpha}$ is needed. In the power method (inverse iteration method), for computing the significant digits of λ_1 (λ_J), an approximation of the eigenvalue λ_2 ($\lambda_K - \sigma$) should be computed. Among the methods for this purpose is the Wielandt deflation technique [14], but is too expensive. In continuation, we show that an approximation of $1 - \alpha$ can be obtained easily. For the power method, from Eqs. (5), (6), we have

$$\frac{\widetilde{\lambda}_m - \widetilde{\lambda}_{m+1}}{\widetilde{\lambda}_m - \lambda_1} = (1 - (\frac{\lambda_2}{\lambda_1})^2) \frac{S(\frac{\lambda_2}{\lambda_1})^{2m} + \mathcal{V}_m}{S(\frac{\lambda_2}{\lambda_1})^{2m} + \mathcal{W}_m} \to 1 - (\frac{\lambda_2}{\lambda_1})^2,$$

as $m \to \infty$. So, for large enough m, we have

$$1 - \alpha \approx \frac{\widetilde{\lambda}_m - \widetilde{\lambda}_{m+1}}{\widetilde{\lambda}_m - \lambda_1}.$$

Since, for our propose a moderate approximation of $1 - \alpha$ would be sufficient, we can obtain an approximate value β_m for $1 - \alpha$ by means of

$$\beta_m = \frac{\widetilde{\lambda}_m - \widetilde{\lambda}_{m+1}}{\widetilde{\lambda}_m - \lambda_*}, \tag{8}$$

where m is large enough and λ_* is the optimal iterate which is furnished by the computer. It can be easily verified that the relation (8) is valid for the inverse iteration method as well. In section 4 we show how, by using the CADNA library and the optimal termination criterion, it is possible to determine the optimal iterate λ_* and to compute the approximate value β_m in order to estimate the exact significant digits of λ_*.

These theoretical results have been obtained by taking into account only the truncation error on two successive iterates of the power and inverse power iteration methods. However computed results are also affected by round-off error propagation. In continuation we describe how round-off errors can be estimated with a probabilistic approach in order to determine the exact significant digits of any computed result.

3. 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. In this section, we briefly review the CESTAC (Control et Estimation Stochastique des Arrondis de Calcul) method [17, 19] which is the basis of DSA [21] and define the concept of computational zero. The stochastic order relations of DSA are presented. With the DSA (Discrete Stochastic Arithmetic), which is the joint use of the synchronous implementation of CESTAC method and the stochastic order relations, it is possible to estimate the accuracy of the results provided by a computer, to detect the numerical instabilities occurring during the run of a scientific code, and to check the branchings that exist in the code.

3.1. Brief recall of the CESTAC method and its implementation

The basic idea of the method is defined in [19, 20] and consists in:

- synchronously performing the same code N times with a different round-off error propagation for each run.

- estimating the common part of these results and to consider that this part is representative of the exact result.

In practice, these different round-off error propagations are obtained in using random rounding mode.

Indeed, each result r of a floating-point operation which is not an exact floating-point value is always bounded by two floating-points values R^- and R^+, each of them being so representative of the exact result.

The random rounding consists at the level of each floating-point operation or assignment to choose as result randomly with an equal probability either R^- or R^+. Then when the same code is executed N times with a computer using this random rounding, N results $R_k, k = 1, \ldots, N$ are obtained. It has been proved in [5, 7] that, under some hypotheses, these N results belong to a quasi-Gaussian distribution centered on the exact result r. So, in practice, by considering the mean value $\overline{R}$ of the R_k as the computed result, and using Student's test, it is possible to obtain a confidence interval of $\overline{R}$ with a probability $(1 - \beta)$ and then to estimate the number of exact significant digits of $\overline{R}$ by the formula (9)

$$C_{\overline{R}} = \log_{10}(\sqrt{N}|\overline{R}|/\tau_\beta \sigma), \tag{9}$$

with $\overline{R} = (1/N)\Sigma_{i=1}^{N} R_i$ and $\sigma^2 = \frac{1}{N-1}\Sigma_{i=1}^{N}(R_i - \overline{R})^2$. τ_β is the value of the Student distribution for $N-1$ degrees of freedom and a probability level $1-\beta$. In practice $N = 3, \beta = 0.05$ and then $\tau_\beta = 4.4303$.

The result provided by equation (9) is reliable if the hypotheses underlying the method hold in practice. It has been proved that [5, 7, 16], these hypotheses hold when:

1) The operands of any multiplication are both significant.

2) The divisor of any division is significant.

It is then absolutely necessary during the run of a code to control the points 1) and 2). This control is done with the concept of computational zero also named computational zero or computed zero [18].

Definition 2. Each result provided by CESTAC method is an computational zero denoted by @.0 iff one of the two conditions holds:

1) $\forall i, i = 1, \ldots, N, R_i = 0.$

2) $C_{\overline{R}} \leq 0$, ($C_{\overline{R}}$ obtained with equation (9)).

When $C_{\overline{R}} \leq 0$, then $\overline{R}$ is an unsignificant value ($\overline{R}$ has no significant digit). From the concept of @.0, discrete stochastic relations (DSR) have been defined (equality and order relations).

Definition 3. Let X and Y be N-samples provided by CESTAC method, discrete stochastic equality denoted by $s =$ is defined as:

$X \, s = \, Y$ if $X - Y = @.0$.

Definition 4. Let X and Y be N-samples provided by CESTAC method, discrete stochastic inequalities denoted by $s >$ and $s \geq$ are defined as:

$X \, s > Y$ if $\overline{X} > \overline{Y}$ and $X - Y \neq @.0$.

$X \, s \geq Y$ if $\overline{X} \geq \overline{Y}$ or $X - Y = @.0$.

The Discrete Stochastic Arithmetic (DSA) is the association of the CESTAC method, the concept of computational zero and the discrete stochastic relations (see [3, 4, 16]). With this DSA it is possible to control the run of a scientific code, to detect the numerical instabilities and the violation of the hypotheses underlying the method. But in practice how to implement this?

As we observed, the two main specificities of the CESTAC method are:

- The random rounding, which consists in creating R^- and R^+ and in choosing randomly one or the other.

- The manner to perform the N runs of a code.

With IEEE arithmetic and the possibilities of ADA, C++, and Fortran to create new structures and to overload the operators it is easy to implement the CESTAC method.

The random rounding uses the IEEE rounding toward $+\infty$ and toward $-\infty$. These roundings occur whenever an arithmetic operation has a result that is not exact. Then no artificial round-off error is introduced in the computation. The choice of the rounding is at random with an equal probability for the $(N-1)$ first samples and the choice of the last one is the opposite of the choice of the $(N-1)$th sample. With this random rounding the theorems on exact rounding are respected.

We have seen previously that it is absolutely necessary to detect, during the run of a code, the emergence of @.0 for controlling the validity of the CESTAC method. To achieve this it suffices to use the synchronous implementation which consists in performing each arithmetic operation N times with the random rounding before performing the next. Thus for each numerical result we have N samples, from which with equation (9) the number of exact significant digits of the mean value, considered as the computed result, is estimated.

With this implementation the stochastic order relations defined above may also be easily created. Then during the run of a code a dynamic control may be done.

3.2. The CADNA library

The CADNA software [2, 6] is a library which implements automatically the DSA in any code written in Fortran. Using the CADNA library (Control of Accuracy and Debugging for Numerical Application), each standard FP types have their corresponding stochastic types. Every intrinsic function and operator are overloaded for those types. When a stochastic variable is printed, only its significant digits are displayed to point out its accuracy. If a number has no significant digit (i.e., a computed zero), the symbol @.0 is displayed.

The modifications that the user has to do in his Fortran source are mainly to change the declaration statements of real type by stochastic type, and the input-output statement (see [6]). Thus, when a modified Fortran source combined with the CADNA library is run, it is as $(N = 3)$ identical codes were simultaneously run on N synchronized computers each of them using the random rounding mode. So round-off error propagation can be analyzed step by step and then any numerical anomaly can be dynamically detected. This leads to the self validation of the method and a numerical debugging scientific codes.

We shall see in the numerical study how the use of the CADNA library has allowed us to obtain the optimal iterate of the power and inverse iteration methods.

4. The use of the CADNA library for the power and inverse iteration methods and numerical examples

According to the previous results, by using the CESTAC method and the CADNA library, we propose to use $|\widetilde{\lambda}_m - \widetilde{\lambda}_{m+1}| = @.0$ as the stopping criterion. The use of this stopping criterion allows us to stop the iterative process as soon as the difference between $\widetilde{\lambda}_m$ and $\widetilde{\lambda}_{m+1}$ is equal to the computational zero and the optimal iterate is reached. In this case, by

the theoretical results presented in section 2, the common significant digits between $\widetilde{\lambda}_m$ and $\widetilde{\lambda}_{m+1}$ are the common significant digits between $\widetilde{\lambda}_m$ and λ_1, up to $\log_{10} \frac{1}{1-\alpha}$. As mentioned in section 2, the number of exact significant digits of the optimal iterate $\widetilde{\lambda}_m$ can be estimated if a moderate approximate of $1-\alpha$ is available. It has been observed in experiments that by using the CADNA library and computing the values β_m, by means of formula (8), we can obtain a good approximation of $1-\alpha$

Let us now present the examples and the results which we obtained by the Fortran codes of the power and inverse iteration methods combined with the CADNA library [1]. All of the numerical experiments were computed in double precision. In all the examples the initial vector v_0 is $[1, 0, \ldots, 0]^T$. The first two examples were devoted to the power method and the rest of the examples were tested for the inverse iteration method. We used some symmetric matrices, since symmetric matrices have orthonormal eigenvectors (See [15], Theorem 6.4.2).

Example 1. Consider the matrix $A = (a_{ij})$ of dimension 10 with

$$a_{ij} = \begin{cases} i, & i = j, \\ 1, & i \neq j. \end{cases}$$

Numerical results of the power method combined with the CADNA library are given in Table 1. At the last line of this table we can find the exact values of λ_1, λ_2 (computed by a MATLAB code), and the value $(1-\alpha)$ which is furnished by these values. As we observe, the computed eigenvalue of largest modulus is $\lambda_* = \widetilde{\lambda}_{26} = 0.153100056907921E+02$, whereas the first digits of the exact one is $\lambda_1 = 0.1531000569079220E+02$. The last column shows that, as expected, the sequence β_m converges toward an approximation of $(1-\alpha)$, but the number of exact significant digits of β_m decreases and serious round-off errors exist at the end of sequence (due to subtraction of two nearly equal numbers). By noting this remark, it has been observed in experiments that we can obtain a good approximation of $(1-\alpha)$ by taking the value of β_m which has 3 or 2 significant digits. For this example, we observe that, the values β_{19}, β_{20}, and $\beta_{21} = 0.683$ which have 3 significant digits are good approximation of $(1-\alpha)$. Finally, from these results and $\log_{10} \frac{1}{\beta_{21}} = 0.166$, we conclude that the significant digits of $\widetilde{\lambda}_{26}$ are in common with λ_1, up to $1 + [0.166] = 1$. As we see, only the last digit of $\widetilde{\lambda}_{26}$ and λ_1 are different.

Example 2. This example is devoted to the Hilbert matrix $H = (h_{ij})$ of dimension 50 with $h_{ij} = \frac{1}{i+j-1}$. This matrix is SPD. Numerical results are given in Table 2. The computed eigenvalue of largest modulus is $\lambda_* = \widetilde{\lambda}_{15} = 0.20762966831311E+01$, whereas the first digits of the exact one (computed by a MATLAB code) is $\lambda_1 = 0.207629668313116E+001$. By taking $\beta_{11} = 0.893$, as an approximation to $(1-\alpha)$, and using $\log_{10} \frac{1}{\beta_{11}} = 0.0491$ we conclude that the significant digits of $\widetilde{\lambda}_{15}$ are in common with λ_1, up to $1 + [0.0491] = 1$. As we see, all of the significant digits of $\widetilde{\lambda}_{15}$ are in common with λ_1.

[1] The CADNA library, URL address: http://www.lip6.fr/cadna

Table 1: Results for Example 1.

m	$\widetilde{\lambda}_m$	$\lvert\widetilde{\lambda}_{m-1} - \widetilde{\lambda}_m\rvert$	β_m
1	0.152173913043478E+02	-	0.8856620546038E+0
2	0.152916690465264E+02	0.7427774217859E-001	0.8020108433500E+0
3	0.153056190401234E+02	0.139499935970E-001	0.760771349153E+0
4	0.153088497407296E+02	0.32307006062E-002	0.73648458702E+0
5	0.153096831581478E+02	0.83341741819E-003	0.72098046897E+0
6	0.153099123308364E+02	0.2291726885E-003	0.7105410649E+0
7	0.153099779861329E+02	0.6565529650E-004	0.7032490105E+0
8	0.153099973245900E+02	0.193384571E-004	0.698021835E+0
9	0.153100031324485E+02	0.580785846E-005	0.694204893E+0
10	0.153100049012371E+02	0.17687886E-005	0.69138040E+0
11	0.153100054454539E+02	0.5442167E-006	0.68927018E+0
12	0.153100056141688E+02	0.1687148E-006	0.6876827E+0
13	0.153100056667694E+02	0.526006E-007	0.6864825E+0
14	0.153100056832387E+02	0.164693E-007	0.685572E+0
15	0.153100056884119E+02	0.517318E-008	0.684879E+0
16	0.153100056900408E+02	0.162892E-008	0.68435E+0
17	0.153100056905547E+02	0.51386E-009	0.6839E+0
18	0.153100056907170E+02	0.1623E-009	0.68366E+0
19	0.153100056907684E+02	0.5133E-010	0.683E+0
20	0.153100056907846E+02	0.162E-010	0.683E+0
21	0.153100056907898E+02	0.51E-011	0.683E+0
22	0.153100056907914E+02	0.16E-011	0.68E+0
23	0.153100056907919E+02	0.5E-012	0.6E+0
24	0.153100056907921E+02	0.16E-012	0.7E+0
25	0.153100056907921E+02	0.5E-013	-
26	0.153100056907921E+02	@.0	-

λ_1=0.1531000569079220E+02, $\lambda_2 = 0.862476530957022E + 01$, $1 - \alpha = 0.68264607$

Table 2: Results for Example 2.

m	$\widetilde{\lambda}_m$	$\lvert\widetilde{\lambda}_{m-1} - \widetilde{\lambda}_m\rvert$	β_m
0	0.190619761499661E+01	-	0.8831742657040E+0
1	0.205642473459327E+01	0.15022711959666E+000	0.891606316528E+0
2	0.207414268943139E+01	0.1771795483811E-001	0.892695652533E+0
3	0.207606555024276E+01	0.192286081136E-002	0.8928207994E+0
4	0.207627191049296E+01	0.20636025020E-0030	0.892834610E+0
5	0.207629402836174E+01	0.221178687E-004	0.89283611E+0
6	0.20762963986357E+01	0.237027400E-005	0.8928362E+0
7	0.20762966526435E+01	0.2540078E-006	0.8928363E+0
8	0.207629667986400E+01	0.2722042E-007	0.892835E+0
9	0.207629668278103E+01	0.29170E-008	0.89284E+0
10	0.207629668309364E+01	0.3126E-009	0.8928E+0
11	0.207629668312714E+01	0.3350E-010	0.893E+0
12	0.20762966831307E+01	0.358E-011	0.89E+0
13	0.207629668313111E+01	0.38E-012	0.9E+0
14	0.207629668313116E+01	0.4E-013	-
15	0.207629668313116E+01	@.0	-
	λ_1=0.207629668313116E+01,	$\lambda_2 = 0.67969375295939E+0$,	$1-\alpha = 0.89283629$

Example 3. Let $A = (a_{ij})$ be a tridiagonal matrix of dimension 10 with

$$a(i,i) = 5, \quad a(i,i-1) = a(i+1,i) = -1.$$

The inverse iteration method combined with the CADNA library with shift $\sigma = 3$ was used to find the eigenvalue A closest to σ. The results are given in Table 3. At the last line of this table we can find the closest eigenvalue of A to the shift σ which is the smallest eigenvalue λ_{10} and the second closest λ_9 which are computed by a MATLAB code. This Table shows that the optimal iterate is reached at iteration 13 and $\lambda_* = \widetilde{\lambda}_{13} = 0.308101405277100E+001$, whereas the first digits of the exact one is $\lambda_{10} = 0.308101405277101E+01$. By taking $\beta_{10} = 0.93$, as an approximation to $(1-\alpha)$, and using $\log_{10}\frac{1}{\beta_{10}} = 0.0315$ we conclude that the significant digits of $\widetilde{\lambda}_{13}$ are in common with λ_{10}, up to $1 + [0.0292] = 1$. As we see, only the last digit of $\widetilde{\lambda}_{13}$ and λ_{10} are different.

Example 4. Let $A = (a_{ij})$ be a matrix of dimension 10 with

$$a_{ij} = \begin{cases} i, & i = j, \\ \frac{1}{n}, & i \neq j. \end{cases}$$

The results of the inverse iteration method combined with the CADNA library and shift $\sigma = 11$ are listed in Table 4. The iterative process converges to the closest eigenvalue to σ , $\lambda_J = \lambda_1 = 0.1003585959779057E+02$, in 23 iterations and $\lambda_* = \widetilde{\lambda}_{23} = 0.100358595977905E+$

Table 3: Results for Example 3.

m	$\widetilde{\lambda}_m$	$\lvert\widetilde{\lambda}_{m-1} - \widetilde{\lambda}_m\rvert$	β_m
1	0.328571428571428E+01	-	0.97618215811950E+0
2	0.308588957055214E+01	0.19982471516213E+000	0.948081618524E+0
3	0.308126718176305E+01	0.462238878909E-002	0.93752379656E+0
4	0.308102986730940E+01	0.23731445364E-003	0.9354228598E+0
5	0.308101507402866E+01	0.1479328074E-004	0.935000067E+0
6	0.308101411915268E+01	0.954875983E-006	0.9349125E+0
7	0.308101405709162E+01	0.6206106E-007	0.934894E+0
8	0.308101405305230E+01	0.403931E-008	0.93489E+0
9	0.308101405278931E+01	0.26298E-009	0.9348E+0
10	0.308101405277219E+01	0.1712E-010	0.93+0
11	0.308101405277108E+01	0.111E-011	0.9E+0
12	0.308101405277101E+01	0.72E-013	-
13	0.308101405277100E+01	@.0	-
	λ_{10}=0.308101405277100E+01,	$\lambda_9 = 0.331749293433764E+01$,	$1-\alpha = 0.93488926$

02. By taking $\beta_{19} = 0.763$, as an approximation to $(1-\alpha)$, and using $\log_{10}\frac{1}{\beta_{19}} = 0.1175$ we conclude that the significant digits of $\widetilde{\lambda}_{23}$ are in common with λ_{10}, up to $1+[0.1175] = 1$. As we see, all of the digits of $\widetilde{\lambda}_{23}$ are correct.

5. Conclusion

In this paper, we observed that the use of the CESTAC method and the CADNA library allows us to find the optimal iterate of the power and inverse iteration methods. It has been shown that, it is possible, on the one hand, by using the optimal termination criterion which uses the computational zero, to stop correctly the iterative process, to save computer time, because many useless iterations are not performed, and on the other hand, by using an approximation of $1-\alpha$, furnished by the power and inverse iteration methods, and the CADNA library, to estimate the accuracy of the optimal iterate. Some numerical results have been given to show the efficiency of the method.

Acknowledgments

The authors are grateful to the anonymous referee for his/her comments which substantially improved the quality of this paper.

Table 4: Results for Example 4.

m	$\widetilde{\lambda}_m$	$\lvert\widetilde{\lambda}_{m-1} - \widetilde{\lambda}_m\rvert$	β_m
1	0.127828531024632E+01	-	0.711972667598672E+0
2	0.751343883744274E+01	0.623515352719641E+001	98065482006121E+0
3	0.998706291430030E+01	0.24736240768575E+001	0.9273701561187E+0
4	0.100323155022867E+02	0.4525258798643E-001	0.79571966959E+0
5	0.100351356087900E+02	0.28201065033E-002	0.77000557970E+0
6	0.100356930843600E+02	0.55747557004E-003	0.7660198259E+0
7	0.100358206369491E+02	0.12755258902E-003	0.7646845546E+0
8	0.100358504297028E+02	0.2979275369E-004	0.764129683E+0
9	0.100358574353108E+02	0.700560799E-005	0.76389068E+0
10	0.100358590872089E+02	0.16518981E-005	0.7637867E+0
11	0.100358594771844E+02	0.3899754E-006	0.7637412E+0
12	0.100358595692963E+02	0.9211190E-007	0.763721E+0
13	0.100358595910579E+02	0.217616E-007	0.763712E+0
14	0.100358595961997E+02	0.51417E-008	0.76370E+0
15	0.100358595974146E+02	0.121492E-008	0.7637E+0
16	0.100358595977017E+02	0.2870E-009	0.7636E+0
17	0.100358595977695E+02	0.6783E-010	0.7636E+0
18	0.100358595977856E+02	0.160E-010	0.763E+0
19	0.100358595977893E+02	0.378E-011	0.76E+0
20	0.100358595977902E+02	0.89E-012	0.7
21	0.100358595977904E+02	0.2E-012	0.8
22	0.100358595977905E+02	0.5E-013	-
23	0.100358595977905E+02	@.0	-

$\lambda_1 = 0.1003585959779057E+02, \quad \lambda_2 = 0.901654830363739E+01, \quad 1-\alpha = 0.76371436$

References

[1] J. M. Chesneaux and F. Jézéquel, Dynamical Control of Computations Using the Trapezodial and Simpson's rules, J. Universal Comput. Sci. 4 (1998) 2-10.

[2] J. M. Chesneaux, L'arithmétique stochastique et le logiciel CADNA, Habilitation á diriger des recherches, Université Pierre et Marie Curie, Paris, 1995.

[3] J. M. Chesneaux, Stochastic arithmetic properties, IMACS Comput. Appl. Math. (1992) 81-91.

[4] J. M. Chesneaux and J. Vignes, Les fondements de l'arithmétique stochastique, C. R. Acad. Sci. Paris, Sér. I Math., 315 (1992) 1435-1440.

[5] J.M. Chesneaux, Study of the computing accuracy by using probabilistic approach, in: C. Ullrich (Ed.), Contribution to Computer Arithmetic and Self-Validating Numerical Methods, IMACS, New Brunswick, NJ, 1990.

[6] J. M. Chesneaux, CADNA: An ADA tool for round-off errors analysis and for numerical debugging, In ADA in Aerospace, Barcelone, Spain, December 1990.

[7] 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.

[8] J. W. Demmel, Applied numerical linear algebra, SIAM, Philadelphia, 1997.

[9] G. Golub and C. Van Loan, Matrix computations, John Hopkins University Press, Baltimore, MD, Third edition, 1996.

[10] F. Jézéquel and J. M. Chesneaux, CADNA: a library for estimating round-off error propagation, Computer Physics Communications 178 (2008) 933955.

[11] F. Jézéquel, Dynamical control of converging sequences computation, Appl. Numer. Math. 50 (2004) 147-164.

[12] F. Jézéquel, A dynamical strategy for approximation methods, C. R. Mecanique 334 (2006) 362-367.

[13] C. D. Meyer, Matrix analysis and applied linear algebra, SIAM, Philadelphia, 2000.

[14] Y. Saad, Numerical methods for large eigenvalue problems: theory and algorithms, Wiley, New York, 1992.

[15] J. Stoer and R. Bulirsch, Introduction to Numerical Analysis, Springer-Verlag, 1980.

[16] J. Vignes, A stochastic approach to the analysis of round-off error propagation. A survey of the CESTAC method, in: Proc. 2nd Real Numbers and Computers Conference, Marseille, France, 1996, pp. 233-251.

[17] J. Vignes, A stochastic arithmetic for reliable scientific computation, Math. Comp. Simul., 35(1993) 233-261.

[18] J. Vignes, Zéro mathématique et zéro informatique, C. R. Acad. Sci. Paris Sér. I Math., 303 (1986) 997-1000.

[19] J. Vignes, New methods for evaluating the validity of the results of mathematical computations, Math. Comp. Simul., 20(1978) 227-249.

[20] J. Vignes and M. La Porte, Error analysis in computing, in: Information Processing 1974, North-Holand, (1974) 610-614.

[21] J. Vignes, Discrete stochastic arithmetic for validating results of numerical software, Numerical Algorithms 37(2004) 377-390.

[22] D. S. Watkins, Fundamentals of matrix computations, John Wiley and Sons, New York, Second edition, 2002.