Electronic Transactions on Numerical Analysis.
Volume 5, pp. 62-76, September 1997.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

ARNOLDI-FABER METHOD FOR LARGE NON HERMITIAN EIGENVALUE PROBLEMS*

VINCENT HEUVELINE † AND MILOUD SADKANE ‡

Abstract. We propose a restarted Arnoldi's method with Faber polynomials and discuss its use for computing the rightmost eigenvalues of large non hermitian matrices. We illustrate, with the help of some practical test problems, the benefit obtained from the Faber acceleration by comparing this method with the Chebyshev based acceleration. A comparison with the implicitly restarted Arnoldi method is also reported.

Key words. Krylov space, block Arnoldi, Faber polynomials, Schwarz-Christoffel.

AMS subject classifications. 65F10.

1. Introduction. The Arnoldi-Chebyshev method [29, 14, 16, 31, 17] constitutes one of the more attractive acceleration techniques for computing the right-most eigenvalues of large matrices. The idea behind this technique is to start with the popular Arnoldi method [1] which approximates the outermost part of the spectrum of the matrix [28], and then restart the Arnoldi process, after a small number of iterations, using polynomials that amplify the components of the required eigendirections while dumping those in the unwanted ones.

The algorithms discussed in the above mentioned references assume that the unwanted eigenvalues lie in an ellipse symmetric with respect to the real axis if the matrix is real as in [29, 14, 16, 31] or an ellipse oblique in the general complex non hermitian case as in [17]. The assumption that the unwanted eigenvalues lie in an ellipse is not without drawbacks since, as pointed out by Saad [30, p. 239], "... the choice of ellipses as enclosing regions in Chebyshev acceleration may be overly restrictive and ineffective if the shape of the convex hull of the unwanted eigenvalues bears little resemblance with an ellipse". He proposed the use of the least squares polynomials on some polygonal domains [30].

This paper is concerned with the computation of a few right-most eigenvalues and the corresponding eigenvectors of a large complex non hermitian matrix A. As in [30], we construct a polygon that contains the unwanted eigenvalues approximated by Arnoldi's process, but the computation of the polynomials is different. We use the Faber polynomials [7] for restarting the method.

This work is inspired by the contribution of Starke and Varga [33] who used Faber polynomials in the context of linear systems and who showed the nearly optimal properties that the Faber polynomials possess, and by the work of Trefethen [34] on the Schwarz-Christoffel mapping function, a powerful tool for constructing the Faber polynomials.

2. Arnoldi-Faber method for computing eigenvalues. From a starting vector v_1 with $\|v_1\|_2 = 1$, the Arnoldi process generates an orthonormal basis $V_m = [v_1, \ldots, v_m]$ of the Krylov space $K_m \equiv K_m(A, v_1) = \{p(A)v_1 \; ; \; p \in \mathcal{P}_{m-1}\}$ where $\mathcal{P}_{m-1}$ is the set of polynomials of degree less or equal to $m-1$ and a Hessenberg matrix $H_m = V_m^* A V_m$ of order m such that

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^*. \tag{2.1}$$

For more details on this algorithm, we refer to [30].

*Received December 13, 1996. Accepted for publication December 15, 1997. Communicated by M. Eiermann.
† University of Heidelberg. IAM. INF 294. 69120 Heidelberg. (vincent.heuveline@iwr.uni-heidelberg.de)
‡UBO. Department of Mathematics. 6, av. Le Gorgeu, 29287 Brest Cédex France. (sadkane@univ-brest.fr)

Let us assume that the right-most eigenvalue λ, we are interested in, is semi-simple and let P denote the corresponding spectral projector. If $Pv_1 \neq 0$, then $u = Pv_1/\|Pv_1\|_2$ is an eigenvector of A associated with λ. Let us define the vector $y = (I-P)v_1/\|(I-P)v_1\|_2$ if $(I-P)v_1 \neq 0$ and $y = 0$ otherwise. Then we have the following proposition

PROPOSITION 2.1. *The angle $\theta(u, K_m)$ between u and the space K_m satisfies*

$$\sin\theta(u, K_m) \leq \min_{\substack{p \in \mathcal{P}_{m-1} \\ p(\lambda)=1}} \|p(A)y\|_2 \tan\theta(u, v_1), \tag{2.2}$$

where $\theta(u, v_1)$ is the angle between u and the starting vector v_1.

Proof. The vector v_1 can be written as $v_1 = \alpha\,(u+\beta y)$, with $\alpha \neq 0$ and $\beta = \tan\theta(u, v_1)$.

Let $q \in \mathcal{P}_{m-1}$ with $q(\lambda) = 1$, then the vector $v := \frac{1}{\alpha} q(A)v_1 \ \in K_m$ and can be written as $v = q(A)(u + \ \beta y) = u + \beta\, q(A)y$. Hence

$$\sin\theta(u, v) = \inf_{\gamma\in\mathbb{C}} \|u - \gamma\, v\|_2 \leq |\beta|\ \|q(A)y\|_2. \tag{2.3}$$

□

We point out that similar result is given in [30, p.204] assuming that the matrix is diagonalizable. Proposition 2.1 essentially means that the space K_m will contain enough information on the eigenvector u (i.e. $\sin\theta(u, K_m) \to 0$) if we can choose a polynomial p such that $p \in \mathcal{P}_{m-1},\ p(\lambda) = 1$ and $\|p(A)y\|$ as small as possible. If A is diagonalizable, this will be possible if the polynomial p realizes the minimax problem

$$\gamma_{m-1}(\Omega) := \min_{\substack{p \in \mathcal{P}_{m-1} \\ p(\lambda)=1}} \max_{z\in\Omega} |p(z)|, \tag{2.4}$$

where $\Omega \subset \mathbb{C}$ is a compact set which contains the spectrum of A except λ.

We will now focus on the solution of the minimax problem (2.4). Let us assume that $\Omega^c := \overline{\mathbb{C}}\backslash\Omega$, the complement of Ω with respect to the extended complex plane $\overline{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ is simply connected. Then the Riemann mapping theorem [22, p. 8] ensures the existence of a function $w = \Phi(z)$ which maps Ω^c conformally onto $D^c := \overline{\mathbb{C}}\backslash D$, the exterior of the unit disk $D := \{w \in \mathbb{C}, |w| \leq 1\}$, and which satisfies the conditions

$$\Phi(\infty) = \infty \ \text{ and } \ 0 < \lim_{z\to\infty} \frac{\Phi(z)}{z} < \infty. \tag{2.5}$$

As a consequence, the Laurent expansion of $\Phi(z)$, at infinity, has the form

$$\Phi(z) = \alpha z + \alpha_0 + \frac{\alpha_1}{z} + \frac{\alpha_2}{z^2} + \cdots \qquad \text{with } \alpha > 0. \tag{2.6}$$

The polynomial part

$$F_k(z) = \alpha_k^{(k)} z^k + \alpha_{k-1}^{(k)} z^{k-1} + \ldots + \alpha_0^{(k)} \tag{2.7}$$

of the Laurent expansion of

$$(\Phi(z))^k = \alpha_k^{(k)} z^k + \alpha_{k-1}^{(k)} z^{k-1} + \ldots + \alpha_0^{(k)} + \frac{\alpha_1^{(k)}}{z} + \ldots \tag{2.8}$$

is called the Faber polynomial of degree k associated with Ω.

In the sequel, the Faber polynomials of degree k generated by Ω will be denoted by $F_{k,\Omega}(z)$ or simply $F_k(z)$ if there are no ambiguities.

Now let $\Psi(w)$ be the inverse of the function $\Phi(z)$ given in (2.5). It can be shown [22, p. 106] that $\Psi(w)$ has a Laurent expansion of the form

$$\Psi(w) = \beta w + \beta_0 + \frac{\beta_1}{w} + \frac{\beta_2}{w^2} + \dots \quad \text{with } \beta = \frac{1}{\alpha}. \tag{2.9}$$

Considering the Laurent expansion (2.9), the Faber polynomials can now directly be computed recursively from

$$F_0(z) = 1, \tag{2.10}$$

$$F_1(z) = (z - \beta_0)/\beta, \tag{2.11}$$

$$\begin{aligned} F_k(z) = (zF_{k-1}(z) - (\beta_0 F_{k-1}(z) + \dots + \\ \beta_{k-1}F_0(z)) - (k-1)\beta_{k-1})/\beta, \quad k \geq 2. \end{aligned} \tag{2.12}$$

2.1. The case where Ω is a disk or an ellipse. Let us consider the particular case where Ω is the closed disk $\mathcal{D}(c,\rho) = \{z \in \mathbb{C} : |z - c| \leq \rho\}$ of center c and radius ρ. Then the map

$$\Psi(w) = \rho w + c, \quad |w| > 1, \tag{2.13}$$

transforms D^c onto the exterior of $\mathcal{D}(c,\rho)$. Comparing (2.13) with (2.9) and (2.10)-(2.12), the Faber polynomials can be written in this case as:

$$F_k(z) = \left(\frac{z-c}{\rho}\right)^k, \quad k \geq 0. \tag{2.14}$$

Now, assume that the domain Ω is the closed interior of the ellipse $\mathcal{E}(c,e,a)$ of center c, foci $c+e,\ c-e$, major semi-axis a and minor semi-axis b. Then the map Ψ is the Joukowski transformation [21, p. 197] from D^c onto the the exterior of $\mathcal{E}(c,e,a)$ given by

$$\psi(w) = \gamma_1 w + c + \frac{\gamma_{-1}}{w}, \quad |w| > 1, \tag{2.15}$$

with $\gamma_1 = \frac{e(|a|+|b|)}{2|e|}$ and $\gamma_{-1} = \frac{e|e|}{2(|a|+|b|)}$. Comparing (2.15) with (2.9) and (2.10)-(2.12), the Faber polynomials can be written in this case

$$F_0(z) = 1, \tag{2.16}$$

$$F_1(z) = (z-c)/\gamma_1, \tag{2.17}$$

$$F_2(z) = \frac{(z-c)^2}{\gamma_1^2} - 2\frac{\gamma_{-1}}{\gamma_1} \tag{2.18}$$

$$F_k(z) = [(z-c)F_{k-1}(z) - \gamma_{-1}F_{k-2}(z)]/\gamma_1, \quad k \geq 2. \tag{2.19}$$

Thus, if we define the polynomials T_k by

$$T_0(z) = 1, \tag{2.20}$$

$$T_k(z) = \frac{1}{2}\left(\frac{|a|+|b|}{|e|}\right)^k F_k(ez+c), \quad k \geq 1, \tag{2.21}$$

we easily see that the polynomials T_k satisfy the three term recurrence of Chebyshev polynomials

$$T_k(z) = 2zT_{k-1}(z) - T_{k-2}(z), \quad k \geq 2, \tag{2.22}$$

with $T_0(z) = 1$ and $T_1(z) = z$.

It is known that when Ω is a circle [25], the normalized polynomial $\frac{F_{m-1}(z)}{F_{m-1}(\lambda)}$ where $F_m(z)$ is given in (2.14), solves the minimax problem (2.4). When Ω is an ellipse [9], the Chebyshev polynomial sequence given in (2.22), is asymptotically optimal with respect to the problem (2.4). That is, for large m the normalized Chebyshev polynomial $\frac{T_{m-1}(\frac{z-c}{e})}{T_{m-1}(\frac{\lambda-c}{e})}$ is closed to the optimal polynomial (see also the discussion in §4).

2.2. The case where Ω is a polygon. For our purpose, the hope is that Faber polynomials have analogous properties for more general domain. The following result, due to Starke and Varga [33], shows that this is indeed the case at least when Ω is convex.

THEOREM 2.2. *Assume that Ω is convex with $\lambda \notin \Omega$. Then the normalized Faber polynomials $\tilde{F}_k(z) := F_k(z)/F_k(\lambda)$ satisfy,*

$$\gamma_k(\Omega) \leq \max_{z\in\Omega} |\tilde{F}_k(z)| < \frac{2}{|\Phi(\lambda)|^k - 1} \leq \frac{2}{1 - \frac{1}{|\Phi(\lambda)|^k}} \gamma_k(\Omega) \quad (k \geq 1). \tag{2.23}$$

Since $|\Phi(\lambda)| > 1$, theorem 2.2 shows that the normalized Faber polynomials F_k lie asymptotically between $\gamma_k(\Omega)$ and $2\ \gamma_k(\Omega)$, and this explains their near optimality. In section 4 we give a numerical illustration of theorem (2.2).

For at least sufficiently large k, the constraint on the convexity of Ω can be relaxed by assuming that Ω is only of bounded boundary rotation [33].

In the context of linear systems, the condition $\lambda \notin \Omega$ is replaced by the condition $0 \notin \Omega$ where Ω is a domain that contains the spectrum of A. It is known that the convexity assumption on Ω and the condition $0 \notin \Omega$ are difficult to reconcile. The situation appears to be favorable in the context of the eigenvalue problem that interests us because the set Ω is a polygon constructed as the convex hull of the spectrum of the matrix H_m and excluding the right-most value λ, so that the condition $\lambda \notin \Omega$ cannot prevent the set Ω from being convex.

Besides the importance of theorem (2.2), there are at least three reasons to believe that this approach is acceptable. First, as we have already mentioned, the polygon, contrary to the ellipse, can better fit the shape of the distribution of the unwanted eigenvalues. Second, the polygon will be constructed from the outermost eigenvalues of H_m and it is known that this part of the spectrum is what Arnoldi's method approximates first [30]. Third, if a Faber polynomial p solves the problem (2.4) in a domain Ω, then p will also be small on neighborhoods of the boundary of Ω [33], and this gives some freedom in the construction of the polygon as will be discussed in section 3.

3. Computation of Faber polynomials for polygons by the Schwarz-Christoffel transformation.

3.1. The Schwarz-Christoffel transformation. We assume from now on that Ω is a polygon with vertices $z_1,\ z_2, \ldots, z_p$ given in counter-clockwise order and interior angles $\alpha_1\pi,\ \alpha_2\pi, \ldots, \alpha_p\pi$ with $0 < \alpha_j < 2,\ j = 1, \ldots, p$ and $\sum_{j=1}^{p} \alpha_j = p - 2$.

The Schwarz-Christoffel transformation [22, p.331] allows us to express the conformal map Ψ through a formula that involves parameters depending only on the geometry of Ω. The Faber polynomials are then computed using these parameters.

THEOREM 3.1. *Let* $a_1 = e^{i\theta_1}, \dots, a_p = e^{i\theta_p}$*, with* $0 \le \theta_1 < \dots < \theta_p < 2\pi$*. The conformal map* Ψ_1*, mapping* D *onto* Ω^c*, and satisfying* $\Psi_1(a_k) = z_k,\ \ k = 1, \dots, p$ *is given by*

$$\Psi_1(w) = C_{\Psi_1} \int_{\tilde{w}}^{w} \prod_{j=1}^{p} (t - a_j)^{\alpha_j - 1} \frac{dt}{t^2} + \Psi_1(\tilde{w}),\ \ w \in D, \tag{3.1}$$

where $w = 0$ *is the inverse of the point at infinity,* $\tilde{w} \in D$ *and* $C_{\Psi_1} \in \mathbb{C}$*.*

Proof. See [22, p. 331]. □

As a consequence, the conformal map Ψ, mapping D^c onto Ω^c, and satisfying $\Psi(\bar{a}_k) = z_k,\ k = 1, \dots, p$ is given by

$$\Psi(w) = C_{\Psi} \int_{w_0}^{w} \prod_{j=1}^{p} (1 - \frac{\bar{a}_j}{t})^{\alpha_j - 1} dt + \Psi(w_0),\ \ w \in D^c, \tag{3.2}$$

with $w_0 = \frac{1}{\tilde{w}}$ and $C_{\psi} = -C_{\Psi_1} \prod_{j=1}^{p} a_j^{\alpha_j - 1}$, where $\tilde{w}$ and C_{Ψ_1} are defined in theorem 3.1.

3.2. Parameter determination in the Schwarz-Christoffel transformation. For arbitrary values of the parameters C_{Ψ},$a_1, \cdots, a_p$, known as the accessory parameters, the representation (3.2) of Ψ, guarantees that the image of the unit circle respects the interior angles of the polygon Ω in the sense that the interior angles corresponding to the vertices $z_j = \Psi(\bar{a}_j),\ j = 1, \dots, p$, are precisely given by $\alpha_j,\ j = 1, \dots, p$. However, Ψ will in general fail to reproduce correctly the side lengths of Ω. Consequently, for a given polygon Ω, we need to determine the accessory parameters C_{Ψ},$a_1, \cdots, a_p$, in order to obtain an explicit expression of Ψ.

This problem, called the accessory parameter problem, and its numerical treatment, have been studied by several authors [34, 15, 10, 3, 4]. In our work, we have adopted the approach due to Trefethen [34]. Trefethen's package SCPACK [35] implements in Fortran 77 the Schwarz-Christoffel transformations from $\mathcal{D}$ onto the interior of Ω, and needs therefore to be modified for our purpose. The Matlab toolbox for the Schwarz-Christoffel mapping developed by Driscoll [4] solves the accessory parameter problem for various Schwarz-Christoffel transformations, among which, the conformal map Ψ_1 used in (3.1) that carries $\mathcal{D}$ into Ω^c. To achieve good performances, we have adapted this Matlab implementation to Fortran 77, using the backbone routines of SCPACK.

In what follows, we briefly describe the implemented scheme and some necessary choices related to its implementation.

We would like to find $a_1 = e^{i\theta_1}, \dots, a_p = e^{i\theta_p}$ such that

$$\Psi(\bar{a}_k) \equiv \Psi_1(a_k) = z_k, \quad k = 1, \dots, p. \tag{3.3}$$

Since Ψ_1 is defined such that the origin $w = 0$ is mapped onto the point at infinity, an arbitrary choice of one of the $a_k's$ determines uniquely the map Ψ_1. We impose $a_p = 1$. The accessory parameters to be determined are therefore $a_1, \cdots, a_{p-1}$ and $C_{\Psi_1} \equiv -C_{\Psi} / \prod_{j=1}^{p-1} a_j^{\alpha_j - 1}$. This amounts to determining $p + 1$ real variables.

As explained in [34] and implemented in [4], by considering the side lengths of Ω, we obtain the $p - 3$ real conditions

$$\frac{|\Psi_1(a_{k+1}) - \Psi_1(a_k)|}{|\Psi_1(a_2) - \Psi_1(a_1)|} = \frac{|\int_{a_k}^{a_{k+1}} \prod_{j=1}^{p} (t - a_j)^{\alpha_j - 1} \frac{dt}{t^2}|}{|\int_{a_1}^{a_2} \prod_{j=1}^{p} (t - a_j)^{\alpha_j - 1} \frac{dt}{t^2}|} = \frac{|z_{k+1} - z_k|}{|z_2 - z_1|} \quad k = 2, \cdots, p - 2. \tag{3.4}$$

If Res (Ψ', ∞) denotes the residue at infinity of Ψ', from (2.9) we have

$$\mathrm{Res}(\Psi', \infty) = 0, \tag{3.5}$$

or equivalently

$$\mathrm{Res}(\Psi'_1, 0) = 0. \tag{3.6}$$

Note that 0 is the unique pole of Ψ'_1 in D. Now if we insist that the integral in (3.1) is path-independent, then form (3.6) and the residue theorem applied to (3.1) we obtain the two real conditions

$$\sum_{j=1}^{p} \frac{\alpha_j - 1}{a_j} = 0. \tag{3.7}$$

The accessory parameters $a_1, \cdots, a_{p-1}$ are therefore determined by solving the following non-linear system of p-1 real equations

$$\text{(S-C)} \quad \begin{cases} \dfrac{\left| \int_{a_k}^{a_{k+1}} \prod\limits_{j=1}^{p} (t-a_j)^{\alpha_j - 1} \frac{dt}{t^2} \right|}{\left| \int_{a_1}^{a_2} \prod\limits_{j=1}^{p} (t-a_j)^{\alpha_j - 1} \frac{dt}{t^2} \right|} = \frac{|z_{k+1} - z_k|}{|z_2 - z_1|} \quad k = 2, \cdots, p-2, \\ \sum_{j=1}^{p} \frac{\alpha_j - 1}{a_j} = 0. \end{cases}$$

The parameter C_{Ψ_1} is then determined using for example the expression

$$C_{\Psi_1} = \frac{z_2 - z_1}{\int_{a_1}^{a_2} \prod\limits_{j=1}^{p} (t - a_j)^{\alpha_j - 1} \frac{dt}{t^2}}. \tag{3.8}$$

As in SCPACK [35], we use the Powell library routine NS01A [27] for solving the unconstrained non-linear system (S-C). This routine makes use of a steepest descent search in early iterations if necessary, followed by a variant of the Newton method.

The evaluation of the Schwarz-Christoffel integrals in the non-linear system (S-C) is the most consuming computational task. The first decision concerns the path of integration. Because of the path-independency, the simplest path of integration seems to be the straight line segments of the form $a_k\ a_{k+1}$. However the neighborhood of the pole $w = 0$ will be avoided by splitting these segments into two segments with endpoints on the unit circle.

We omit some important details discussed in [34] on the use of the compound Gauss-Jacobi quadrature for computing the integrals in (S-C) and on the change of variable

$$y_j = \log \frac{\theta_j - \theta_{j-1}}{\theta_{j+1} - \theta j} \quad 1 \leq j < p-1, \text{ with } \theta_0 = 0,\ \theta_p = p, \tag{3.9}$$

that eliminates the constraints (see theorem 3.1)

$$0 < \theta_j < \theta_{j+1} \text{ for } j = 1, \cdots, p-1, \tag{3.10}$$

in the non-linear system (S-C).

In our application, it is not possible to know in advance the shape and localization of the considered polygon since this method is intended to be used in the context of iterative methods. This may lead to the most troublesome aspect of numerical Schwarz-Christoffel mapping, the so-called *crowding phenomenon* [15, p.428]. An almost uniform distribution for the vertices of the polygon may indeed yield a highly nonuniform distribution for the pre-vertices, such that, some of them may become indistinguishable in finite precision arithmetic. Typically, this phenomenon appears for domains that are elongated in one or more directions and/or have small-angled corners [26, 24, 18].

To partly circumvent this difficulty we impose for each polygon a filtering process whose main objective is to suppress clustered vertices. Let z_k and z_{k+1} be two vertices of the considered polygon. If the ratio of the distance between this two vertices through the length of the longest side of the polygon is lower than $r_{min} = 5.10^{-2}$, then both vertices are suppressed and replaced by $\frac{1}{2}(z_k + z_{k+1})$, and this process is repeated until all the ratios exceed r_{min}. The resulting polygon is taken to be the convex hull of the remaining vertices.

This process has the advantage of reducing the number p of vertices and therefore the CPU time necessary to solve the parameter problem which is estimated in [34] to be of order $O(p^3)$. Moreover it is theoretically validated in that the polynomials $\tilde{F}_k$ defined in theorem 2.2 are also nearly optimal in the neighborhood of the polygon Ω.

3.3. Construction of Faber polynomials. We assume in this subsection that the accessory parameters C_Ψ,a_1,$a_2, \cdots, a_p$ are known and we address the problem of computing the Faber polynomial F_k given in the recurrence (2.10)-(2.12). We clearly need the coefficients β,$\beta_0, \cdots, \beta_{k-1}$ of the Laurent expansion (2.9) of Ψ.

As in the approach adopted in [33], using the derivative of the two expressions of $\Psi(w)$ given in (2.9) and (3.2), we get

$$\beta - \sum_{l=2}^{\infty} \frac{(l-1)\beta_{l-1}}{w^l} = C_\psi \left[\prod_{j=1}^{p} \left(\sum_{l=0}^{\infty} \gamma_l^j \left(\frac{1}{w}\right)^l \right) \right]. \tag{3.11}$$

Comparing corresponding powers of w in (3.11), we obtain β,$\beta_1, \cdots, \beta_{k-1}$ as

$$\begin{bmatrix} \beta \\ 0 \\ -\beta_1 \\ -2\beta_2 \\ \vdots \\ -(k-1)\beta_{k-1} \end{bmatrix} = C_\Psi \prod_{j=1}^{p} \begin{bmatrix} 1 & & & & & \\ \gamma_1^j & \ddots & & & & \\ \gamma_2^j & \gamma_1^j & \ddots & & & \\ \vdots & \ddots & \ddots & \ddots & & \\ \vdots & \cdots & & & \ddots & \\ \gamma_k^j & & & \gamma_2^j & \gamma_1^j & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}, \tag{3.12}$$

where

$$\gamma_l^j = \begin{pmatrix} \alpha_j - 1 \\ l \end{pmatrix} \left(\frac{-1}{a_j}\right)^l. \tag{3.13}$$

Now from the equalities $z_j = \Psi(\bar{a}_j),\ j = 1, \ldots, p$, we obtain the approximations

$$z_j \approx \beta\bar{a}_j + \beta_0 + \sum_{l=1}^{k-1} \frac{\beta_l}{\bar{a}_j^l} \quad \text{for } j = 1, \ldots, p.$$

Hence an approximation of β_0:

$$\beta_0 \approx \frac{1}{p} \sum_{j=1}^{p} \left[z_j - \left(\beta\bar{a}_j + \sum_{l=1}^{k-1} \frac{\beta_l}{\bar{a}_j^l} \right) \right]. \tag{3.14}$$

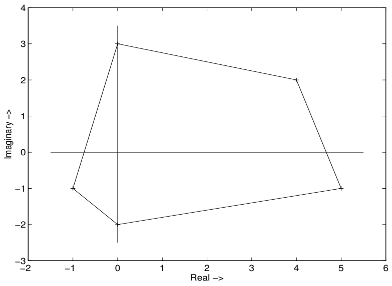

FIG. 4.1. *Polygon* Ω

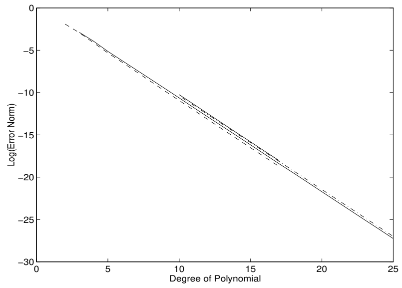

FIG. 4.2. *Faber Error norm* - $\lambda = 10$

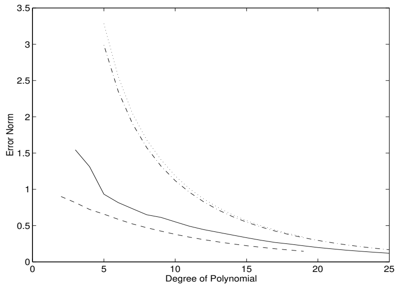

FIG. 4.3. *Faber Error norm* - $\lambda = 5$

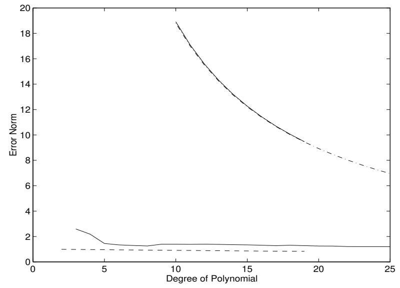

FIG. 4.4. *Faber Error norm* - $\lambda = 4.7$

In the Figures 4.2, 4.3 and 4.4, $\gamma_k(\Omega)$, $\max_{z\in\Omega} |\tilde{F}_k(z)|$, $\frac{2}{|\phi(\lambda)|^k-1}$ and $\frac{2}{1-\frac{1}{|\phi(\lambda)|^k}}$ are respectively represented with dashed lines, solid lines, dash-dot lines and dotted lines.

4. Numerical results. The numerical experiments have been carried out on a SUN UltraSPARC workstation with IEEE double precision. The first point we would like to illustrate is the numerical behavior of the inequalities (2.23) given in theorem 2.2, for different values of the degree k of the polynomial $\tilde{F}_k$ and for different normalization points λ. We assume that Ω is the convex polygon (see figure 4.1) consisting of the vertices: $(0,-2)$,$(-1,-1)$,$(0,3)$,$(4,2)$ and $(5,-1)$. We consider three situations corresponding to $\lambda = 10$ in figure 4.2, $\lambda = 5$ in figure 4.3 and $\lambda = 4.7$ in figure 4.4.

The computations of $\gamma_k(\Omega)$ is carried out with the Matlab package COCA.2 (COmplex linear Chebyshev Approximation) developed by Fisher and Modersitzki [8]. The implemented scheme is based on a reformulation of the minimax problem (2.4) as a semi-infinite optimization problem [11, 13]. The discretization of the boundary of Ω and the connection between the primal and the dual reformulation of the problem lead to a low dimensional optimization problem whose size depends only on the degree of the searched polynomial. Moreover, upper and lower bounds for $\gamma_k(\Omega)$ are available at any step of the algorithm. In order to attain the precision required for our comparisons, we set the number of discretization points of the boundary to $n_c = 10000$ and impose that the difference between the upper and the lower bounds of $\gamma_k(\Omega)$ must be less than $\epsilon_c = 10^{-10}$. We consider polynomials of degree less or equal to $k_{max} = 19$, since for polynomial degrees exceeding this value, we encounter some numerical instabilities due to near singular systems.

The Faber polynomials $F_k(z)$ and $\Phi(\lambda)$ are computed using the package SC-TOOLBOX 1.3 [4]. Similarly to COCA.2, we change the default tolerance to $\epsilon_{sc} = 10^{-14}$ in our computation. For the computation of $\max_{z\in\Omega} |\tilde{F}_k(z)|$, we discretize the boundary of Ω with $n_\Omega = 100000$ points.

The curves in figures 4.2, 4.3 and 4.4 show the behavior of the quantities involved in the inequalities (2.23). i.e., $\gamma_k(\Omega)$, $\max_{z\in\Omega}|\tilde{F}_k(z)|$, $\frac{2}{|\phi(\lambda)|^k-1}$ and $\frac{2}{1-\frac{1}{|\phi(\lambda)|^k}}$. The curves clearly show the expected fact that the upper bounds in (2.23) give better approximation of the error norm when the degree k of the approximant increases. Furthermore these upper bounds greatly depend on the location of the normalization point λ which can easily be seen considering the denominator of both last parts of the inequalities (2.23).

We notice from figure 4.4 the bad approximation of $\max_{z\in\Omega}|\tilde{F}_k(z)|$ to $\gamma_k(\Omega)$, due to the closeness of λ to the boundary of Ω, and that the error appears to tends towards 0 for large degree k as predicted by the theory.

An analogy with the work by Fischer and Freund [9] can be made at this point. The authors of [9] have proved that the result due to Clayton [2] concerning the optimality of Chebyshev polynomials with respect to the ellipse, is not true in general. They showed the non optimality by considering normalization points near enough to the considered ellipse [9, Th.1(b)].

Now, we would like to show the benefit that can be obtained from the Faber polynomial acceleration when combined with Arnoldi's method for computing eigenvalues. But first let us mention some important points in our algorithm. The Arnoldi algorithm that we have implemented is a block version of Arnoldi, implemented in the same spirit as in [31]. We use an implicit deflation technique which may briefly summarized as follows: every time an eigenvector converges, it is put, after orthonormalization, at the beginning of the basis V_k so that all subsequent constructed vectors are orthogonalized against it, the algorithm continues then with a reduced block size. In the remaining of this section we call BAD this deflated version of the block Arnoldi method. As we have already mentioned, the unconstrained nonlinear system (S-C) is solved by using the Powell library routine NS01A [27].

A computed eigenpair (μ, x) is considered as convergent if the associated residual $Ax - \mu x$ satisfies the condition

$$\|Ax - \mu x\|_2 \leq tol\ \|A\|_F, \tag{4.1}$$

where $\|A\|_F$ is the Frobenius norm of A and tol is a tolerance parameter indicated for each test example. We also indicate, for each test example, the number $mmax$ which is the total number of vectors built at each iteration of BAD, the block size nb, the number of eigenvalues $nval$ and the parameter $maxit$ which is an upper bound on the number of iterations (i.e: the number of restarts). The algorithm terminates when $maxit$ is exceeded.

Our aim is to compare our deflated block-Arnoldi Faber algorithm with the deflated block-Arnoldi Chebyshev version that has been developed in [17]. In what follows, the notations BADC and BADF respectively stand for the deflated block Arnoldi Chebyshev and the deflated block Arnoldi Faber methods. The notation BADF20, for example, means that the method BADF is used with a Faber polynomial of degree 20.

For each test problems we have furthermore considered the computation of the solution using the package ARPACK [20] which is available in Netlib. This package implements the implicitly restarted Arnoldi method of Sorensen [32]. Contrary to our approach, which uses an explicit restart, the degree of the considered acceleration polynomial, in ARPACK, depends directly on the size of the Krylov subspace. This difference as well as the difference in the deflation strategies, used in ARPACK and in our approach make the comparisons, under the same memory and/or CPU requirements, very difficult and sometimes unfair. Nevertheless, we present some results concerning ARPACK for illustration purposes, not for a true benchmarking.

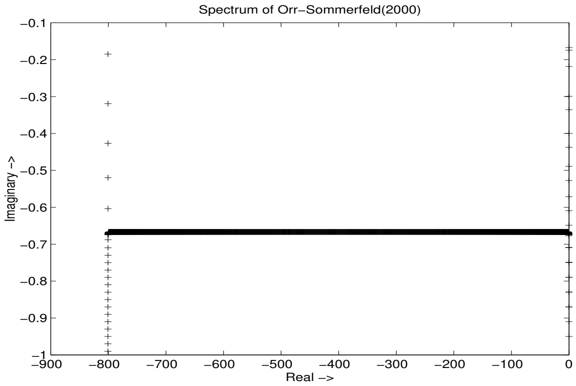

FIG. 4.5. *Spectrum of the Orr-Sommerfeld matrix*

We consider the Orr-Sommerfeld operator [19] defined by

$$\frac{1}{\alpha R}L^2 y - i(ULy - U^{''}y) - \lambda Ly = 0, \tag{4.2}$$

where α and R are positive parameters, λ is a spectral parameter number, $U = 1 - x^2$, y is a function defined on $[-1, +1]$ with $y(\pm 1) = y^{'}(\pm 1) = 0$ and $L = \frac{d^2}{dx^2} - \alpha^2$.

Discretizing this operator using the following simple approximation

$$x_i = -1 + ih, \; h = \frac{2}{n+1},$$
$$L_h = \frac{1}{h^2} \text{ Tridiag } (1, -2 - \alpha^2 h^2, 1),$$
$$U_h = \text{diag } (1 - x_1^2, \ldots, 1 - x_n^2),$$

gives rise to the eigenvalue problem

$$Au = \lambda u \text{ with } A = \frac{1}{\alpha R}L_h - iL_h^{-1}(U_h L_h + 2I_n). \tag{4.3}$$

Taking $\alpha = 1, \; R = 5000, \; n = 2000$ yields a complex non hermitian matrix A (order $n = 2000$, $\|A\|_F = 21929$) containing $4.00e+06$ nonzero elements. Its spectrum is plotted in Figure 4.5.

We compute the four rightmost eigenpairs of the Orr-Sommerfeld matrix using the different methods BAD, BADC and BADF. Table 4.1 shows the results obtained for several values of the degree of the Faber and Chebyshev polynomials. Clearly the Faber polynomials considerably improve the results of BADC.

It is important to notice that for $mmax = 60$ neither BAD nor BADC20 achieve the convergence of the four eigenvalues while the convergence was obtained within 130 iterations with BADF20. The best result, regarding the CPU time, was obtained with BADF20 and $mmax = 80$. Figure 4.6 (left) shows its convergence behavior. For the sake of clarity, we show in Figure 4.6 (right) a comparison of the convergence behavior of the third and fourth eigenvalues for BADF20 and BADC20 with $m_{max} = 80$. The convergence using ARPACK occurs in 89 iterations (824 sec.) for $m_{max} = 80$ and was achieved for 2 eigenvalues in 85 iterations (610 sec.) with $m_{max} = 60$.

Method	$mmax$	Matrix-vector multiplications	Iterations	Time(sec)	# of converged eigenpairs
BAD	60	12000	200	581.2	0
BADF20	60	13904	130	644.9	4
BADC20	60	27656	200	1001.4	1
BAD	80	15978	200	971.3	1
BADF15	80	11599	99	627.2	4
BADC15	80	26361	195	1270.3	4
BADF20	80	9592	65	445.0	4
BADC20	80	28798	187	1314.6	4
BADF25	80	15490	104	728.3	4
BADC25	80	27038	175	1225.4	4

TABLE 4.1
Computation of the four rightmost eigenvalues of the Orr-Sommerfeld matrix with different methods. $nb = nval = 4,\ itmax = 200,\ tol = 1.00e-7$

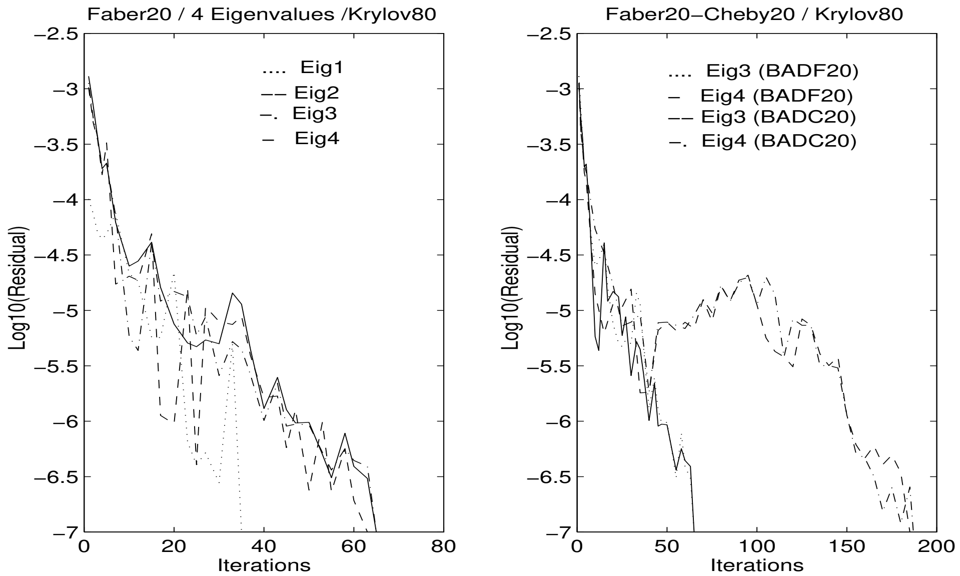

FIG. 4.6. *Convergence behavior of the four rightmost eigenpairs with BADF20 (left) and comparison of convergence behavior for the third and fourth eigenpairs with BADF20 and BADC20 (right). (Orr-Sommerfeld matrix)*

The second test matrix is the matrix Young1c taken from the Harwell-Boeing collection of test matrices [5]. This matrix arises when modeling the acoustic scattering phenomenon. It is complex (order $n = 841$, $\|A\|_F = 6448$) and contains 4089 nonzero elements. Its spectrum is plotted in Figure 4.7.

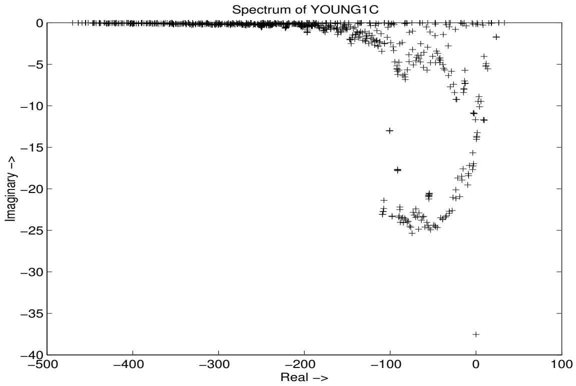

FIG. 4.7. *Spectrum of the matrix Young1c*

We compute, in Table 4.2, the eleventh rightmost eigenpairs of the matrix Young1c using the different methods BAD, BADC and BADF. Similarly to the Orr-Sommerfeld case, the method BAD has difficulties for computing all the wanted eigenpairs. The results obtained with the method BADF are always better than those with BADC. Again the best result is obtained with BADF20. Figure 4.8 (left) shows the convergence behavior of the two first and the two last computed rightmost eigenpairs with the method BADF20. Similarly to the previous examples, the plot, in figure 4.8 (right) shows a comparison of the convergence behavior of the tenth and eleventh eigenpairs with the methods BADF20 and BADC20. ARPACK computed the sought eigenpairs within 18 iterations (15 sec.).

Method	Matrix-vector multiplications	Iterations of	Time(sec)	# of converged eigenpairs
BAD	3488	100	56.2	10
BADF15	1914	16	15.2	11
BADC15	2551	20	26.6	11
BADF20	1926	12	13.4	11
BADC20	3893	22	38.4	11
BADF25	2333	13	18.2	11
BADC25	5104	22	47.8	11

TABLE 4.2
Computation of the twelve rightmost eigenvalues of the matrix Young1c with different methods. $mmax = 36$, $nb = 12$, $nval = 11$, $itmax = 100$, $tol = 1.00e-10$

Finally let us give an idea of the percentage of the overall execution time spent in the main steps of our block Arnoldi Faber method. The version BADF20 gave the best results and will therefore be used for this task. The main steps in BADF20 can be summarized as follows:

1. the matrix vector multiplication involved in Arnoldi's iteration,
2. the orthogonalization among the constructed Arnoldi's vectors,
3. the Schwarz-Christoffel transformation, i.e.: construction of the polygon,
4. the Faber polynomial acceleration. i.e. : matrix vector multiplication involved at each restart.

In Table 4.3, we give the time and the percentage of time spent in each of the above steps. The dominating parts are, as usual, the matrix-vector multiplications and the orthogonalization

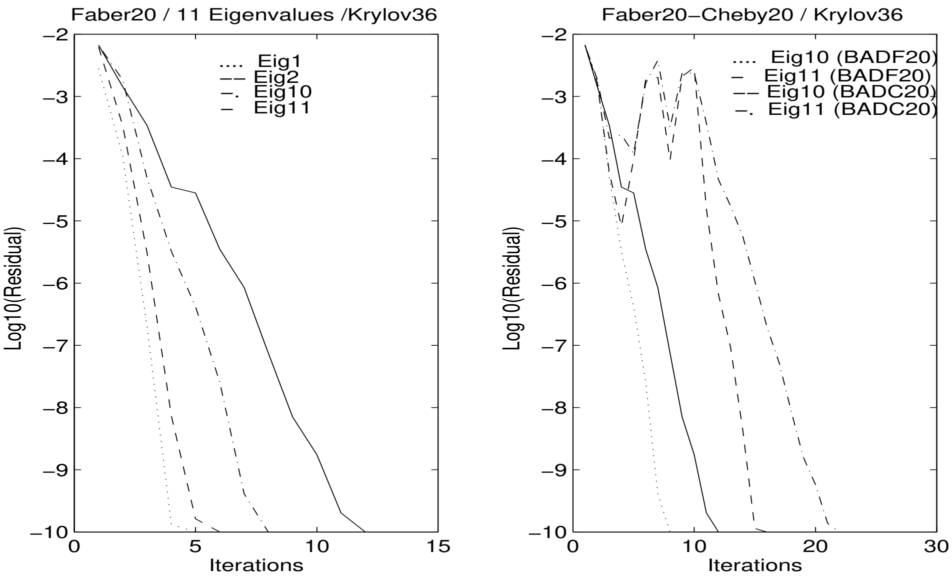

FIG. 4.8. *Convergence behavior of the first, second, tenth and eleventh right-most eigenpairs with BADF20 (left) and comparison of convergence behavior for the tenth and eleventh eigenpairs with BADF20 and BADC20 (right). (Young1c matrix)*

process. The time required for the Schwarz-Christoffel transformation is negligible for the matrix Young1c and equivalent to the time required for the Faber acceleration for the matrix Orr-Sommerfeld.

step	Orr-Sommerfeld		Young1c	
Matrix-vector	138.9	31.2 %	1.3	9.6 %
Orthogonalization	166.0	37.3 %	3.3	24.5 %
Schwarz-Christoffel	65.0	14.6 %	1.0	7.5 %
Faber acceleration	66.2	14.9 %	6.1	45.7 %

TABLE 4.3
Time(sec) and percentage of time for the main steps in algorithm BADF20

5. Conclusion. We have proposed a Faber polynomial acceleration for computing eigenvalues of large non hermitian matrices. The nice properties of this method is that the convex polygon Ω constructed from the outermost unwanted eigenvalues approximated by Arnoldi's method is always feasible, and that the associated Faber polynomial is asymptotically optimal even in the neighborhood of the the boundary of Ω. The numerical tests show that the method performs well and overcomes the acceleration based on Chebyshev polynomials.

Acknowledgments. Part of this work was done while the authors were at INRIA-IRISA, Rennes, France.

REFERENCES

[1] W. E. ARNOLDI, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17-29.
[2] A. J. CLAYTON, *Further results on polynomials having least maximum modulus over an ellipse in the complex plane*, UKAEA Memorandum, AEEW, 1963.
[3] T. K. DELILLO AND A. R. ELCRAT, *Numerical conformal mapping methods for exterior regions with corners*, J. Comput. Phys., 108 (1993), pp. 199–208.
[4] T. A. DRISCOLL, *A Matlab toolbox for Schwarz-Christoffel mapping*, ACM Trans. Math. Software, 22 (1996), pp. 168-186.
[5] I. S. DUFF, R. G. GRIMES, AND J. G. LEWIS, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (1989), pp. 1–14.
[6] M. EIERMANN, *On semiiterative methods generated by Faber polynomials*, Numer. Math., 56 (1989), pp. 139-156.
[7] G. FABER, *Über polynomische Entwicklungen*, Math. Ann., 57 (1903), pp. 389–408.
[8] B. FISCHER AND J. MODERSITZKI, *An algorithm for complex linear approximation based on semi-infinite programming*, in Algorithms for Approximation 5, M.G. Cox et al., eds, J. C. Baltzer AG, 1993, pp. 287–297.
[9] B. FISCHER AND R. W. FREUND, *Chebyshev polynomials are not always optimal*, J. Approx. Theory, 65 (1991), pp. 261–272.
[10] J. M. FLORYAN AND C. ZEMACH, *Schwarz-Christoffel mappings: A general approach*, J. Comput. Phys., 72 (1987), pp. 347–371.
[11] K. GLASHOFF AND K. ROLEFF, *A new method for Chebyshev approximation of complex-valued functions*, Math. Comp., 36 (1981), pp. 233–239.
[12] G. H. GOLUB AND C. F. VAN LOAN, *Matrix computations, 2nd ed.*, The Johns Hopkins University Press, Baltimore, 1989.
[13] U. GROTHKOPF AND G. OPFER, *Complex Chebyshev polynomials on circular sectors with degree six or less*, Math. of Comp., 39 (1982), pp. 599–615.
[14] D. HO, F. CHATELIN, AND M. BENNANI, *Arnoldi-Tchebychev procedure for large scale nonsymmetric matrices*, RAIRO Modél. Math. Anal. Numér, 24 (1990), pp. 53–65.
[15] P. HENRICI, *Applied and computational complex analysis III*, Wiley, New York London Sydney Toronto, 1986.
[16] D. HO, *Tchebychev acceleration technique for large scale nonsymmetric matrices*, Numer. Math., 86 (1990), pp. 721–734.
[17] V. HEUVELINE AND M. SADKANE, *Chebyshev acceleration techniques for large complex non hermitian eigenvalue problems*, Reliab. Comput., 2 (1996), pp. 111-118.
[18] L. H. HOWELL AND L. N. TREFETHEN, *A modified Schwarz-Christoffel transformation for elongated regions*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 928–949.
[19] W. KERNER, *Large-scale complex eigenvalue problems*, J. Comput. Phys., 85 (1989), pp. 1–85.
[20] R. B. LEHOUCQ, D. C. SORENSEN AND C. YANG, *ARPACK Users Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods*, 1996. Available from netlib@ornl.gov under the directory scalapack.
[21] A. I. MARKUSHEVICH, *Theory of functions of a complex variable I* Prentice-Hall, 1965.
[22] A. I. MARKUSHEVICH, *Theory of functions of a complex variable III*, Prentice-Hall, 1965.
[23] K. MEERBERGEN AND D. ROOSE, *Matrix transformations for computing rightmost eigenvalues of large nonsymmetric matrices*, IMA J. Numer. Anal., 16 (1996), pp. 297–346.
[24] S. O'DONNELL AND V. ROHKLIN, *A fast algorithm for the numerical evaluation of conformal mapping*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 475–487.
[25] G. OPFER AND G. SCHOBER, *Richardson's iteration for nonsymmetric matrices*, Linear Algebra Appl., 58 (1984), pp. 343-361.
[26] N. Papamichael, C. Kokkinos, and M. Warby. *Numerical techniques for conformal mapping onto a rectangle*, J. Comput. Appl. Math., 20 (1987), pp. 349–358.
[27] M. J. D. POWELL, *A Fortran subroutine for solving systems of non-linear algebraic equations*, AERE-R 5947, Harwell, England, 1968.
[28] Y. SAAD, *Projection method for solving large sparse eigenvalue problems*, in Matrix Pencils, Lect. Notes Math., B. Kågström and A. Ruhe, eds., Springer-Verlag, 973 (1982), pp. 121–144.
[29] Y. SAAD, *Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems*, Math. Comp., 42 (1984), pp. 567-588.
[30] Y. SAAD, *Numerical methods for large eigenvalue problems*, Algorithms and Architectures for Advanced Scientific Computing. Manchester University Press, Manchester, UK, 1992.
[31] M. SADKANE, *A block Arnoldi-Chebyshev method for computing the leading eigenpairs of large sparse unsymmetric matrices*, Numer. Math., 64 (1993), pp. 181–193.
[32] D.C. SORENSEN, *Implicit application of polynomial filters in a k-step Arnoldi method* , SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357-385.

[33] G. STARKE, R. S. VARGA, *A hybrid Arnoldi-Faber method for nonsymmetric systems of linear equations*, Numer. Math., 64 (1993), pp. 213-240.

[34] L. N. TREFETHEN, *Numerical computation of the Schwarz-Christoffel transformation*, SIAM J. Sci. Stat. Comput., 1 (1980), pp. 82-101.

[35] L. N. TREFETHEN, *SCPACK user's guide*, Technical Report 89-2, MIT Numerical Analysis Report, 1989.