



**HAL**  
open science

## Counting eigenvalues in domains of the complex field

Emmanuel R. Kamgnia, Bernard Philippe

► **To cite this version:**

Emmanuel R. Kamgnia, Bernard Philippe. Counting eigenvalues in domains of the complex field. INRIA. 2011. hal-00634065v3

**HAL Id: hal-00634065**

**<https://hal.science/hal-00634065v3>**

Submitted on 2 Mar 2012

**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.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

*Counting eigenvalues in domains  
of the complex field*

Emmanuel Kamgnia — Bernard Philippe

N° 7770

October 2011

Observation and Modeling for Environmental Sciences

*R* *appo*  
*t*  
*de* *eche*  
*che*



## Counting eigenvalues in domains of the complex field

Emmanuel Kamgnia <sup>\*†</sup>, Bernard Philippe <sup>‡†</sup>

Theme : Observation and Modeling for Environmental Sciences  
Computational Sciences for Biology, Medicine and the Environment  
Équipe-Projet Sage

Rapport de recherche n° 7770 — October 2011 — 18 pages

**Abstract:** A procedure for counting the number of eigenvalues of a matrix in a region surrounded by a closed curve is presented. It is based on the application of the residual theorem. The quadrature is performed by evaluating the principal argument of the logarithm of a function. A strategy is proposed for selecting a path length that insures that the same branch of the logarithm is followed during the integration. Numerical tests are reported for matrices obtained from conventional matrix test sets.

**Key-words:** Eigenvalues, resolvent, determinant, complex logarithm

\* University of Yaounde I, Cameroun.

† This work was pursued within the team MOMAPPLI of the LIRIMA (Laboratoire International de Recherche en Informatique et Mathématiques Appliquées).

‡ Centre INRIA de Rennes Bretagne Atlantique

## Dénombrément de valeurs propres dans le plan complexe

**Résumé :** Nous proposons une approche pour compter le nombre de valeurs propres d'une matrice, situées dans un domaine borné du plan complexe. L'approche est fondée sur l'application du théorème des résidus. L'intégration nous ramène à l'évaluation de l'argument principal du logarithme d'une fonction. Nous proposons une stratégie pour le choix du pas qui permette de rester sur la même branche du logarithme pendant l'intégration. Des résultats numériques sont obtenus à partir de tests conduits sur des matrices tirées d'ensembles classiques de matrices.

**Mots-clés :** Valeurs propres, résolvante, déterminant, logarithme complexe

## 1 Introduction

The localization of eigenvalues of a given matrix  $A$  in a domain of the complex plane is of interest in scientific applications. When the matrix is real symmetric or complex hermitian, a procedure based on computations of Sturm sequences allows to safely apply bisections on real intervals to localize the eigenvalues. The problem is much harder for non symmetric or non hermitian matrices and especially for non normal ones. This last case is the main concern of this work. Proceeding by trying to compute the eigenvalues of the matrix may not always be appropriate for two reasons.

First most of the iterative methods frequently used to calculate eigenvalues of large and sparse matrices may lose some of them, since only a part of the spectrum is computed, and as such there is no guarantee to localize all the eigenvalues of the selected domain. When a shift-and-invert transformation is used, the eigenvalues are obtained in an order more or less dictated by their distance from the shift, and if one eigenvalue is skipped, there is no easy strategy that allows to recover it.

Second the entries of the matrix may be given with some errors and then the eigenvalues can only be localized in domains of  $\mathbb{C}$ .

Many authors have defined regions in the complex plane that include the eigenvalues of a given matrix. One of the main tool is the Gershgorin theorem. Since a straight application of the theorem often leads to large disks, some authors extended the family of inequalities for obtaining smaller regions by intersections which include eigenvalues (see e.g., [7, 11]). Other techniques consist to consider bounds involving the singular values (see e.g., [4]), the eigenvalues of the hermitian part and the skew-hermitian part of the matrix (see e.g., [2]), or the field of values of inverses of the shifted matrices (see e.g., [10]).

For taking into account, possible perturbations of the matrix, Godunov[8] and Trefethen [14] have separately defined the notion of the  $\epsilon$ -spectrum or pseudospectrum of a matrix to address the problem. The problem can then be reformulated as that of determining level curves of the 2-norm of the resolvent  $R(z) = (zI - A)^{-1}$  of the matrix  $A$ .

The previous approaches determine a priori enclosures of the eigenvalues. A dual approach can be considered: given some curve ( $\Gamma$ ) in the complex plane, count the number of eigenvalues of the matrix  $A$  that are surrounded by ( $\Gamma$ ). This problem was considered in [5] where several procedures were proposed. In this paper, we make some progress with respect to the work in [5]. Our procedure is based on the application of the residual theorem: the integration process leads to the evaluation of the principal argument of the logarithm of the function  $g(z) = \det((z+h)I - A)/\det(zI - A)$ . This function is also considered in [6] to count the eigenvalues when a nonlinear eigenvalue problem is perturbed.

This work is mainly concerned with the control of the integration path so as to stay on the same branch along an interval when evaluating the principal argument of a logarithm.

In section 2, we present the mathematical tools. In section 3, we present the basis of our strategy for following a branch of the logarithm function and conditions for controlling the path length. Section 4 deals with the implementation of our method: we show how to safely compute the determinant and how to include new points along the boundary. In section 5 we present numerical

test results carried out on some test matrices and in section 6, we conclude with some few remarks and future works.

## 2 Mathematical tool and previous works

In this section we present the Cauchy's argument principle and some previous works on counting eigenvalues in regions of the complex field.

### 2.1 Use of the argument principle

The localization of the eigenvalues of matrix  $A$  involves the calculation of determinants. Indeed let  $(\Gamma)$  be a closed piecewise regular Jordan curve (piecewise  $C^1$  and of winding number 1) in the complex plane which does not include eigenvalues of  $A$ . The number  $N_\Gamma$  of eigenvalues surrounded by  $(\Gamma)$  can be expressed by the Cauchy formula (see e.g., [12, 13]):

$$N_\Gamma = \frac{1}{2i\pi} \int_\Gamma \frac{f'(z)}{f(z)} dz, \quad (1)$$

where  $f(z) = \det(zI - A)$  is the characteristic polynomial of  $A$ .

If  $\gamma(t)_{0 \leq t \leq 1}$  is a parametrization of  $\Gamma$  the equation (1) can be rewritten as

$$N_\Gamma = \frac{1}{2i\pi} \int_0^1 \frac{f'(\gamma(t))}{f(\gamma(t))} \gamma'(t) dt. \quad (2)$$

The primitive  $\varphi$  defined by

$$\varphi(u) = \int_0^u \frac{f'(\gamma(t))}{f(\gamma(t))} \gamma'(t) dt, \quad u \in [0, 1],$$

is a continuous function which is a determination of  $\log(f \circ \gamma)$  (e.g. see [13]):

$$\log f(\gamma(t)) = \log |f(\gamma(t))| + i \arg(f(\gamma(t))), \quad t \in [0, 1].$$

It then follows that

$$N_\Gamma = \frac{1}{2\pi} \varphi_I(1),$$

where  $\varphi_I(1)$  is the imaginary part of  $\varphi(1)$  since its real part vanishes.

### 2.2 Counting the eigenvalues in a region surrounded by a closed curve

In [5], two procedures were proposed for counting the eigenvalues in a domain surrounded by a closed curve.

The first method is based on the series expansion of  $\log(I + hR(z))$ , where  $R(z) = (zI - A)^{-1}$ , combined with a path following technique. The method uses a predictor - corrector scheme with constant step size satisfying the constraint

$$|\varphi_I(z + \Delta z) - \varphi_I(z)| < \pi,$$

for a discrete list of points  $z$ . The implementation of the algorithm requires the computation of a few of the smallest singular values and the corresponding left and right singular vectors of  $(zI - A)$ ; they are used to follow the tangent to the level curve of the smallest singular value of  $(zI - A)$ .

In the second procedure, the domain is surrounded by a parameterized user-defined curve  $z = \gamma(t)$  and thus

$$N_\Gamma = \frac{1}{2i\pi} \int_{\gamma(0)}^{\gamma(1)} \frac{\frac{d}{dt} \det(\gamma(t)I - A)}{\det(\gamma(t)I - A)} dt \quad (3)$$

Since  $\gamma(0) = \gamma(1)$ , the function  $\gamma(t)$  defined on  $[0, 1]$ , can be extended onto  $\mathbb{R}$  by

$$\gamma_{ext}(t) = \gamma(t \bmod 1).$$

By subdividing the interval  $[\gamma(0), \gamma(1)]$  into subintervals of equal length, and by assuming that  $\gamma_{ext} \in \mathcal{C}^\infty$ , they make use of a fundamental result from quadrature of periodic function to prove an exponential convergence of the integral. The method is compared to other integrators with adaptive step sizes.

Each of these methods makes use of the computation of

$$u(t) = \frac{\det(\gamma(t)I - A)}{|\det(\gamma(t)I - A)|},$$

which is efficiently computed through a LU factorization of the matrix  $(\gamma(t)I - A)$  with partial pivoting. In order to avoid underflow or overflow, the quantity is computed by

$$\frac{\det(\gamma(t)I - A)}{|\det(\gamma(t)I - A)|} = \prod_{i=1}^n \frac{u_{ii}}{|u_{ii}|}$$

where  $u_{ii}$  is the  $i$ -th diagonal element of  $U$  in the LU factorization. The product is computed using the procedure that will be described later on in section 4.

Our work, which can be viewed as an improvement of [5], mostly deals with the control of the integration so as to stay on the same branch along an interval, during the evaluation of the principal argument of the logarithm of the function  $g(z)$  defined in the introduction.

### 3 Integrating along a curve

In this section, we describe strategies for the integration of the function  $g(z) = \frac{f'(z)}{f(z)}$ , where  $f(z) = \det(zI - A)$ , along the boundary of a domain limited by a user-defined curve ( $\Gamma$ ) that does not include eigenvalues of  $A$ .

#### 3.1 Following a branch of $\log(f(z))$ along the curve

To simplify the presentation and without loss of generalization, let us assume that  $\Gamma = \bigcup_{i=0}^{N-1} [z_i, z_{i+1}]$  is a polygonal curve.

Let  $\text{Arg}(z) \in (-\pi, \pi]$  denote the principal determination of the argument of a complex number  $z$ , and  $\arg(z) \equiv \text{Arg}(z) + 2\pi k$ , be any determination of the argument of  $z$ . In this section, we are concerned with the problem of following a branch of  $\log(f(z))$  when  $z$  runs along ( $\Gamma$ ). The branch (i.e. a determination



$\arg_0$  of the argument), which is to be followed along the integrating process, is fixed by selecting an origin  $z_0 \in (\Gamma)$  and by insuring

$$\arg_0(f(z_0)) = \text{Arg}(f(z_0)). \quad (4)$$

Let  $z$  and  $z + h$  two points of  $(\Gamma)$ . Since

$$\begin{aligned} (z + h)I - A &= (zI - A) + hI \\ &= (zI - A)(I + hR(z)), \end{aligned}$$

where  $R(z) = (zI - A)^{-1}$ , it then follows that

$$f(z + h) = f(z) \det(I + hR(z)). \quad (5)$$

Let  $\Phi_z(h) = \det(I + hR(z))$ , then

$$\begin{aligned} \int_z^{z+h} \frac{f'(z)}{f(z)} dz &= \log(f(z + h)) - \log(f(z)) \\ &= \log\left(\frac{f(z + h)}{f(z)}\right) \\ &= \log(\Phi_z(h)) \\ &= \log|\Phi_z(h)| + i \arg(\Phi_z(h)). \end{aligned}$$

In the previous approach [5], given  $z$ , the step  $h$  is chosen such that condition

$$|\arg(\Phi_z(h))| < \pi, \quad (6)$$

is satisfied. In [5] condition (6) is only checked at point  $z + h$  but we want the condition to be satisfied at all the points  $s \in [z, z + h]$ , so as to guarantee that we stay on the same branch along the interval  $[z, z + h]$ . We need a more restrictive condition which is mathematically expressed by the following lemma:

**Lemma 3.1 (Condition (A))** *Let  $z$  and  $h$  be such that  $[z, z + h] \subset (\Gamma)$ .*

*If*

$$|\text{Arg}(\Phi_z(s))| < \pi, \quad \forall s \in [0, h], \quad (7)$$

*then,*

$$\arg_0(f(z + h)) = \arg_0(f(z)) + \text{Arg}(\Phi_z(h)), \quad (8)$$

*where  $\arg_0$  is the determination of the argument determined as in (4) by an a priori given  $z_0 \in (\Gamma)$ .*

**Proof.** We prove it by contradiction. Let us assume that there exists  $k \in \mathbb{Z} \setminus \{0\}$  such that

$$\arg_0(f(z + h)) = \arg_0(f(z)) + \text{Arg}(\Phi_z(h)) + 2k\pi.$$

By continuity of the branch, there exists  $s \in [0, h]$  such that  $|\text{Arg}(\Phi_z(s))| = \pi$ , which contradicts (7).  $\diamond$

Condition (7) is called **Condition (A)**.

### 3.2 Step size control

In our approach, given  $z$ , the step  $h$  is chosen such that condition of Lemma (3.1) is satisfied. **Condition (A)** is equivalent to

$$\Phi_z(s) \notin (-\infty, 0], \forall s \in [0, h].$$

In order to find a practical criterion to insure it, we look for a more severe condition by requiring that  $\Phi_z(s) \in \Omega$ , where  $\Omega$  is an open convex set, neighborhood of 1, and included in  $\Omega \subset \mathbb{C} \setminus (-\infty, 0]$ . Possible options for  $\Omega$  are the positive real half-plane, or any disk included in it and centered in 1.

Since  $\Phi_z(0) = 1$ , let

$$\Phi_z(s) = 1 + \delta, \text{ with } \delta = \rho e^{i\theta}.$$

A sufficient condition for (7) to be satisfied is  $\rho < 1$ , i.e.

$$|\Phi_z(s) - 1| < 1, \quad \forall s \in [0, h] \quad (9)$$

This condition will be referred to as **Condition (B)**, and, when only verified at  $z + h$ , i.e.

$$|\Phi_z(h) - 1| < 1, \quad (10)$$

it will be referred to as **Condition (B')**. This last condition is the condition used in [5]. It is clear that **Condition (B)** implies **Condition (A)** whereas this is not the case for **Condition (B')**.

Since it is very difficult to check (9), we apply the condition on the linear approximation  $\Psi_z(s) = 1 + s\Phi'_z(0)$  of  $\Phi_z(s)$  at 0. Replacing function  $\Phi_z$  by its tangent  $\Psi_z$  in (9), leads to

$$|\Psi_z(s) - 1| < 1, \quad \forall s \in [0, h], \quad (11)$$

which is equivalent to the following condition, referred as **Condition (C)**:

$$|h| < \frac{1}{|\Phi'_z(0)|}. \quad (12)$$

**Example 3.1 (First illustration)** Let  $A = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$ . It then follows that

$$\begin{aligned} f(z) &= z(z-1), \\ \Phi_z(h) &= \left(1 + \frac{h}{z}\right)\left(1 + \frac{h}{z-1}\right), \\ \Phi'_z(0) &= \frac{1}{z} + \frac{1}{z-1}. \end{aligned}$$

Let us assume that we are willing to integrate along the segment from  $z = 2$  to  $z = 1 + i$ . In order to see if intermediate points are needed to insure that the branch of the logarithm is correctly followed, we consider the previously introduced conditions on  $h = t(-1 + i)$  where  $t \in [0, 1]$ .

**Condition (A):**  $\Phi_2(h) = 1 + \frac{3h}{2} + \frac{h^2}{2}$  is a non positive real number if and only if  $h \in [-2, -1] \cup (-\frac{3}{2} + i\mathbb{R})$ . From that, it can easily be seen that the segment  $[0, -1 + i]$  does not intersect the forbidden region. Therefore no intermediate points are needed.

**Condition (B):** this condition is equivalent to  $|h||3+h| < 2$ . By studying the function  $\phi(t) = |h||3+h| = \sqrt{2t}|3-t+it|$ , the parameter  $t$  must remain smaller than  $\alpha \approx 0.566$ .

**Condition (B'):** in this example, this condition is equivalent to the previous one, since the function  $\phi(t)$  is increasing with  $t$ .

**Condition (C):** since  $\Phi'_2(h) = \frac{3}{2} + h$ , this condition limits the extent of the interval to  $|h| < \frac{2}{3}$  or equivalently  $t < \frac{\sqrt{2}}{3} \approx 0.471$ .

In the second example, we illustrate the lack of reliability of **Condition (B')**.

**Example 3.2 (Second illustration)** Let  $A = \lambda I_n$ , where  $\lambda \in \mathbb{R}$  and  $I_n$  is the identity matrix of order  $n$ . It then follows that

$$\begin{aligned} f(z) &= (z - \lambda)^n, \\ \Phi_z(h) &= \left(1 + \frac{h}{z - \lambda}\right)^n, \\ \Phi'_z(0) &= \frac{n}{z - \lambda}. \end{aligned}$$

Let us assume that we are willing to integrate from  $z = \lambda + 1$  to  $z + h = \lambda + e^{i\theta}$ . We consider the previously introduced conditions on  $h$ .

**Condition (A):**  $|\theta| < \frac{\pi}{n}$ .

**Condition (B):**  $|\theta| < \frac{\pi}{3n}$ .

**Condition (B'):**  $\cos n\theta > \frac{1}{2}$  which is satisfied for values that violate **Condition (A)**.

**Condition (C):**  $|\frac{\theta}{2}| < \arcsin \frac{1}{2n}$ , which is more severe than  $|\theta| < \frac{1}{n}$  and therefore guaranties **Condition (A)**.

In this example, if  $(\Gamma)$  is the circle with center  $\lambda$  and radius 1, the step size must be reduced in such a way that more than  $2n$  intervals are considered to satisfy **Condition (A)**, or even  $6n$  and  $2\pi n$  intervals with **Condition (B)** and **Condition (C)** respectively.

Practically, we consider that **Condition (C)** implies **Condition (A)**, as long as the linear approximation is valid. Problems may occur when  $\Phi'_z$  vanishes. The following example illustrates such a situation.

**Example 3.3 (Critical situation)** Let us consider the matrix of Example 3.1. For  $z = 1/2$ ,  $\Phi_{1/2}(h) = 1 - 4h^2$ , and  $\Phi'_{1/2}(0) = 0$ , and the conditions become

**Condition (A):**  $h \notin \mathbb{R}$  or  $|h| < 1/2$ ,

**Condition (B):**  $|h| < 1/2$ ,

**Condition (B'):**  $|h| < 1/2$ ,

**Condition (C):** is satisfied for all  $h \in \mathbb{C}$ .

## 4 Implementation

In this section, we describe the numerical implementation of our method. Strategies for including new points and a procedure for safely computing the determinants are given.

### 4.1 Avoiding overflows and underflows

The implementation of our method requires the computation of

$$\Phi_z(h) = \frac{\det((z+h)I - A)}{\det(zI - A)}.$$

In order to avoid underflow or overflow, we proceed as follows:

For any non singular matrix  $M \in \mathbb{C}^{n \times n}$ , let us consider its LU factorization  $PM = LU$  where  $P$  is a permutation matrix of signature  $\sigma$ . Then  $\det(M) = \sigma \prod_{i=1}^n (u_{ii})$  where  $u_{ii} \in \mathbb{C}$  are the diagonal entries of  $U$ . If the matrix  $M$  is not correctly scaled, the product  $\prod_{i=1}^n (u_{ii})$  may generate an overflow or an underflow. To avoid this, the determinant is characterized by the triplet  $(\rho, K, n)$  so that

$$\det(A) = \rho K^n \quad (13)$$

where:

$$\begin{aligned} \rho &= \sigma \prod_{i=1}^n \frac{u_{ii}}{|u_{ii}|}, \quad (\rho \in \mathbb{C} \text{ with } |\rho| = 1), \text{ and} \\ K &= \sqrt[n]{\prod_{i=1}^n |u_{ii}|} \quad (K > 0). \end{aligned}$$

The quantity  $K$  is computed through its logarithm:

$$\log(K) = \frac{1}{n} \sum_{i=1}^n \log(|u_{ii}|).$$

By this way, the exact value of the determinant is not computed, as long as the scaling of the matrix is not adequate.

In section 3, it was indicated that our algorithm will heavily be based on the computation of  $\Phi_z(h) = \det(I + hR(z))$ . For  $h$  of moderate modulus, the determinant does not overflow. This can be verified since

$$\Phi_z(h) = \frac{\det((z+h)I - A)}{\det(zI - A)} = \frac{K_2^n \rho_2}{K_1^n \rho_1},$$

where  $\det(zI - A)$  and  $\det((z+h)I - A)$  are respectively represented by the triplets  $(\rho_1, K_1, n)$  and  $(\rho_2, K_2, n)$ . Before raising to power  $n$ , to protect from under- or overflow, the ratio  $K_2/K_1$  must be in the interval  $[\frac{1}{\sqrt[n]{M_{fl}}}, \sqrt[n]{M_{fl}}]$  where  $M_{fl}$  is the largest floating point number. When this situation is violated, intermediate points must be inserted between  $z$  and  $z+h$ .

## 4.2 Estimating the derivative

An easy computation shows that the derivative  $\Phi'_z(0)$  can be expressed by :

$$\Phi'_z(0) = \text{trace}(R(z)). \quad (14)$$

The evaluation of this simple expression involves many operations, as we show it now. By using the LU-factorization  $P(zI - A) = LU$  which is available at  $z$ , and by using (14), we may compute  $\Phi'_z(0) = \sum_{i=1}^n u_i^* l_i$ , where  $l_i = L^{-1}e_i$  and  $u_i = (U^*)^{-1}e_i$ , with  $e_i$  being the  $i$ -th column of the identity matrix. When  $A$  is a sparse matrix, the factors  $L$  and  $U$  are sparse but not the vectors  $u_i$  and  $l_i$ . Therefore, the whole computation involves  $2n$  sparse triangular systems. Experiments showed that they involve more operations than the LU factorization of the matrix. Approximations of the trace of the inverse of a matrix have been investigated. They involve less operations than use of the LU factorization but they are only valid for symmetric or hermitian matrices [3, 9].

If the derivative is approximated by its first order approximation, sparsity helps. More specifically, given  $z$  and  $z+h$ , the derivative of  $\Phi_z$  at 0 is estimated by

$$\Phi'_z(0) \approx \frac{\Phi_z(s) - 1}{s},$$

where  $s = \alpha h$  with  $\alpha = \min(10^{-6}\mu/|h|, 1)$ , and  $\mu = \max_{z \in \Gamma} |z|$ . Therefore, the computation imposes an additional LU factorization for evaluating the quantity  $\Phi_z(s)$ . It is known that, for a sparse matrix, the sparse LU factorization involves much less operations than its dense counterpart.

## 4.3 Test for including new points

In this subsection, we describe a heuristic procedure for including new points in the interval  $[z, z+h]$ . In section 3, we introduced **Condition (B)** which is more severe than **Condition (A)** but might be easier to verify, and we proposed to test its linear approximation called **Condition (C)**. Unfortunately Example 3.3 has exhibited that **Condition (C)** may be satisfied while **Condition (B')** and therefore **Condition (B)** is violated. To increase our confidence in accepting the point  $z+h$ , we simultaneously check **Condition (C)** and **Condition (B')**.

When **Condition (C)** is violated, we insert  $M$  regularly spaced points between  $z$  and  $z+h$  where

$$M = \min \left( \lceil |h| |\Phi'_z(0)| \rceil, M_{max} \right), \quad (15)$$

with  $M_{max}$  being some user defined parameter.

In addition, we insist that **Condition (C)** is satisfied at each bound of the segment  $[z, z+h]$ . Therefore, on exit, the condition  $|h| < \frac{1}{|\Phi'_{z+h}(0)|}$  must also be guaranteed. When it is violated, we insert the point  $z+h/2$  in the list.

The following example illustrates the effect of this step size control.

**Example 4.1** Let  $A$  be the random matrix :

$$A = \begin{pmatrix} -0.63 & 0.80 & 0.68 & 0.71 & -0.31 \\ -0.81 & 0.44 & -0.94 & 0.16 & 0.93 \\ 0.75 & -0.09 & -0.91 & -0.83 & -0.70 \\ -0.83 & -0.92 & 0.03 & -0.58 & -0.87 \\ -0.26 & -0.93 & -0.60 & -0.92 & -0.36 \end{pmatrix}.$$

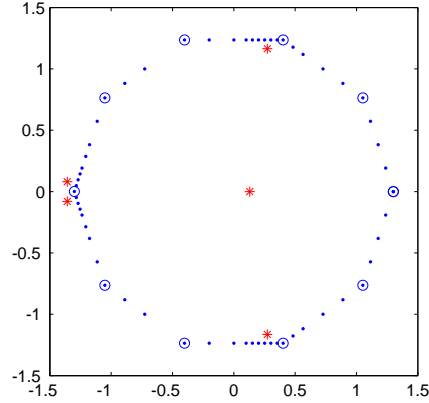


Figure 1: Example 4.1. The eigenvalues are indicated by the stars. The polygonal line is defined by the 10 points with circles; the other points of the line are automatically introduced to insure the conditions as specified in section 4.3 ( $M_{max} = 1$  in (15)).

The polygonal line ( $\Gamma$ ) is determined by 10 points regularly spaced on the circle of center 0 and radius 1.3. In Figure 1, are displayed the eigenvalues of  $A$ , the line ( $\Gamma$ ) and the points that are automatically inserted by the procedure. The figure illustrates that, when the line gets closer to some eigenvalue, the segment length becomes smaller.

#### 4.4 Global algorithm

The algorithm is sketched in Table 1. From a first list  $Z$  of points, it extends the list  $Z$  in order to determine a safe split of the integral (1). The complexity of the algorithm is based on the number of computed determinants. For each  $z \in Z$ , the numbers  $\det(zI - A)$  and  $\Phi'_z(0)$  are computed; they involve two evaluations of the determinant. Therefore, for  $N$  final points in  $Z$ , the complexity can be expressed by:

$$\mathcal{C} = 2\mathcal{L}_{LU}N,$$

where  $\mathcal{L}_{LU}$  is the number of operations involved in the complex LU factorization of  $zI - A$ .

When the matrix  $A$  is real and, assuming that the polygonal line ( $\Gamma$ ) is symmetric w.r.t. the real axis and intersects it only in two points, half of the computation can be saved since

$$N_{\Gamma} = \frac{1}{\pi} \mathcal{I} \left( \int_{\Gamma_+} \frac{f'(z)}{f(z)} dz \right),$$

where  $(\Gamma_+)$  is the upper part of  $(\Gamma)$  when split by the real axis, and  $\mathcal{I}(Z)$  denotes the imaginary part of  $Z$ .

ALGORITHM: EIGENCNT**Input** $Z = \{\text{edges of } (\Gamma)\};$  $M_{pts} = \text{maximum number of allowed points};$  $M_{max} = \text{maximum number of points to insert simultaneously};$ **Output** $neg = \text{number of eigenvalues surrounded by } (\Gamma);$ Status( $Z$ )=-1;**while** Status( $Z$ ) $\neq 0$  **and** length( $Z$ ) <  $M_{pts}$ ,  **for**  $z \in Z$  such that Status( $z$ )=-1,    Compute  $\det(zI - A)$  and  $\Phi'_z(0)$ ;    Status( $z$ ) = 1;  **end**  **for**  $z \in Z$  such that Status( $z$ )=1,    **if** **Condition (C)** not satisfied at  $z$ ,      Generate  $M$  points  $\tilde{Z}$  as in (15);       $Z = Z \cup \tilde{Z}$ ; Status( $\tilde{Z}$ )=-1;    **elseif** **Condition (B')** not satisfied at  $z + h$ ;       $Z = Z \cup \{z + h/2\}$ ; Status( $z + h/2$ )=-1;    **else**      Status( $z$ )=0;    **end**  **end****if** no new points were inserted in  $Z$ ;  **for**  $z \in Z$ ,    **if** **Condition (C)** is backwardly violated;       $Z = Z \cup \{z - h/2\}$ ; Status( $z - h/2$ )=-1;  **end****end****end**Integral =  $\sum_{z \in Z} \text{Arg}(\Phi'_z(0))$ ;  $neg = \text{round}(\text{Integral} / 2\pi)$ ;Table 1: Algorithm for counting the eigenvalues surrounded by  $(\Gamma)$ .

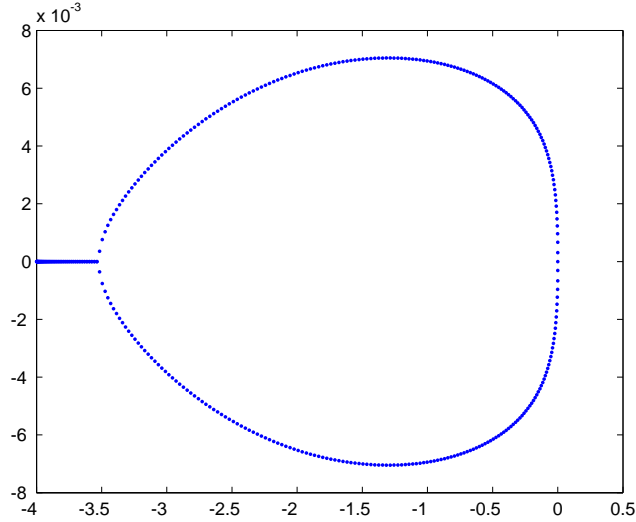


Figure 2: Spectrum of the matrix of Example 5.1.

## 5 Numerical tests

The tests are run on a laptop Dell (Processor Intel Core i7-2620M CPU, clock: 2.70 GHz, RAM: 4 GB). The program `eigcnt` is coded in Matlab.

In the following tests, we describe the performances of the algorithm for three real matrices chosen from the set Matrix Market [1]. The maximum inserted points in an interval is  $M_{max} = 10$ . When  $(\Gamma)$  is symmetric w.r.t. the real axis, only half of the integration is performed. The storage of the matrices is kept sparse (except for computing the spectra of the matrices of the two first examples).

**Example 5.1 (Matrix ODEP400A)** *This matrix is a model eigenvalue problem coming from an ODE.*

$n$	$\ A\ _1$	Spectral radius	Spectrum included in
400	7	4.00	$[-4, 4.38e-4] \times [-0.01, 0.01]$

This matrix is of small order and its spectrum is displayed in Figure 2.

The first experiment consists to focusing on the right part of the spectrum by defining a regular polygon of 10 vertices; the polygon is centered in the origin, symmetric w.r.t. with the real axis as shown in figure 3 (only its upper part is drawn), and of radius  $R = 10^{-3}$ . Five eigenvalues were correctly found as surrounded by the polygon. Some statistics are displayed in the first line of Table 2.

The second experiment focuses on the bifurcation between real and complex eigenvalues in the neighborhood of  $-3.5$ . In the box  $[-4, -3.4] \times [-10^{-4}i, 10^{-4}i]$ , 89 eigenvalues are counted (see the statistics in the second line of Table 2). The aspect ratio of the box is large. The refining process proceeds in 16 steps to produce 1519 intervals from the initial four. If the integral is computed by the



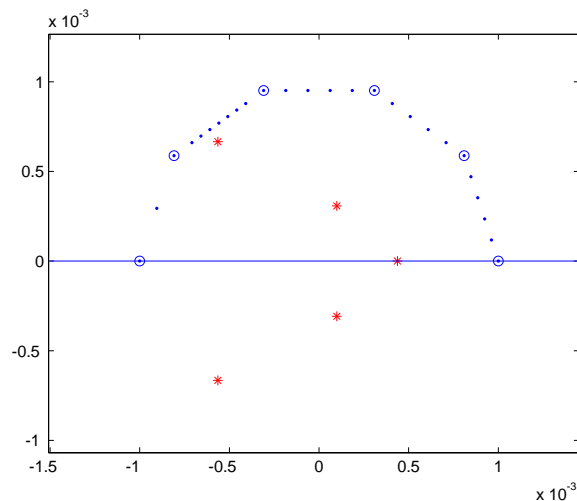


Figure 3: Example 5.1: First experiment on the right end of the spectrum.

Table 2: Statistics for Example 5.1.

	Nb. of eigenvalues in $(\Gamma)$	nb. of intervals	elapsed time
Exper. 1	5	25	5.6e-2 s
Exper. 2	89	1519	1.1 s

relation (8) at each step (hence even if the necessary conditions for correctness are not satisfied), it would only have been correct at the fifth step and after ; this corresponds to 825 intervals. This illustrates the loss in efficiency which is imposed by the constraint for a safe computation.

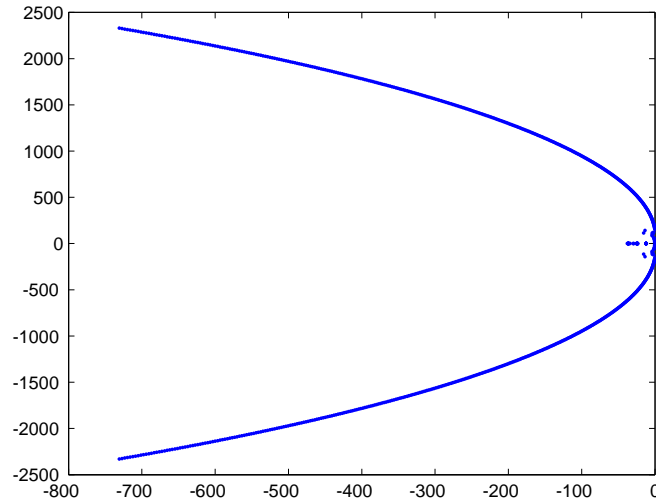
**Example 5.2 (Matrix TOLS2000)** *This matrix comes from a stability analysis of a model of an airplane in flight.*

$n$	$\ A\ _1$	Spectral radius	Spectrum included in
2000	$5.96 \times 10^6$	$r = 2.44 \times 10^3$	$[-750, 0] \times [-r, +r]$

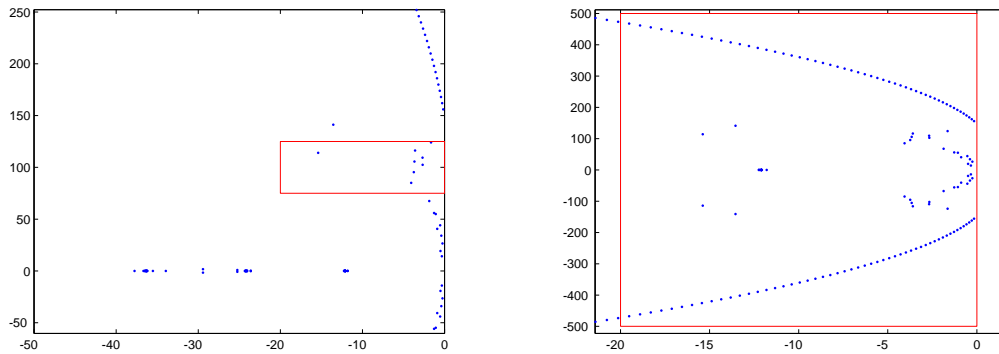
Two experiments consider the right part of the spectrum. A first box  $[-20, 0] \times [75i, 125i]$  is not symmetric w.r.t. the real axis. Therefore, the integration is not reduced. Eight eigenvalues are decoupled. The second box  $[-20, 0] \times [-500i, 500i]$  is symmetric w.r.t. the real axis but it includes 542 eigenvalues. The statistics are reported in Table 3. In Figure 4, the spectrum and two zooms on it are displayed.

	Nb. of eigenvalues in $(\Gamma)$	nb. of intervals	elapsed time
Exper. 1	8	2611	11.0 s
Exper. 2	542	15669	57.7 s

Table 3: Statistics for Example 5.2.



Entire spectrum of the matrix TOLS2000



Exper. 1: Box= $[-20, 0] \times [75i, 125i]$

Exper. 2: Box= $[-20, 0] \times [-500i, 500i]$

Figure 4: Example 5.2: Spectrum of the matrix (up) and zooms on the two regions of experiments.

**Example 5.3 (Matrix E40R5000)** *This sparse matrix comes from modeling 2D fluid flow in a driven cavity, discretized on a  $40 \times 40$  grid and with a Reynolds number is  $Re = 5000$ .*

$n$	$\ A\ _1$	Spectral radius	Spectrum included in
17281	$1.21 \times 10^2$	$r=65.5$ (*)	$[0, 20.2] \times [-r, +r]$

(\*) Estimated by the Matlab procedure `eigs`.

This example shows the reliability of the proposed procedure. Computing the 6 eigenvalues of largest real part with the Matlab procedure `eigs` (it imple-

ments the ARPACK code) returns the eigenvalues  $\{8.3713 \pm 64.653i, 8.8025 \pm 64.876i, 16.203, 20.166\}$ . By increasing the number  $p$  of requested eigenvalues, only a few of them converged: for instance for  $p = 20$ , only the two rightmost were found. Increasing even further up to  $p = 100$ , 14 eigenvalues were given back, including the already computed two rightmost and 12 additional ones with real parts belonging to the interval  $[12.2, 12.9]$ . Therefore, the user is inclined to ask for the exact situation in this region. Defining the rectangle  $\Gamma = \Gamma_+ \cap \Gamma_-$  where  $\Gamma_+ = (14, 14 + 2i, 12 + 2i, 12)$  and where  $\Gamma_-$  is the symmetric of  $\Gamma_+$  w.r.t. the real axis, the procedure `eigcnt` returns

Number of eigenvalues in $(\Gamma)$	number of intervals	elapsed time
116	7986	54 h 42 mn

Table 4: Statistics for Example 5.3.

Actually, the right number of eigenvalues was already given before the last refining step with 3994 intervals. Taking into account the result of the experiment, after several tries of shifts in `eigs`, all the 116 eigenvalues surrounded by  $(\Gamma)$  were obtained by requesting  $p = 200$  eigenvalues in the neighborhood of the shift  $\sigma = 13.5$  (elapsed time: 10.2s).

## 6 Conclusion

In this paper, we have developed a reliable method for counting the eigenvalues in a region surrounded by a user-defined polygonal line. The main difficulty to tackle lies in the step control which must be used during the complex integration along the line. The method is reliable but it involves a high level of computation. This is the price to pay for the reliability. In forthcoming works, a parallel version of the method will be developed and implemented. The code involves a high potential for parallelism since most of the determinant computations are independent. A second level of parallelism is also investigated within the computation of a determinant for matrices arising in domain decompositions.

## References

- [1] Matrix market. Service of the Mathematical and Computational Sciences Division / Information Technology Laboratory / National Institute of Standards and Technology. <http://math.nist.gov/MatrixMarket/>.
- [2] M. Adam and M. J. Tsatsomeros. An eigenvalue inequality and spectrum localization for complex matrices. *ETNA*, 15:239–250, 2006.
- [3] Z. Bai and G. H. Golub. Bounds for the trace of the inverse and the determinant of symmetric positive definite matrices, 1996.
- [4] Christopher Beattie and Ilse C.F. Ipsen. Inclusion regions for matrix eigenvalues. *Linear Algebra and its Applications*, 358(1-3):281 – 291, 2003.
- [5] O. Bertrand and B. Philippe. Counting the eigenvalues surrounded by a closed curve. *Siberian Journal of Industrial Mathematics*, 4:73–94, 2001.

- 
- [6] D. Bindel. Error bounds and error estimates for nonlinear eigenvalue problems. In *Householder Meeting 2008*, 2008.
  - [7] R. A. Brualdi and S. Mellendorf. Regions in the complex plane containing the eigenvalues of a matrix. *The American Mathematical Monthly*, 10:975–985, 1994.
  - [8] S. Godunov. Spectral portraits of matrices and criteria of spectrum dichotomy. In L. Atanassova and J. Hezberger, editors, *Third Int'l. IMACS-CAMM Symposium on Computer Arithmetic and Enclosure Methods*, Amsterdam, 1992.
  - [9] G. H. Golub and G. Meurant. *Matrices, Moments and Quadrature with Applications*. Princeton University Press, 2009.
  - [10] M. E. Hochstenbach, D. A. Singer, and P. F. Zachlin. Eigenvalue inclusion regions from inverses and of shifted matrices. *LAA*, 429:2481–2496, 2008.
  - [11] T.-Z. Huang, W. Zhang, and S.-Q. Shen. Regions containing eigenvalues of a matrix. *ELA*, 15:215–224, 2006.
  - [12] W. Rudin. *Real and complex analysis*. McGraw Hill, New York, 1970.
  - [13] R. A. Silverman. *Introductory Complex Analysis*. Dover Publications, Inc. New York, 1972.
  - [14] L. Trefethen. *Pseudospectra of matrices*, pages 234–266. Longman, 1992. Dundee 1991.

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Mathematical tool and previous works</b>	<b>4</b>
2.1	Use of the argument principle . . . . .	4
2.2	Counting the eigenvalues in a region surrounded by a closed curve	4
<b>3</b>	<b>Integrating along a curve</b>	<b>5</b>
3.1	Following a branch of $\log(f(z))$ along the curve . . . . .	5
3.2	Step size control . . . . .	7
<b>4</b>	<b>Implementation</b>	<b>9</b>
4.1	Avoiding overflows and underflows . . . . .	9
4.2	Estimating the derivative . . . . .	10
4.3	Test for including new points . . . . .	10
4.4	Global algorithm . . . . .	11
<b>5</b>	<b>Numerical tests</b>	<b>13</b>
<b>6</b>	<b>Conclusion</b>	<b>16</b>



---

Centre de recherche INRIA Rennes – Bretagne Atlantique  
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex  
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier  
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq  
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique  
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex  
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex  
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex  
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

---

Éditeur  
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)  
<http://www.inria.fr>  
ISSN 0249-6399