Spectral Portait Computation by a Lanczos Method

(Augmented Matrix version)

Osni A. MARQUES[†]
Vincent TOUMAZOU[‡]

Technical Report TR/PA/95/05, February 1995

Abstract

This work describes a software tool intended for the evaluation of spectral portraits of nonsymmetric matrices. This requires to compute the 2-norm of the matrix $(A - zI)^{-1}$, for points z in a discretized region of the complex plane. In order to estimate this 2-norm, i.e, the smallest singular value of $(A - zI)$, a Lanczos procedure is applied to the matrix $\mathcal{H}_a(z)$ where

$$\mathcal{H}_a(z) = \begin{pmatrix} 0 & (A - zI) \\ (A - zI)^* & 0 \end{pmatrix} .$$

The importance and fundamentals of spectral portraits are first reviewed. Then, we describe the sequence of operations required for their evaluation, including the estimation of the required eigenvalues and implementation details. Next, we give a wide range of study cases. Finally, a user's guide is provided, with a description of the required input and output files.

[†]Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique, 42 av. G. Coriolis, 31057 Toulouse Cedex, France, e-mail: marques@cerfacs.fr

[‡]ERIN-ESTIN et Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique, 42 av. G. Coriolis, 31057 Toulouse Cedex, France, e-mail: toumazou@cerfacs.fr

Contents

1 Introduction **4**

2 Spectral portrait description **5**

2.1 Singular Value Decomposition method 6

2.2 Normal Equation method 6

2.3 Augmented Matrix method 6

3 Spectral portrait computation **8**

3.1 Spectral portrait determination 8

3.2 Eigenvalue computation 9

3.2.1 Generalities 9

3.2.2 Method 10

3.2.3 The computation of $\|A\|_2$ 11

3.2.4 The computation of $\|(A - zI)^{-1}\|_2$ 11

3.2.5 Restarting strategy 12

3.2.6 Implementation details 13

4 Numerical results **13**

4.1 Description of the Figures 13

4.2 Test cases 14

4.2.1 W(50), a Wilkinson matrix and a very striking example 14

4.2.2 La Rose matrix 15

4.2.3 Tolosa matrix 16

4.2.4 Pores3 matrix . . . 17

4.2.5 Matrix from Electromagnetism . . . 17

4.3 About CPU time . . . 19

5 Conclusion **19**

A Description of the code **22**

B User's guide **22**

B.1 INPUT files . . . 22

B.2 OUTPUT files . . . 24

B.3 *Matlab* post-processing . . . 25

1 Introduction

Eigenvalue computations are very important for the study of the stability of physical problems.
However, in finite precision arithmetic, some problems may occur in the computation of eigenvalues related with highly nonnormal operators. Tests performed with matrices for which the departure from normality is parametrized, show that the QR method may converge to eigenvalues far away from the exact solutions (see Chatelin and Frayssé (1993)).
For some applications in physics, for instance, one must be sure that all the eigenvalues have a negative real part. If the associated operator is nonnormal, the real part of some computed eigenvalues may become positive. On the other hand, Trefethen has shown that in hydrodynamics, the use of perturbed operators may lead, in several cases, to results matching those of physical experiments (see Trefethen, Trefethen, Reddy, and Driscoll (1993)).
Therefore, the question is : How does a perturbation affect an eigenvalue?
A possible answer is given by the condition number, which can be computed for simple eigenvalues, but in the case of defective multiple eigenvalues only an upper bound exists. The classical tools turn out to be unsuitable.
In order to overcome this, one should examine the eigenvalues of $A + \Delta A$ and not only those of A. Thus, we define an ε-pseudoeigenvalue and an ε-pseudospectrum of A :

- λ is an ε-pseudoeigenvalue of A if

$$\lambda \text{ is an eigenvalue of } A + E \text{ with } \|E\|_2 \leq \varepsilon \|A\|_2$$

- The ε-pseudospectrum of A is defined by

$$\Lambda_\varepsilon(A) = \{z \in \mathbb{C}; z \text{ is an } \varepsilon\text{-pseudoeigenvalue of } A\}$$

We will see in the next section that, for a fixed ε, the border of $\Lambda_\varepsilon(A)$ can be defined as $\{z \in \mathbb{C}; \|A\|_2 \|(A - zI)^{-1}\|_2 = \frac{1}{\varepsilon}\}$.

Godunov (1991) calls the graphical representation of

$$z \longrightarrow log_{10}(\|A\|_2 \|(A - zI)^{-1}\|_2)$$

the **spectral portrait** of A.
Unlike the condition number, which is a first order estimation and allows the study only in a circle around the singularity with radius tending to 0 (asymptotic behaviour), the spectral portrait allows the study of the influence of the singularity in a larger topological neighbourhood.
In a previous report (Marques and Toumazou (1995)), we have listed three different

ways to compute the spectral portrait and focused on the Normal Equation method. Because of the high condition number of this method for computing $\|(A - zI)^{-1}\|_2$, some problems with accuracy and pigmentation of the spectral portrait in the neighbourhood of some singularities , occured with nonnormal matrices. We will study now the third method, called the Augmented Matrix method.

2 Spectral portrait description

By means of Turing's theorem, one can show that $\{z \in \mathbb{C}; z$ is an eigenvalue of $A + \Delta A;\ \|\Delta A\|_2 \leq \varepsilon \|A\|_2\}$ is equivalent to $\{z \in \mathbb{C};\ \|(A - zI)^{-1}\|_2 \|A\|_2 \geq \frac{1}{\varepsilon}\}$.
The proof can be found in the book *Non-normal matrices and Pseudo-Eigenvalues* writen by Trefethen.
Therefore, to compute a spectral portrait, the main problem consists in the evaluation of

$$z \longrightarrow \|(A - zI)^{-1}\|_2 \text{ for } z \in \mathbb{C}\ . \tag{1}$$

We recall that $\|B\|_2 = \sigma_{max}(B)$ and $\|B^{-1}\|_2 = \dfrac{1}{\sigma_{min}(B)}$, where B is an $n \times n$ matrix and $\{\sigma_i(B)\}_{i=1}^{n}$ its singular values. We denote by $\sigma_{min}(B)$ and $\sigma_{max}(B)$ the smallest and the largest singular value of B, respectively.

Equation (1) can be rewritten as

$$z \longrightarrow \frac{1}{\sigma_{min}(A - zI)} \tag{2}$$

which requires the determination of the smallest singular value of a matrix.

The evaluation of (2) can be performed in different ways. Three of them were listed in Marques and Toumazou (1995), namely the Singular Value Decomposition (SVD) method, the Normal Equation method and the Augmented Matrix method. In that report, the Normal Equation method was examined and its numerical results were compared with the ones provided by the SVD method.
In the next sections, we will briefly review the SVD method, the Normal Equation method and their respective drawbacks (CPU time, condition number) with respect to the computation of $\sigma_{min}(A - zI)$. The Augmented Matrix method will then be described in more detail.

2.1 Singular Value Decomposition method

This method (see Golub and Van Loan (1989)) is very reliable but computationally expensive. The corresponding condition number for the computation of $\sigma_{min}(A-zI)$ is given by

$$CN(svd) = \frac{\|(A-zI)\|_2}{\sigma_{min}(A-zI)} \ .$$

The proof can be found in Marques and Toumazou (1995).

2.2 Normal Equation method

We compute $\sigma_{min}(A-zI)$ in the following way :

$$\sigma_{min}(A-zI) = \sqrt{\lambda_{min}((A-zI)^*(A-zI))}$$

where $(A-zI)^*$ denotes the conjugate transpose of $(A-zI)$.
Since we only need to compute the smallest eigenvalue, we can use a projection method. However, the drawback of this method is that the condition number for the computation of $\sigma_{min}(A-zI)$ is

$$CN(ne) \sim (CN(svd))^2 \ .$$

In Marques and Toumazou (1995), the reader can find the proof and numerical examples for which such high condition number causes problems.

2.3 Augmented Matrix method

This method is based on the use of the augmented matrix $\mathcal{H}_a(z) = \begin{pmatrix} 0 & (A-zI) \\ (A-zI)^* & 0 \end{pmatrix}$ to compute the singular values of $(A-zI)$.
Since

$$sp((A-zI)^*(A-zI)) = \{(\sigma_1(A-zI))^2, (\sigma_2(A-zI))^2, ..., (\sigma_n(A-zI))^2\} \ ,$$

we can write

$$\begin{aligned} sp(\mathcal{H}_a(z)) &= \{-\lambda_n(\mathcal{H}_a(z)), ..., -\lambda_1(\mathcal{H}_a(z)), \lambda_1(\mathcal{H}_a(z)), ..., \lambda_n(\mathcal{H}_a(z))\} \\ &= \{-\sigma_n(A-zI), ..., -\sigma_1(A-zI), \sigma_1(A-zI), ..., \sigma_n(A-zI)\} \end{aligned}$$

We recall that $sp(B)$, $\lambda(B)$ and $\sigma(B)$ denote the spectrum, an eigenvalue and a singular value of any matrix B, respectively.

Thus to determine $\|(A-zI)^{-1}\|_2$, we have to compute $\sigma_1(A-zI) = \lambda_1(\mathcal{H}_a(z))$ which is an eigenvalue of the hermitian matrix $\mathcal{H}_a(z)$.

Analogously, we have $\|A\|_2 = \lambda_{max}\begin{pmatrix} 0 & A \\ A^* & 0 \end{pmatrix} = \lambda_{max}(\mathcal{H}_a(0))$.

The Augmented Matrix method works with a matrix whose size is twice as large as the corresponding matrices of the other two methods. However, we can profit from the block structure of $\mathcal{H}_a(z)$ and its hermitian property. Therefore, we never store $\mathcal{H}_a(z)$ but only A.
The condition number for the computation of $\sigma_{min}(A-zI)$ satisfies

$$CN(am) \leq \sqrt{2}\sqrt{n}\ CN(svd)\ .$$

<u>Proof :</u>

The condition number of the eigenvalue $\lambda_1(\mathcal{H}_a(z)) = \sigma_1(A-zI)$ is

$$CN(am) = \frac{\|\mathcal{H}_a(z)\|_2\ \|x_*\|\ \|x\|}{|\lambda_1(\mathcal{H}_a(z))|\ |x_* x|}\ ,$$

where x and x_* are the right and left eigenvectors of $\mathcal{H}_a(z)$ corresponding to $\lambda_1(\mathcal{H}_a(z))$.
Since $\mathcal{H}_a(z)$ is hermitian, we have $x_* = x$ and $\|x\|_2 = 1$.
Thus, we can write

$$CN(am) = \frac{\|\mathcal{H}_a(z)\|_2}{|\lambda_1(\mathcal{H}_a(z))|}\ ,$$

but

$$\|\mathcal{H}_a(z)\|_F^2 = \sum_{i,j} a_{ij}^2 + \sum_{i,j} a_{ij}^{*2} = \|(A-zI)\|_F^2 + \|(A-zI)^*\|_F^2 = 2\|(A-zI)\|_F^2\ ,$$

where a_{ij} and a_{ij}^* are the coefficients of the matrices $(A-zI)$ and $(A-zI)^*$, respectively.
Because $\|(A-zI)\|_F \leq \sqrt{n}\|(A-zI)\|_2$, we can write

$$\|\mathcal{H}_a(z)\|_2 \leq \sqrt{2}\sqrt{n}\|(A-zI)\|_2$$

and

$$CN(am) \leq \frac{\sqrt{2}\sqrt{n}\|(A-zI)\|_2}{|\lambda_1(\mathcal{H}_a(z))|} \leq \frac{\sqrt{2}\sqrt{n}\|(A-zI)\|_2}{\sigma_{min}(A-zI)} \leq \sqrt{2}\sqrt{n}\ CN(svd)\ .$$

$\square$

The method chosen to compute $\lambda_{min}(\mathcal{H}_a(z))$ and $\lambda_{max}(\mathcal{H}_a(0))$ is based on the Lanczos method, which is very appropriate for computing the extreme eigenvalues of hermitian matrices.

Our work consists in :

- computing $\lambda_{max}(\mathcal{H}_a(0))$
- discretizing the complex plane
- computing, for each point z

$$\begin{aligned} z \longrightarrow \phi(z) &= log_{10}(\|A\|_2 \|(A - zI)^{-1}\|_2) \\ &= log_{10}[\lambda_{max}(\mathcal{H}_a(0))\lambda_{min}(\mathcal{H}_a(z))] \ . \end{aligned}$$

Because the variations of $\|(A - zI)^{-1}\|_2$ in the neighbourhood of an eigenvalue are extremely stiff, the level curves of $z \longrightarrow \|A\|_2\|(A - zI)^{-1}\|_2$ corresponding to a uniform scale are not very informative. To circumvent this difficulty, we can use rather an **exponential** scale, which amounts to a uniform scale for $z \longrightarrow \phi(z)$.

3 Spectral portrait computation

In this section we examine the sequence of operations required for the evaluation of a spectral portrait. First we list the steps to determine the spectral portrait, as well as the algorithm used to estimate the required eigenvalues. Then, we present the strategies and versions implemented to improve the computational performance.

3.1 Spectral portrait determination

The basic steps required for the determination of the spectral portrait are described below :

- Step 1. Compute $\|A\|_2$.
- Step 2. Define a region of the complex plane by z_1 and z_2.
- Step 3. Discretize the region by $xmesh$ and $ymesh$.
- Step 4. For each point z of the discretized region, compute $\|(A - zI)^{-1}\|_2$.

- Step 5. Graphical processing using *Matlab*.

z_1 and z_2 are the lower left point and higher right point of the discretized space, respectively.
$xmesh$ and $ymesh$ are the number of discretization points on the real and imaginary axis, respectively.

In order to define the region to be studied, we recall that each eigenvalue λ of A satisfies $|\lambda| \leq \|A\|_2$. Thus, for a global analysis of the spectrum, the region under study must contain more than the disk $(0, \|A\|_2)$ because of the possible diffusion of some eigenvalues outside the spectrum (see Figure 7).

Moreover, in the case of real matrices, the spectral portrait is symmetric with respect to the real axis. Thus, if the chosen region includes the real axis, we can restrict the computation to the upper or the lower part of the complex plane.

The computation of $\phi(z)$ requires the determination of $\lambda_{max}(\mathcal{H}_a(0))$ and $\lambda_{min}(\mathcal{H}_a(z))$, which will be examined in the next section.

3.2 Eigenvalue computation

The technique used for the computation of the eigenvalues is based on a Lanczos method applied to $\mathcal{H}_a(z) = \begin{pmatrix} 0 & (A - zI) \\ (A - zI)^* & 0 \end{pmatrix}$.

We use the subroutine `HLDRVS`, which is described in Marques (1994). Some important characteristics of this code are given below.

3.2.1 Generalities

The unique routine `HLDRVS` sets values for control parameters, computes some eigenvalues of $\mathcal{H}_a(z)$ (usually the extreme ones) and, if required, the associated eigenvectors. The algorithm used by `HLDRVS` requires the multiplication of a vector by the matrix $\mathcal{H}_a(z)$, until the convergence for the prescribed number of solutions is reached. However, every time a matrix-vector multiplication has to be performed, the control is returned to the user (reverse communication strategy). Thus, the matrix $\mathcal{H}_a(z)$ does not have to be passed as an argument.

3.2.2 Method

The algorithm used by HLDRVS is based on the Lanczos method, in combination with a modified partial reorthogonalization strategy.
The basic idea consists in the generation of a Krylov basis of vectors $Q_j = [q_1, ..., q_j]$, with $j < n$, such that the projection of $\mathcal{H}_a(z)$ into Q_j leads to a reduced problem, i.e.

$$Q_j^* \mathcal{H}_a(z) Q_j = T_j \ ,$$

where T_j is a symmetric tridiagonal matrix.
Assuming that (θ, y) is an exact eigensolution of the reduced problem, $(\theta, Q_j y)$ is an approximate solution of the original one. We define as $(\tilde{\lambda}, \tilde{x})$ the pair $(\theta, Q_j y)$ computed *in finite precision arithmetic.*
We denote by m the maximal size of projection and p the number of required eigenpairs. Thus the minimal size of the Krylov basis is greater than or equal to p.
tol denotes the threshold for convergence (backward error).

The algorithm is summarized below.

1. *Initialization*:

set $q_0 = 0$ and $\beta_1 = 0$
set $q_1 \neq 0$ so that $q_1^* q_1 = 1$

2. *Lanczos steps*:

for j=1,2,... m

a) compute r_j from q_j
b) $r_j \leftarrow r_j - q_{j-1}\beta_j$
c) $\alpha_j \leftarrow q_j^* r_j$
d) $r_j \leftarrow r_j - q_j \alpha_j$
e) $\beta_{j+1} \leftarrow \sqrt{r_j^* r_j}$
f) $q_{j+1} \leftarrow \frac{1}{\beta_{j+1}} r_j$
g) if required orthogonalize q_j and q_{j+1} against the vectors of Q_{j-1}
h) insert q_j into Q_j, $t_{j,j-1} \leftarrow \beta_j$, $t_{j-1,j} \leftarrow \beta_j$, $t_{j,j} \leftarrow \alpha_j$
i) solve the reduced problem $T_j y_k = y_k \theta_k$
j) check nc, the number of eigenpairs for which
$\|\mathcal{H}_a(z)\tilde{x} - \tilde{\lambda}\tilde{x}\|_2 = \|\beta_{j+1} y_j^{(k)}\|_2 \leq tol\ \|\mathcal{H}_a(z)\|_2$

If $nc \geq p$, exit.

3. *Compute the restarting vector if necessary and Goto 2*

The steps a) and j) will differ according to the matrix to be treated : A or $(A-zI)^{-1}$. We will now examine both cases.

3.2.3 The computation of $\|A\|_2$

We compute $\|A\|_2 = \lambda_{max}(\mathcal{H}_a(0))$.

Step a) of the algorithm corresponds to

$$r_j \leftarrow \mathcal{H}_a(0) q_j \ ,$$

and taking into account the sparsity and the block structure of $\mathcal{H}_a(0)$, we can rather compute

$$r_j^I = A q_j^{II} \text{ and } r_j^{II} = A^* q_j^I$$

where $r_j = (r_j^I, r_j^{II})^T$ and $q_j = (q_j^I, q_j^{II})^T$.

The stopping criterion (step j) of the algorithm is based on the backward error analysis.
The backward error related to the i^{th} eigenpair $(\tilde{\lambda}_i, \tilde{x}_i)$ is given by

$$\eta(0)_i = \frac{\|\mathcal{H}_a(0)\tilde{x}_i - \tilde{\lambda}_i \tilde{x}_i\|_2}{\|\mathcal{H}_a(0)\|_2} \ .$$

The numerator can be computed using information provided by the Lanczos code. Indeed, at the j^{th} step of the algorithm, we have

$$\|\mathcal{H}_a(0)\tilde{x}_i - \tilde{\lambda}_i \tilde{x}_i\|_2 = \beta_{j+1} |y_i^{(j)}| \ ,$$

where β_{j+1} is the normalizing factor associated with the $(j+1)^{th}$ Lanczos vector and $y_i^{(j)}$ is the last component of the eigenvector related to the i^{th} eigenvalue of the reduced problem[1].
Since $\mathcal{H}_a(0)$ is hermitian, $\|\mathcal{H}_a(0)\|_2$ can be estimated by $\lambda_{max}(\mathcal{H}_a(0))$.

3.2.4 The computation of $\|(A-zI)^{-1}\|_2$

We want to compute $\|(A-zI)^{-1}\|_2 = \lambda_{min}(\mathcal{H}_a(z))$.

Since the Lanczos algorithm converges faster to the largest eigenvalues, we compute

[1] provided that $\frac{\beta_{j+1}|y_i^{(j)}|}{\|\mathcal{H}_a(0)\|_2} \geq machineprecision$. See Bennani and Braconnier (1993b).

$\lambda_{max}(\mathcal{H}_a(z)^{-1})$ instead of $\lambda_{min}(\mathcal{H}_a(z))$.
Step a) of the algorithm previously described, becomes

$$r_j \leftarrow \mathcal{H}_a(z)^{-1} q_j \ .$$

Actually, we do not explicitly invert the matrix $\mathcal{H}_a(z)$, but we solve the system $\mathcal{H}_a(z) r_j = q_j$ exploiting its block structure.
In other words, we compute

$$(A - zI)^* r_j^I = q_j^{II} \text{ and } (A - zI) r_j^{II} = q_j^I \ .$$

However, using the subroutine `ZGETRF` from the *LAPACK* library (see Anderson, Bai, Bischof and al. (1992)), we perform only the LU factorization of $(A - zI)$ which allows us to solve both linear systems using the subroutine `ZGETRS`.

It should be noted that, if z corresponds to an eigenvalue, the LU factorization is not possible. In this case, the *LAPACK* library returns an appropriate flag and by default we set $\|(A - zI)^{-1}\|_2 = 10^{16}$ which corresponds to the machine precision.

As in the case of $\|A\|_2$ computation, the stopping criterion associated to the i^{th} eigenpair $(\tilde{\lambda}_i, \tilde{x}_i)$

$$\eta(z)_i = \frac{\|\mathcal{H}_a(z)^{-1}\tilde{x}_i - \tilde{\lambda}_i \tilde{x}_i\|_2}{\|\mathcal{H}_a(z)^{-1}\|_2} \ .$$

can be estimated by the Lanczos code for the numerator and by $\lambda_{max}(\mathcal{H}_a(z)^{-1})$ for the denominator.

As we have seen, we actually compute the largest eigenvalue for both $\|A\|_2$ and $\|(A - zI)^{-1}\|_2$. In some cases, we have to use a restarting strategy.

3.2.5 Restarting strategy

The vector q_1 introduced in the *Initialization* step of the algorithm previously described is defined by

$$q_1 = \frac{1}{\sqrt{n}}(1,\ 1,\ \ldots 1)^* \ .$$

When the basis size reaches the maximal number of steps allowed, m, and the convergence of $\lambda_{max}(\mathcal{H}_a(z))$ is not reached, the Lanczos code is restarted taking

$$q_1 = \frac{\tilde{x}_{max}}{\|\tilde{x}_{max}\|_2}$$

as starting vector, where $\tilde{x}_{max}$ is the eigenvector associated with the largest eigenvalue of $\mathcal{H}_a(z)$.

3.2.6 Implementation details

It should be noted that the sparsity of the matrix is not taken into account. As it can be seen in the *User's Guide* section, the spectral portrait computation code called `portrait` can deal with matrices from the Harwell-Boeing collection (Duff, Grimes, and Lewis (1992)). For the input of the matrices, we use the subroutine `readmt` from the *SPARSKIT* library (Saad (1993)) but such matrices are stored as dense.

4 Numerical results

4.1 Description of the Figures

Before presenting the numerical results, in Table 1 we list the characteristics of the examined problems as well as the corresponding Figures.

Figure	Matrix	Size	Mesh	m	p	tol	Symmetry
3	Wilkinson	50	128×128	35	3	10^{-3}	yes
4	La Rose	10	128×128	10	3	10^{-4}	yes
5	Tolosa	135	128×128	120	3	10^{-4}	yes
6	Pores3	532	50×100	300	3	10^{-4}	no
7	Electromagnetism	288	128×128	150	3	10^{-4}	yes

Table 1: Description of the Figures

As we can see in Table 1, the minimal projection size p is equal to 3 in all cases. This size has shown to be suited for all applications since it is a minimum and we have the possibility to restart.

A value of about 10^{-4} for tol, the backward error, seems to be generally enough to assure a good estimation of the eigenvalue. A smaller tol would lead only to a refinement of the associated eigenvector, which is not required in the analysis.

The scale given with the pictures indicates $z \longrightarrow \phi(z) = log_{10}(\|A\|_2\|(A - zI)^{-1}\|_2)$. However, a value of 16 has been used as a cut off, which corresponds to $-log_{10}\varepsilon$, where ε is the machine precision in double precision arithmetic.

In addition, we have taken profit of the symmetry, as described in section 3.1, as often as possible. The last column of the Table 1 indicates if the symmetry was taken into account or not. For the Wilkinson matrix, for example, we have examined a 128×128 mesh. However, taking the symetry into account, the analysis was restricted to the half upper plane with a 64×128 mesh.

4.2 Test cases

4.2.1 W(50), a Wilkinson matrix and a very striking example

The family of Wilkinson matrices is made of upper bidiagonal matrices. A matrix of size n, denoted by $W(n)$, is defined as follows

$$w_{i,i} = i, \;\; w_{i,i+1} = n \text{ and } w_{ij} = 0 \text{ for } j \neq i \text{ or } i+1,$$

so that its eigenvalues are $\{1, 2, ..., n\}$.

In Figure 1, we have plotted the spectrum of $W(50)$ computed by the QR algorithm available in *Matlab.* In Figure 2, we give the spectrum of $Q^*W(50)Q$ computed in the same way, where Q is an orthonormal matrix. Thus $Q^*W(50)Q$ and $W(50)$ have the same spectrum, $Q^*W(50)Q$ is not bidiagonal but full. As we can see in Figure 2, the computed spectrum is far away from the exact one.

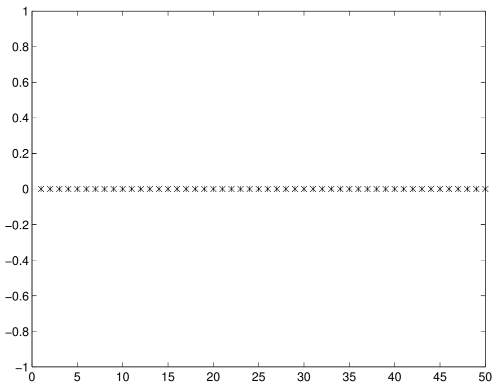

Figure 1: Spectrum of $W(50)$

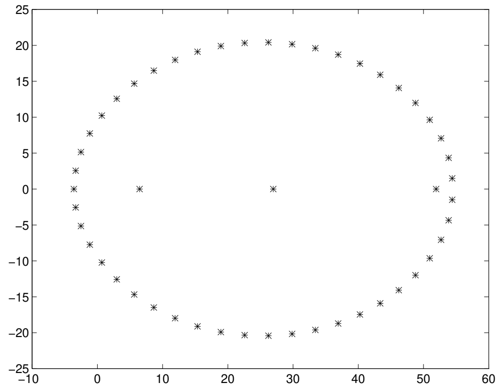

Figure 2: Spectrum of $Q^*W(50)Q$

Such a behaviour can be explained by the examination of the spectral portrait. As we can see in figure 3, even with a double precision arithmetic (10^{-16}) it is very difficult to compute the eigenvalues : a large ellipse encloses the exact spectrum. This ellipse corresponds to the set of eigenvalues of any $A + \Delta A$, where $\|\Delta A\|_2 \leq 10^{-16}\|A\|_2$.

In other words, the spectral portrait shows that all the values inside the ellipse might be erroneously considered as possible eigenvalues (i.e. approximated eigenvalues with a backward error of machine precision).

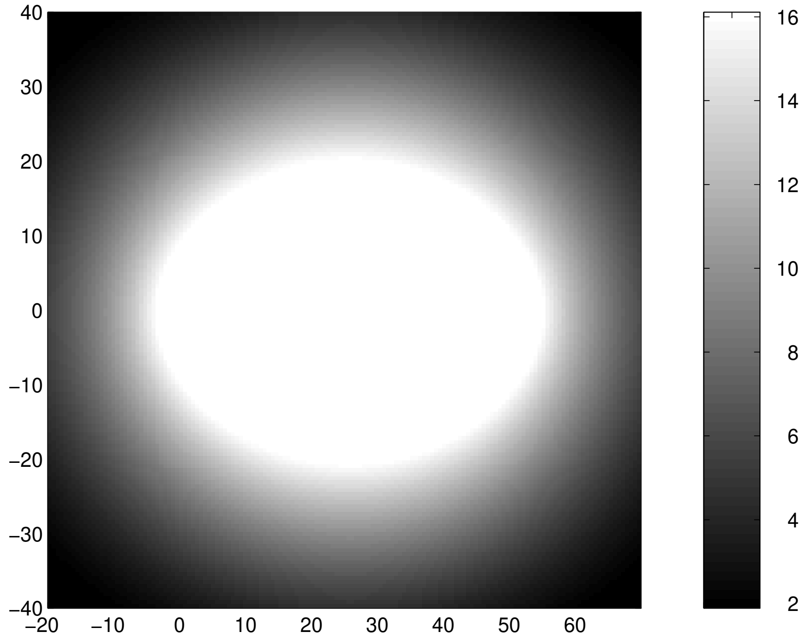

Figure 3: Wilkinson matrix

4.2.2 La Rose matrix

The matrix called La Rose is the companion matrix associated with the polynomial $P(x) = (x-1)^3(x-2)^3(x-3)^3(x-4)$.
It is defined by

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -864 & 4968 & -12492 & 18086 & -16703 & 10290 & -4287 & 1194 & -213 & 22 \end{pmatrix}$$

so that its spectrum has one simple eigenvalue, equal to 4, and three defective eigenvalues of multiplicity 3 : $1, 2$ and 3 respectively. Considering these multiplicities, the spectrum is difficult to compute.

The analysis of the Figure 4 shows a collective behaviour mainly in the neighbourhood of the eigenvalues 2 and 3. In this region, even with a precision equal to 10^{-16}, it becomes very difficult (indeed impossible) to compute the exact eigenvalues.

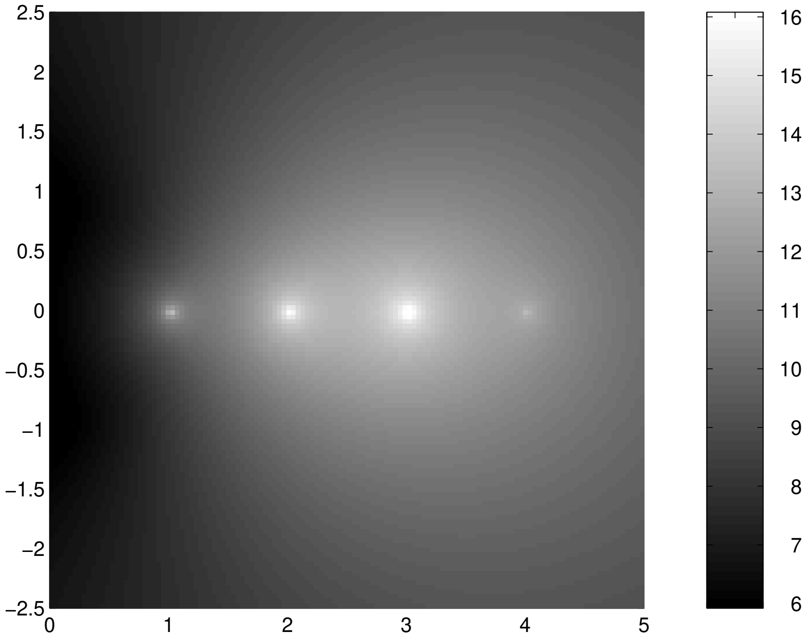

Figure 4: La Rose matrix

Moreover, with this matrix, we have shown that a pigmentation appears in the neighbourhood of the eigenvalues 2 and 3 if the spectral portrait is computed using the Normal Equation method (see Marques and Toumazou (1995)). This problem, related with the high condition number of this method for the computation of $\|(A - zI)^{-1}\|_2$, disappears with the Augmented Matrix method.

4.2.3 Tolosa matrix

This matrix has dimension 135 and was provided by engineers from *Aerospatiale*, who wanted to compute the eigenvalues with largest imaginary part. The matrix is associated with the analysis of a flutter problem (Braconnier, Chatelin, and Dunyach (1995)).
It is available in the Harwell Boeing collection.

With the Normal Equation method, we used a previous point strategy to speed the convergence up. The computed spectral portrait did not correspond to the one computed with the SVD method. As we can see in Figure 5, the spectral portrait is now correctly computed and furthermore a good speed up is achieved with the Augmented Matrix method (see section 4.3).

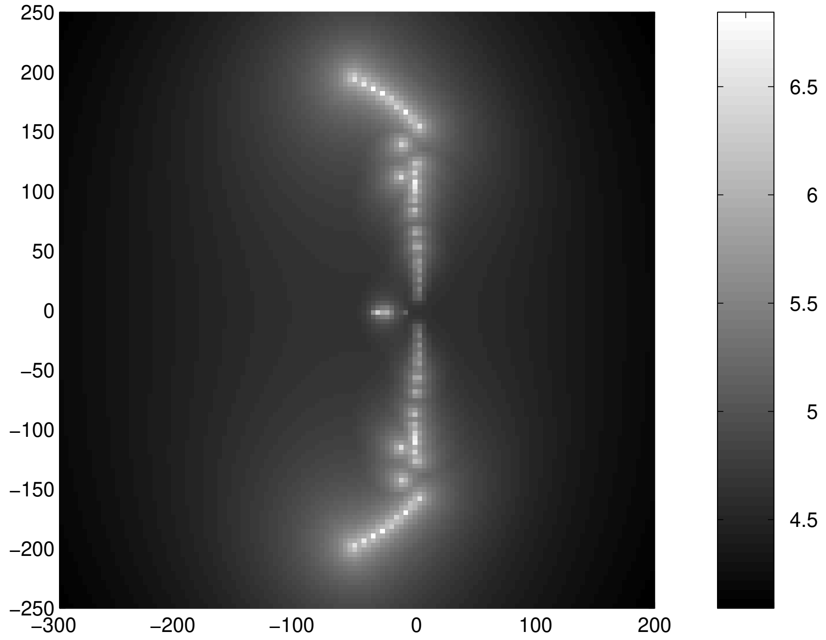

Figure 5: Tolosa matrix

4.2.4 Pores3 matrix

This matrix, from the *Harwell Boeing* collection, has a symmetric pattern but unsymetric entries. It is related with a reservoir simulation and its spectral portrait was previously computed in Carpraux, Erhel, and Sadkane (1993).

As we can see in Figure 6, the spectrum of Pores3 can be considered as stable because the largest value on the scale of the spectral portrait is moderate. Therefore, perturbations less than or equal to 10^{-7} do not lead to significant changes in its eigenvalues.

4.2.5 Matrix from Electromagnetism

This problem was given by F. Collino (INRIA and Cerfacs). This matrix is related with an electromagnetism problem that deals with the diffraction of a transverse-magnetic wave by a periodic $2D$ structure. The solution x is computed using an iterative scheme $x_{k+1} = Ax_k + b_k$ where b_k tend to 0.
As it can be seen on Figure 7, some values larger than 1 might be seen as eigenvalues. This behaviour can explained the divergence of the iterative scheme used to solve the equation. This problem will be detailed in a forthcoming technical report.

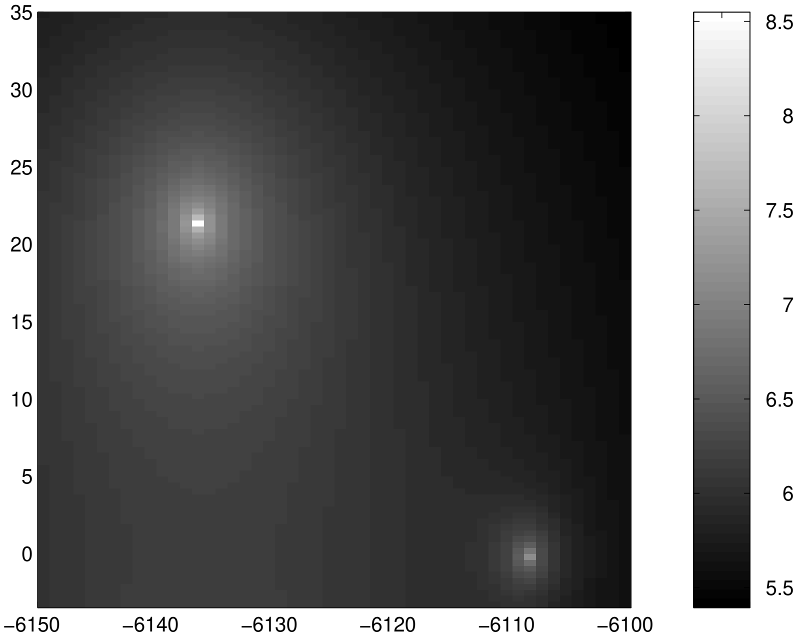

Figure 6: Pores3 matrix

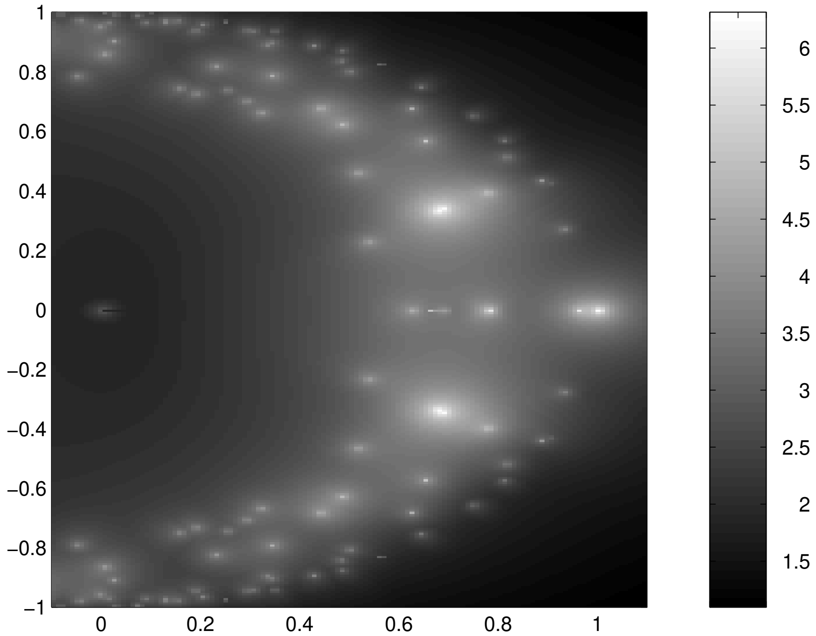

Figure 7: Matrix from Electromagnetism

4.3 About CPU time

In Table 2, we list the CPU time (in seconds) required for the spectral portrait computation of the matrices presented in the previous section.

Matrix	Size	Mesh	tol	CPU time
Wilkinson	50	128×128	10^{-4}	356
La Rose	10	256×256	10^{-4}	340
Tolosa	135	256×256	10^{-4}	69837
Pores3	532	50×100	10^{-4}	14256
Electromagnetism	288	128×128	10^{-4}	15133

Table 2: CPU time

It should be noted that all the spectral portraits described in Table 2 were computed without taking into account the symmetry with respect to the real axis when it was possible, in order to make a fair comparison.
Some CPU time tests were performed with the SVD method. Two matrices, namely La Rose and Tolosa, were treated using the same mesh, 256×256. The CPU time required for the spectral portrait computation was 588 seconds for the matrix La Rose and 220840 seconds for the matrix Tolosa. Therefore, in the case of La Rose, the Augmented Matrix method is about two times faster than the SVD method. In the case of Tolosa, the gain is larger and the Augmented Matrix method is about three times faster than the SVD method.
Thus, we can say that the Augmented Matrix method is clearly faster than the SVD method.

5 Conclusion

This work describes an alternative approach to the SVD method for the evaluation of the spectral portrait. We have used a Lanczos based code to compute some eigenvalues of a matrix $\begin{pmatrix} 0 & (A - zI) \\ (A - zI)^* & 0 \end{pmatrix}$, for distinct values of z.
This method has been designed to cope with the numerical difficulties encountered with the Normal Equation method (i.e. the computation of the eigenvalues of $(A - zI)^*(A - zI)$).
The gain in using the Augmented Matrix method is twofold :

- the computation of $\phi(z)$ is as reliable as with the SVD method,
- the computation of $\phi(z)$ is much faster than with the SVD method.

Furthermore, the structure of the Augmented Matrix can be taken into account so that the storage is kept moderate.

A parallel version of the code is currently being performed as well as a sparse storage version, tuned for the Harwell Boeing collection.

References

E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. GreenBaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, (1992), **LAPACK User's Guide**, SIAM.

M. Bennani and T. Braconnier, (1993a), *Comparative behaviour of eigensolvers on highly nonnormal matrices.* Submitted to Jour. Num. Lin. Alg. Appl.

M. Bennani and T. Braconnier, (1993b), *Stopping criteria for eigensolvers.* Submitted to IMA Num. Anal.

T. Braconnier, F. Chatelin, and J. C. Dunyach, (1995), *Highly Nonnormal Eigenvalue Problems in the Aeronautical*, Japan Journal of Industrial and Applied Mathematics., **12**, 1.

T. Braconnier, F. Chatelin, and V. Frayssé, (1993), *The influence of large nonnormality on the quality of convergence of iterative methods in linear algebra.* Submitted to Lin. Alg. Appl.

T. Braconnier, (1994), *Sur le calcul de valeur propres en précision finie*, Ph. D. dissertation, Université Henri Poincaré, Nancy I.

J. Carpraux, J. Erhel, and M. Sadkane, (1993), *Spectral portrait for non hermitian large sparse matrices*, Tech. Rep. 777, INRIA / IRISA, unité de recherche INRIA–Rennes, Campus de Beaulieu, 35042 Rennes cedex, France.

F. Chatelin and V. Frayssé, (1993), *Qualitative Computing : elements of a theory for finite precision computation.* Lecture Notes for the Comett European Course, June 8-10, Orsay.

F. Chatelin, (1988), **Valeurs propres de matrices**, Masson, Paris.

F. Chatelin, (1994), *Influence of nonnormality on the reliability of iterative methods in Linear Algebra.* ILAS 94, Rotterdam.

I. Duff, R. Grimes, and J. Lewis, (1992), *User's Guide for the Harwell-Boeing Sparse Matrix Collection*, Tech. Rep. TR/PA/92/86, CERFACS.

V. Frayssé, (1992), *Reliability of computer solutions (Sur la fiabilité des calculs sur ordinateurs)*, Ph. D. dissertation, Institut National Polytechnique de Toulouse.

S. K. Godunov, (1991), *Spectral portraits of matrices and criteria of spectrum dichotomy*, in **International symposium on computer arithmetic and scientific computation**, J. Herzberger and L. Atanassova, eds., Oldenburg, Germany, North-Holland.

G. Golub and C. Van Loan, (1989), **Matrix computations**, John Hopkins University Press. Second edition.

O. Marques and V. Toumazou, (1995), *Spectral portrait computation by a Lanczos method (Normal Equation method)*, Tech. Rep. TR/PA/95/02, CERFACS.

O. Marques, (1994), *An Interactive Complex Hermitian-Lanczos Eigensolver*, Tech. Rep. TR/PA/94/16, CERFACS.

A. Ruhe, (1994), *The Rational Krylov algorithm for large nonsymmetric eigenvalues-mapping the resolvent norms (pseudospectrum).* Sparse Days at St-Girons, France.

Y. Saad, (1993), *SPARSKIT: a basic tool kit for sparse matrix computations. Version2.*

L. N. Trefethen, A. E. Trefethen, S. C. Reddy, and T. A. Driscoll, (1993), *Hydrodynamics stability without eigenvalues*, Science, **261**, 578–584.

L. N. Trefethen, (1991), *Pseudospectra of matrices*, in **14th Dundee Biennal Conference on Numerical Analysis**, D. F. Griffiths and G. A. Watson, eds.

L. N. Trefethen, (In preparation), **Non-normal matrices and Pseudo-Eigenvalues**, Department of Computer Science, Cornell University.

J. Wilkinson, (1965), **The Algebraic Eigenvalue Problem**, Oxford University Press.

A Description of the code

The main subprogram for the spectral portrait evaluation is `portrait.f`, which has the following characteristics :

- It defines work arrays whose dimensions are specified by the parameters `NMAX` and `MAXW`. In order to match a particular application, these parameters can be redefined by the user as follows :
  `NMAX` must be greater than or equal to n, the dimension of the matrix.
  `MAXW` corresponds to the workspace required by the eigenvalue computation code.
- It reads the variables that define the type of the matrix to be studied and the mesh on the complex plane. It also reads the entries of the matrix.
- It calls the subroutine `readHB`, which is an interface for `READMT` from *SPARSKIT* (see Saad (1993)), for the input of matrices from the Harwell-Boeing collection.
- It calls `maillage` for the computation of $\|(A - zI)^{-1}\|_2$ and `norme` for the computation of $\|A\|_2$. These subroutines are interfaces for the eigenvalue computation package.

B User's guide

B.1 INPUT files

This section describes the input files required by the spectral portrait evaluation code, which are the following

- `STRATEGY` : This file contains the type of matrix to be treated.

  ```
  H/B matrix? 0 for no, 1 for y
  0
  If H/B matrix give the name (*_rua)
  no name
  ```

- `PARAMETER` : This file contains the mesh parameters and the control values for the eigenvalue package. The bounds for the real part are given by `xmin` and `xmax` while the bounds for the imaginary part are given by `ymin` and `ymax`. The partitioning of the region is defined by `xmesh` and `ymesh`.

```
Give xmin
-4.0
Give xmax
4.0
Give ymin
0.0
Give ymax
1.0
Give xmesh
256
Give ymesh
256
Give the precision(backward error)
1.e-4
Give the size of the matrix
7
Give the maximum number of step
6
Give the number of required eigenpairs
3
```

- `MATRIX` : When the matrix does not belong to the Harwell-Boeing collection, it must be specified in this file, column by column. See example below.

```
-2.0
0.0
0.0
0.0
0.0
0.0
0.0
25.0
-3.0
0.0
0.0
0.0
0.0
0.0
0.0
10.0
2.0
0.0
0.0
0.0
0.0
```

```
0.0
3.0
15.0
0.0
0.0
0.0
0.0
0.0
3.0
3.0
15.0
3.0
0.0
0.0
0.0
3.0
3.0
3.0
10.0
-2.0
0.0
0.0
0.0
0.0
0.0
0.0
25.0
-3.0
```

B.2 OUTPUT files

This section describes the OUTPUT files which are

- `OUTPUT1` : gives the real part of each point of the mesh.
- `OUTPUT2` : gives the imaginary part of each point of the mesh.
- `OUTPUT3` : gives the value of $\|A\|_2 \|(A - zI)^{-1}\|_2$ for each point of the mesh, associated with `OUTPUT1` and `OUTPUT2`.

These files will be used by the *Matlab* post-processing, in order to visualize the spectral portrait.

B.3 *Matlab* post-processing

The spectral portrait will be visualized using the *Matlab* routine `dessin.m`. After reading the OUTPUT files, this routine uses the intrinsic *Matlab* functions, `pcolor` and `colorbar`, and the subroutines `forme.m` and `reforme.m`.