



HAL
open science

A simple and efficient algorithm for eigenvalues computation

René Alt

► **To cite this version:**

René Alt. A simple and efficient algorithm for eigenvalues computation. The 6th international conference on Numerical methods and applications, Aug 2007, Borovets, Bulgaria. pp.271-278, 10.1007/978-3-540-70942-8_32 . hal-01334828

HAL Id: hal-01334828

<https://hal.science/hal-01334828>

Submitted on 5 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

A Simple and Efficient Algorithm for Eigenvalues Computation

René Alt

University Pierre et Marie Curie – Paris 6, Laboratoire LIP6, CNRS UMR 7606
4 Place Jussieu, Paris 75230, France
`Rene.Alt@lip6.fr`

Abstract. A simple algorithm for the computation of eigenvalues of real or complex square matrices is proposed. This algorithm is based on an additive decomposition of the matrix. A sufficient condition for convergence is proved. It is also shown that this method has many properties of the QR algorithm : it is invariant for the Hessenberg form, shifts are possible in the case of a null element on the diagonal. Some other interesting experimental properties are shown. Numerical experiments are given showing that most of the time the behavior of this method is not much different from that of the QR method, but sometimes it gives better results, particularly in the case of a bad conditioned real matrix having real eigenvalues.

Keywords: Linear algebra, eigenvalues computation, CESTAC method.

1 Introduction

In this paper the additive reduction (AR) algorithm for computation of the eigenvalues of a real or complex matrix is recalled. The main properties of this algorithm are shown, and numerical experiments allow us to compare it with the classical QR algorithm. The paper is organized as follows. First, the AR algorithm is described. Then it is shown that it is invariant for the Hessenberg form, and a sufficient condition for convergence in terms of this form is proved. Finally, some numerical experiments are given, showing that the method is very efficient and often even more than the QR method, particularly in the case of multiple eigenvalues or for computation of roots of polynomials using the companion matrix.

2 The Additive Reduction (AR) Algorithm

2.1 Definition of the Algorithm

Let us consider a real or complex square matrix A with dimension n . One wishes to compute its eigenvalues. The algorithm, formerly proposed in [1] and called additive reduction algorithm (AR) is the following.

1. Split the matrix A into a sum of two triangular matrices L and U such that L is the lower triangular part of A including the diagonal and U is the strict upper triangular part of A . So $A = L + U$.
2. Set $A_0 = A = L_0 + U_0$
3. Suppose that L_{k-1} is invertible for $k > 0$, and compute

$$A_k = L_{k-1}^{-1} A_{k-1} L_{k-1} = L_{k-1} + L_{k-1}^{-1} U_{k-1} L_{k-1}. \quad (1)$$

4. Split $A_k = L_k + U_k$.
5. Perform iterations until reaching the tolerance prescribed or the maximum number of iterations specified.

It is clear that the eigenvalues are invariant in all the computed matrices. Thus, if the method converges to a lower triangular matrix, then the eigenvalues are found on the main diagonal of the limit matrix.

2.2 Invariance of the Algorithm for the Hessenberg Form

Let us recall that a matrix $A = (a_{i,j})$ has a lower Hessenberg form if and only if $a_{i,j} = 0$ for $j \geq i + 2$. Any matrix is similar to a Hessenberg matrix which can be obtained in finite time by the algorithms of Givens or Householder. Both algorithms are close to each other and require $O(n^3)$ operations. Classically, the LR and QR algorithm, being invariant for the Hessenberg form, start with a transformation of the initial matrix into this form. Let us see now that this is also possible for the AR algorithm.

Proposition 1. *If at iteration k the matrix A_k has a Hessenberg form, then this property also prevails at iteration $k + 1$ of the AR algorithm.*

Proof. The proof is a simple straightforward calculation. Suppose that a matrix A has a lower Hessenberg form, that is $a_{ij} = 0$ for $j \geq i + 2$, then from the definition of the AR algorithm, the next iterate is

$$B = L^{-1} A L = L + L^{-1} U L. \quad (2)$$

Set $V = UL$, then $v_{i,j} = \sum_{k=\max\{i+1,j\}}^n a_{i,k} a_{k,j}$. But, since $a_{i,k} = 0$ for $k \geq i + 2$,

we have $v_{i,j} = a_{i,i+1} a_{i+1,j}$, if $j \leq i + 1$, and $v_{i,j} = 0$ for $j \geq i + 2$. Hence $V = UL$ has a Hessenberg form. It is easy to check that the product of a lower diagonal matrix L^{-1} with a lower Hessenberg matrix V gives a lower Hessenberg matrix. Thus $Z = L^{-1}UL$ is a lower Hessenberg matrix, and so is $B = L + Z$. \square

3 Convergence of the Additive Reduction Method

Finding a necessary and sufficient condition for the convergence of the AR algorithm is still an open problem. However, a sufficient condition for convergence in the case of a lower Hessenberg matrix can be easily shown.

Proposition 2. Let $A = (a_{i,j})$ be a lower complex Hessenberg matrix. Assume that the following properties are fulfilled:

- 1) All the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$ of A are distinct in moduli.
- 2) At each iteration k the matrix A_k is such that

$$|a_{1,1}^{(k)}| > |a_{2,2}^{(k)}| > \dots > |a_{n,n}^{(k)}|,$$

i.e., the moduli of its diagonal elements form a strictly decreasing sequence.

- 3) The maximum ratio $r = \max_{i=1}^{n-1} |a_{i+1,i+1}^{(k)}|/|a_{i,i}^{(k)}|$ is bounded by a constant $C < 1$, independent of k .

Then the additive reduction method is convergent.

Here the matrix A is supposed to be complex, the operations involve complex numbers and the notation $|x|$ denotes the modulus of the complex number x .

Proof. Proposition 2 follows from a theorem due to Bauer and Fike [4]. As a consequence of this theorem, if A is diagonalisable, i.e., $A = P^{-1}AP$ with $A = \text{diag}(\{\lambda_i\}_{i=1}^n)$, then for each eigenvalue μ of $L = A - U$ there exists an eigenvalue λ of A such that:

$$|\lambda - \mu| < \|P\| \cdot \|P^{-1}\| \cdot \|U\|.$$

Thus, it suffices to prove that under the three precedent hypotheses the norm $\|U_k\|$ of the matrix U_k tends to zero. This will prove that the eigenvalues of L tend to the eigenvalues of A , as all they are of different moduli.

Using notation and the formulae from Proposition 1, it is clear that U_k is the strictly triangular superior part of the matrix Z . Thus, for $i = 1, \dots, n-1$,

$$\begin{aligned} u_{i,i+1}^{(k)} &= u_{i,i+1}^{(k-1)} a_{i+1,i+1}^{(k-1)} / a_{i,i}^{(k-1)} \\ u_{i,j}^{(k)} &= 0 \quad \text{for } j \geq i+2. \end{aligned}$$

Here, the superscript (k) denotes the number of the iteration. From assumption 3) we obtain

$$|u_{i,i+1}^{(k)}| \leq C |u_{i,i+1}^{(k-1)}| \quad \text{with } C < 1,$$

and thus

$$|u_{i,i+1}^{(k)}| \leq C^k |u_{i,i+1}^{(0)}| \quad \text{with } C < 1.$$

Let us now choose the classical norm $\|U\| = \max_{1 \leq i \leq n} \sum_{j=1}^n |u_{i,j}|$. In the present case $\|A_k\|$ is a lower Hessenberg matrix, U_k has only its first over diagonal different from zero, that is, $u_{i,j}^{(k)} = 0$ for $i < j-1$ and for $i \geq j$. The same property is also true for U_{k+1} . We have

$$\|U_k\| = \max_{1 \leq i \leq n-1} |u_{i,i+1}^{(k)}| \leq C \max_{1 \leq i \leq n-1} |u_{i,i+1}^{(k-1)}| = \|U_{k-1}\|,$$

and consequently

$$\|U_k\| \leq C^k \|U_0\| \quad \text{with } C < 1. \tag{3}$$

This proves Proposition 2. \square

Remark 1. If A is a real matrix with different real eigenvalues, then all the computations are done with real numbers and formula (3) shows that convergence still occurs as the main over diagonal of the lower Hessenberg iterates tends to zero. Thus this method cannot, in theory, lead to erroneous complex conjugate eigenvalues.

Remark 2. In the case of a matrix A with eigenvalues having different moduli, all the numerical experiments that have been done have shown that in most cases, even if at the beginning the elements of the main diagonal of A are not ordered in decreasing order of their moduli, this property becomes true at some iteration and stays true during all remaining iterations.

Remark 3. As was already mentioned, a necessary and sufficient condition for convergence is not known, although the numerical experiments have shown that the method may converge even in the case of multiple eigenvalues.

Remark 4. From the definition of the algorithm, it is clear that its complexity per iteration is $O(n^3)$, as each iteration requires solution of a triangular linear system. Thus, the complexity of AR is of the same order as the one of QR.

4 Case of a Null Element on the Main Diagonal

The detection of the fact that at some iteration L_k is not invertible is here very easy, as L_k is lower triangular.

In this case, in the same way that it is done in the classical QR or LU methods, the algorithm can be continued by adding an arbitrary constant to all the elements of the diagonal of A_k . This shift is then subtracted from the computed values at the end of the process.

5 Deflations of Columns and Termination of Iterations

As was said above, the eigenvalues are obtained on the main diagonal of a lower triangular matrix, which is the limit of the process. We already mentioned that, in most cases, on this diagonal the eigenvalues appear experimentally to be in decreasing order, i.e., the smallest one is in the last line and last column. Thus the iterations can be stopped using at least two criteria.

- Criterion 1: The elements of the diagonal of the matrix A_k become stable to some extent, i.e.,

$$\max_{1 \leq n} |a_{i,i}^{(k)} - a_{i,i}^{(k-1)}| \leq \varepsilon. \quad (4)$$

- Criterion 2: The elements of the upper triangular part U_k of A_k are stochastic zeroes, i.e. only correspond to round-off errors.

Let us recall here that a stochastic zero is a real or a floating point number which contains an error greater than itself. In other words, it has no significant

digit. In the theory of stochastic arithmetic ([8], [9], [2], [3], [7]), any stochastic number is represented as (m, σ) , where m is the mean value and σ is the standard deviation of a Gaussian distribution, i.e., m represents the value and σ is the error on m . A stochastic zero is such that $\sigma \geq m$. The corresponding notion in the theory of interval arithmetic is an interval containing zero. Remember also that the number of significant digits of the result of any floating point computation and hence detection of a stochastic zero can be obtained using the CESTAC method ([8], [9]).

The experiments show another feature of the method: in the case of a complex matrix A and computations using complex numbers or of a real matrix having real eigenvalues, during the iterations the matrices U_k and hence the matrices A_k have their columns tending to zero column after column, starting from the last. This means that column n tends to zero, then column $n - 1$ and so on to column 1.

In the case of a given real matrix A and computations done using only real numbers, there may exist complex conjugate roots. But it also happens that the columns corresponding to real roots tend to zero in the same order from the last one as in the complex case and that the columns corresponding to two complex conjugate roots, say λ_j and λ_{j+1} tend to zero except the four values $a_{j,j}$, $a_{j,j+1}$, $a_{j+1,j}$, $a_{j+1,j+1}$. The characteristic polynomial of this 2×2 block sub-matrix provides the two complex eigenvalues. This property is exactly the same as in QR or LR algorithms.

This experimental property can be exploited to increase the speed of convergence of the algorithm in the following manner: once the last column (resp, the last two columns) of the matrix U_k is (are) considered sufficiently small in norm, or that each of its (resp. their) components is equal to a stochastic zero, then the corresponding eigenvalue is obtained and it is possible to reduce the size of the current matrix A_k by suppressing the last (resp. last two) line(s) and column(s). This process can be re-done each time when a real or two complex conjugate eigenvalues are obtained until the dimension of the final matrix becomes 1 or 2. The speed of convergence of the algorithm is thus highly increased. Moreover, this property also experimentally diminishes the propagation of round-off errors and increases the accuracy of the results.

6 Numerical Experiments

Several numerical experiments have been done for small size matrices ($n \leq 100$). Many of them are taken from ([6], [5], [10]). Three of them are reported here to show the efficiency of the AR method. The results are compared with those provided by the classical QR algorithm [11]. The QR program is taken from [12], it provides the maximum possible accuracy. Since the original program does not give the number of necessary iterations to reach the solution, a counter has been added in order to compare the speed of convergence of the two methods. Only a very naive programming has been done for the AR method, in particular, there is

no preconditioning. The termination criterion is on the stability of the diagonal of the iterates, i.e., criterion 1 with formula (4).

Example 1: The Fibonacci matrix

$$A_1 = \begin{pmatrix} 3 & 5 & 8 & 13 & 21 \\ 5 & 8 & 13 & 21 & 34 \\ 8 & 13 & 21 & 34 & 55 \\ 13 & 21 & 34 & 55 & 89 \\ 21 & 34 & 55 & 89 & 144 \end{pmatrix}$$

The obtained results are given in Table 1. These results show that the two algorithms are equivalent concerning the number of iterations, but the AR algorithm produces only real eigenvalues, which is closer to the real situation, as the initial matrix is symmetric.

Table 1. Eigenvalues of the Fibonacci matrix

index	AR $\varepsilon = 10^{-5}$ Nb.iter = 5	QR Nb.iter = 4
1	0.2311038 10^3	0.2311038 10^3
2	-0.1038494 10^0	-0.1038494 10^0
3	-0.2687906 10^{-14}	-0.3158887 10^{-14}
4	-0.5342609 10^{-16}	-0.1528318 $10^{-28} + i$ 0.1209090 10^{-21}
5	-0.16677753 10^{-28}	-0.1528318 $10^{-28} - i$ 0.1209090 10^{-21}

Example 2. ([10])

$$A_2 = \begin{pmatrix} 10 & 1 & 4 & 0 \\ 1 & 10 & 5 & -1 \\ 4 & 5 & 10 & 7 \\ 0 & -1 & 7 & 9 \end{pmatrix}$$

The computed eigenvalues are shown in Table 2.

Table 2. Eigenvalues of matrix A_2 computed with AR and QR methods

index	AR $\varepsilon = 10^{-5}$, Nb. iter = 56	QR Nb. iter = 3
1	0.1912248 10^2	0.1912248 10^2
2	0.1088245 10^2	0.5342609 10^{-3}
3	0.8994541 10^1	0.1088282 10^2
4	0.5342609 10^{-3}	0.8994170 10^1

One can see that the results obtained with the two methods AR and QR are very close to each other, except that the first one provides the eigenvalues in decreasing order. For the AR algorithm with termination criterion 1 and $\varepsilon = 10^{-5}$, the convergence speed is noticeably slower compared to that of QR.

Example 3. A matrix with double eigenvalues (Rutishauser [10])

$$A_3 = \begin{pmatrix} 6 & 4 & 4 & 1 \\ 1 & 6 & 4 & 4 \\ 4 & 1 & 6 & 4 \\ 1 & 4 & 4 & 4 \end{pmatrix}$$

The exact eigenvalues are 15, 5, 2, 2. Both methods give the exact values. Again, the eigenvalues provided by the QR algorithm are obtained in the order 5, 2, 15, 2, whereas with the AR algorithm they are obtained in decreasing order. For the AR algorithm the iterations are stopped with criterion 1 and $\varepsilon = 10^{-7}$. The number of iterations is 20. For QR the number of iterations is 4, so QR is faster again.

Example 4. A real matrix with real opposite eigenvalues.

$$A_4 = \begin{pmatrix} 1.5 & 1 & -2 & 1 \\ 1 & 0.5 & -3 & -2 \\ -2 & -3 & -0.5 & -1 \\ 1 & -2 & -1 & -1.5 \end{pmatrix}$$

The eigenvalues of A_4 are -1.5 , $+1.5$, -4.5 , $+4.5$. They are exactly computed by both AR and QR algorithms, with 19 iterations for AR and 4 iterations for QR. As in the preceding examples, AR performs slightly slower than QR. However, it must be reminded that the program for AR is a very simple and naive one, particularly concerning the termination criterion, while the program for QR is rather sophisticated.

7 Conclusion

In this paper we have shown that the eigenvalues of a real or complex square matrix can be computed with a method called additive reduction method, which is different from the classical methods. A sufficient condition for convergence has been proved, and it has been also proved that it has the same properties as the QR or LU algorithms. In particular, it is invariant for the Hessenberg form and has the same order of complexity per iteration as QR. Some other experimental properties is also given: the eigenvalues are obtained in decreasing order on the main diagonal of a lower triangular matrix, and consequently the algorithm provides them the last one (i.e. the smallest) first. Many numerical experiments have shown that the method is often as good as the classical general methods for eigenvalues, and in some cases even better, particularly for a real matrix with real eigenvalues. Some of these experiments are reported here. It is clear that, in many cases, this method is not really competitive with QR, but it shows that the eigenvalue problem can be also solved by a different approach from the classical one. Many questions are still open, for example, find a necessary and sufficient condition for convergence, explain why in most cases the eigenvalues are obtained the smallest first.

References

1. Alt, R.: Un algorithme simple et efficace de calcul de valeurs propres. *C. R. Acad. Sc. Paris* **306(1)** 1988 437–440
2. Alt, R., Markov, S.: On the Algebraic Properties of Stochastic Arithmetic. Comparison to Interval Arithmetic. In: W. Kraemer and J. Wolff von Gudenberg, (eds.): *Scientific Computing, Validated Numerics, Interval Methods*. Kluwer (2001) 331–341
3. Alt R., Lamotte J.-L., Markov, S.: Numerical Study of Algebraic Solutions to Linear Problems Involving Stochastic Parameters. *Large Scale Scientific Computing conf. Lecture Notes in Computer Science Vol. 3743* (2006) 273–280
4. Bauer, F.L., Fike, C. T.: Norms and exclusion theorems. *Numer. Math.* **2** (1960) 137–141
5. Gregory, R., Kerney, D.: *A collection of matrices for testing computational algorithms*. Wiley, New-York (1969)
6. Higham, N.: Algorithm 694: A Collection of Test Matrices in MATLAB. *ACM Transactions on Mathematical Software Vol. 17, no. 3* (1991) 289–305
7. Markov, S., Alt, R.: Stochastic arithmetic: Addition and multiplication by scalars. *Applied Num. Math.* **50** (2004) 475–488
8. Vignes, J., Alt, R.: An Efficient Stochastic Method for Round-Off Error Analysis. In: *Accurate Scientific Computations. Lecture Notes in Computer Science Vol. 235*. Springer (1985) 183–205
9. Vignes, J.: A Stochastic Arithmetic for Reliable Scientific Computation. *Math. Comp. in Sim.* **35** (1993) 233–261
10. Westlake, J. R.: *A handbook of Numerical Matrix inversion and solution of linear equations*. J. Wiley (1968) 136–157
11. Wilkinson, J.: Convergence of the LR, QR and related algorithms. *Computer J.* **4** (1965) 77–84
12. Wilkinson, J. H., Reinsch, C. (eds.): *Linear Algebra. Handbook for Automatic Computation Vol. II*. Springer Verlag, New York (1971)