Karlsruhe Institute of Technology

wave phenomena
analysis and numerics

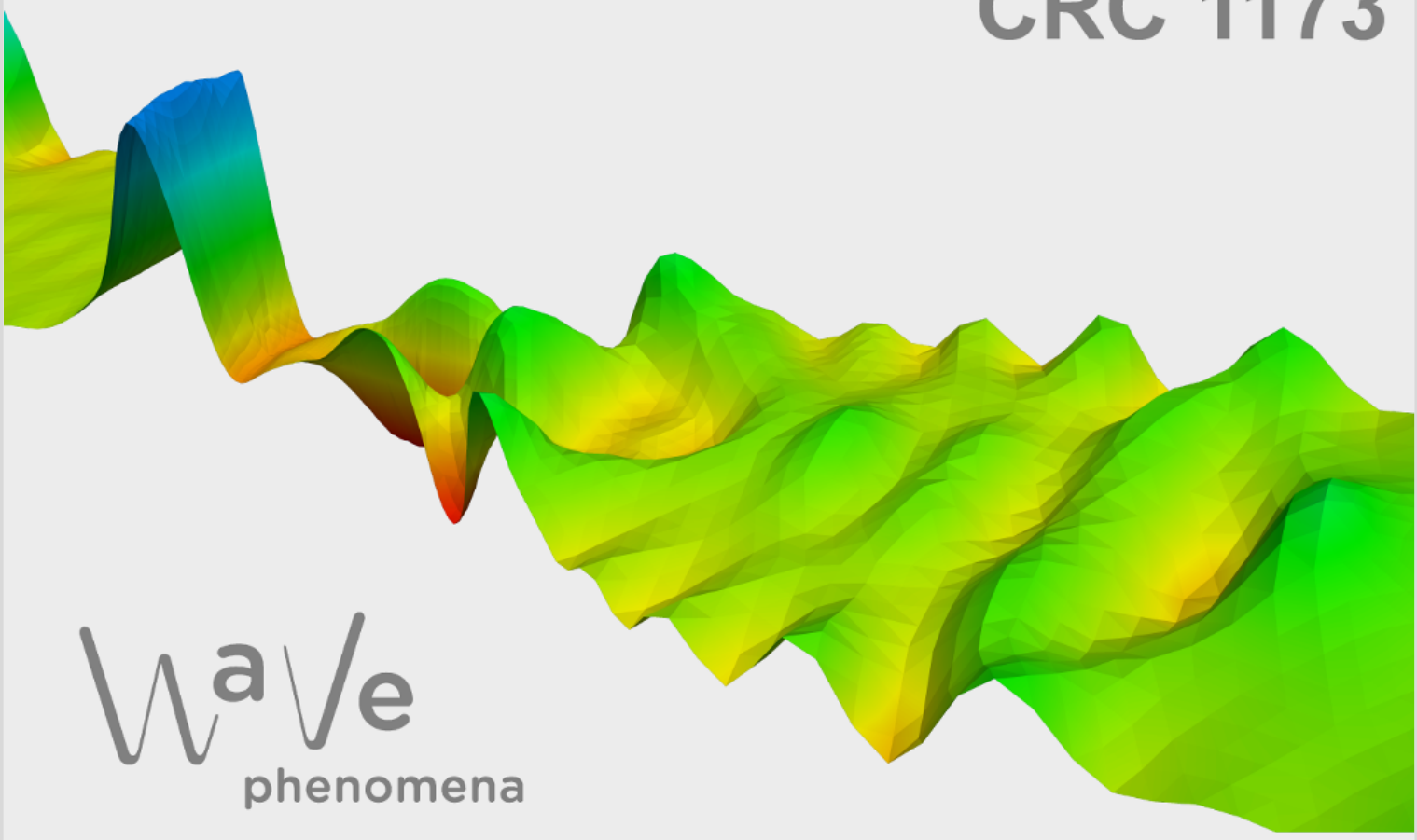
A Chebychev method for matrix function approximation

Daniel Eckhardt, Volker Grimm, Marlis Hochbruck

CRC Preprint 2026/17, May 2026

KARLSRUHE INSTITUTE OF TECHNOLOGY

CRC 1173



Participating universities

Funded by

ISSN 2365-662X

A Chebyshev Method for Matrix Function Approximation *

Daniel Eckhardt† Volker Grimm† Marlis Hochbruck†

Abstract. We present a Chebyshev-based method for approximating matrix functions or products of matrix functions with vectors. Our main interest is in matrix functions that arise in exponential integrators. The approach builds upon the construction of a Faber expansion of the function on an ellipse that encloses the field of values of the matrix. We derive error bounds for the proposed approximations and provide an efficient residual-based error estimator for the product of a matrix function with a vector, that can be computed efficiently using short recurrences. Since Faber polynomials rely on a priori information on the spectrum of the field of values of the matrix, we propose a novel algorithm to determine a suitable ellipse from appropriate Ritz values. Numerical examples demonstrate the effectiveness of the proposed method.

Key words. Krylov subspace methods, Chebyshev, Faber, matrix functions, exponential integrators, residuals, Arnoldi algorithm, error estimation

AMS subject classifications. 15A16, 65F60, 65F15, 65L05, 41A25, 41A10

1. Introduction

The numerical approximation of matrix function

$$f(\mathbf{S}) \quad \text{or} \quad f(\mathbf{S})v, \qquad \text{where} \qquad \mathbf{S} \in \mathbb{C}^{N\times N}, \quad v \in \mathbb{C}^N,$$

where $f\colon \mathbb{C} \to \mathbb{C}$ is analytic in a neighborhood of the field of values of $\mathbf{S}$, plays a fundamental role in a wide range of scientific and engineering applications (e.g., Section 2 in [24]). Our prime interest is in φ_k-functions defined as

$$\varphi_0(z) = e^z, \qquad \varphi_k(z) = \int_0^1 e^{(1-s)z} \frac{s^{k-1}}{(k-1)!}\, ds, \quad k \geq 1\,. \tag{1.1}$$

These analytic functions arise in connection with exponential integrators, which are used to solve large systems of ordinary differential equations [28]. Exponential integrators are particularly effective in applications involving stiff or highly oscillatory problems. A common way to approximate matrix functions is to first construct a projection of the original matrix onto a lower-dimensional Krylov subspace. This is done by building a suitable basis of this subspace using the Arnoldi or Lanczos algorithm. The matrix function is then calculated using the significantly smaller projected matrix (cf., e.g., [23, 27, 38]). However, constructing such a basis is computationally expensive, particularly when weighted inner products are required, as it is

*Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 258734477 – SFB 1173.

†Institut für Angewandte und Numerische Mathematik, Karlsruher Institut für Technologie, Englerstr. 2, D-76131 Karlsruhe

necessary for matrices resulting from spatial discretizations of differential operators methods. In addition, even for the Lanczos method, which uses short recurrence, memory requirements can be high, since the entire basis must be stored.

To address these issues, we consider an approach for approximating matrix functions in the spirit of semiiterative methods [19], which relies on approximations of the field-of-values of the matrix. In particular, we employ Faber polynomials associated with suitable ellipses. These polynomials can be computed efficiently via a three-term recurrence, leading to a significant reduction in both computational cost and memory requirements. Since, for an ellipse with distinct semi-axes, Faber polynomials coincide with shifted and scaled Chebyshev polynomials, we refer to the proposed approach as *Chebyshev method*.

Chebyshev and Faber polynomials are widely used for approximating matrix functions, see, e.g., [28, 30, 31]. Error analyses, particularly for the matrix exponential, can be found in [4, 14, 15, 16, 27, 36, 39]. In this our work, we obtain error bounds that apply to the broader class of φ_k-functions. Our approach builds on the results of [3] and extends them to general inner products and their corresponding norms. This extension is motivated by applications in finite element discretization of partial differential equations, where discrete Sobolev norms naturally arise (see, e.g., [18, 26]). A popular alternative to Krylov subspace approximations is interpolation at Leja points [9, 8, 29], which provides superlinear convergence rates but also requires a prior information about the field-of-values.

The main contribution of this work is the development of an efficient algorithm for approximating the matrix φ_k-functions using the Chebyshev method, focusing on two main aspects: First, we provide a residual-based error estimator to terminate the Chebyshev method. Second, we develop a method for constructing suitable ellipses by approximating the field-of-values with the convex hull of suitable Ritz values. Our approach employs spectral preconditioning techniques in combination with the Arnoldi algorithm. This idea was originally used to compute eigenvalues with the largest (or smallest) real part, see, e.g., [17, 25, 35, 40]. In the context of exponential integrators, the proposed algorithm can be viewed as a preprocessing step within an exponential integrator using a fixed linearization, since the ellipse needs to be constructed only once and can then be reused at each time step. In contrast to [39], which determines ellipses for specific problems based on empirical results from numerical experiments, our approach is significantly more general.

The paper is structured as follows. In Section 2, we present the Chebyshev method for approximating matrix functions and derive the error estimates. In Section 3, we introduce a residual-based strategy for selecting the degree of the approximating polynomial. The algorithm for computing the enclosing ellipse is described in Section 4. Finally, in Section 5, we apply the method to systems of spatially discretized partial differential equations and report numerical results, illustrating its effectiveness compared to other methods.

2. Error bounds for truncated Faber/Chebyshev series

We are interested in the approximation of matrix functions for matrices $\mathbf{S}$ whose field-of-values satisfies

$$W(\mathbf{S}) = \big\{(\mathbf{S}v, v) \mid v \in \mathbb{C}^N, \|v\| = 1\big\} \subset K \tag{2.1}$$

where K is a compact set bounded by an ellipse of the form

$$\mathcal{E}_\gamma^{\alpha,\beta} = \{z \in \mathbb{C} \mid z = \alpha\cos\eta + \mathrm{i}\,\beta\sin\eta + \gamma, \eta \in [-\pi, \pi]\}, \qquad \alpha, \beta \geq 0,\ \alpha^2 + \beta^2 > 0, \quad \text{and } \gamma \in \mathbb{R}.$$

Here, $\|\cdot\|$ denotes a suitable norm in $\mathbb{C}^N$ which is induced by an inner product $(\cdot,\cdot)$. The induced matrix norm is also designated by $\|\cdot\|$. Let $\mathbf{S}^\dagger$ be the adjoint matrix to $\mathbf{S}$ with respect to $(\cdot,\cdot)$.

If $\mathbf{S}$ is normal, i.e., $\mathbf{S}^\dagger\mathbf{S} = \mathbf{S}\mathbf{S}^\dagger$, then the field-of-values is the convex hull of the eigenvalues of $\mathbf{S}$. We will use Faber polynomials with respect to the set K in order to approximate the function. For the convenience of the reader, we review some basic facts about Faber polynomials and Faber series. For more detailed properties of Faber polynomials, we refer to the book by Gaier [22], the survey by Suetin [46], and [19], respectively. Let $\mathbb{D}_\rho = \{z \in \mathbb{C} \mid |z| \leq \rho\}$ be the closed disk with radius ρ and center zero. The mapping

$$\psi(w) = \frac{\alpha+\beta}{2}\, w + \gamma + \frac{\alpha-\beta}{2}\,\frac{1}{w}, \qquad \alpha,\beta \geq 0, \quad \alpha^2+\beta^2 > 0, \quad \gamma \in \mathbb{R}, \tag{2.2a}$$

maps the exterior of the unit disc $\mathbb{D}$ to the exterior of the ellipse $\mathcal{E}_\gamma^{\alpha,\beta}$. The map is holomorphic and one-to-one for $|w| > \sqrt{|\alpha-\beta|/(\alpha+\beta)}$ (cf. page 582 in [12]). The boundary of the unit circle is mapped to the ellipse $\mathcal{E}_\gamma^{\alpha,\beta}$, i.e., $\mathcal{E}_\gamma^{\alpha,\beta} = \psi(\partial\mathbb{D})$. The sets

$$K_\rho := \mathbb{C}\backslash\psi(\mathbb{C}\backslash\mathbb{D}_\rho), \qquad \rho \geq 1, \tag{2.2b}$$

are compact sets with ellipses as boundaries. We also use the short notation $\mathbb{D} = \mathbb{D}_1$ and $K = K_1$. The coefficient $(\alpha+\beta)/2$ is commonly referred to as the logarithmic capacity of K, p. 51 in [49]. The Faber polynomials F_ℓ, $\ell = 1, 2, \ldots$, for the set K are defined by the generating function

$$\frac{w\psi'(w)}{\psi(w)-z} = 1 + \sum_{\ell=1}^{\infty} F_\ell(z) w^{-\ell}\,,$$

where $z \in K$ and $|w| > 1$. By equating the coefficients, one obtains the three-term recursion

$$\begin{aligned} F_0 &\equiv 2, \qquad F_1(z) = 2\,\frac{z-\gamma}{\alpha+\beta}\,, \\ F_{\ell+1}(z) &= 2\,\frac{z-\gamma}{\alpha+\beta}F_\ell(z) - \frac{\alpha-\beta}{\alpha+\beta}F_{\ell-1}(z)\,, \qquad \ell = 1, 2, \ldots \end{aligned} \tag{2.3}$$

It is well known that Faber polynomials for ellipses ($\alpha \neq \beta$) are shifted and scaled Chebyshev polynomials (cf. [3, 12, 46]). More exactly, the Faber polynomials (2.3), are given as

$$F_\ell(z) = 2\left(\frac{\alpha-\beta}{\alpha+\beta}\right)^{\frac{\ell}{2}} T_\ell\left(\frac{z-\gamma}{\sqrt{\alpha^2-\beta^2}}\right), \qquad \ell = 1, 2, \ldots\,, \tag{2.4}$$

where T_ℓ are the Chebyshev polynomials of the first kind. The branch of the square root needs to be chosen appropriately. For circles, $\alpha = \beta$, the Faber polynomials are shifted and scaled monomials

$$F_\ell(z) = \left(\frac{z-\gamma}{\alpha}\right)^\ell, \qquad \ell = 1, 2, \ldots\,. \tag{2.5}$$

Functions holomorphic in a neighborhood of K can be given as a series of Faber polynomials. The following theorem is inspired by Faber's original ideas (cf. [20, 21]) and Section 5 of [12].

Theorem 2.1. *Let F_ℓ be the Faber polynomials for the set $K = K_1$ defined in (2.2). Let $\Omega \supset K$ be an open set and f be holomorphic on Ω. Then there is a $\rho^* > 1$ such that $K_\rho \subset \Omega$ for $1 \leq \rho < \rho^*$. The Faber series for the function f*

$$f(z) = c_0 + \sum_{\ell=1}^{\infty} c_\ell F_\ell(z), \qquad c_\ell = \frac{1}{2\pi i}\int_{\partial\mathbb{D}} f(\psi(w)) w^{-\ell-1}\, dw\,,$$

converges absolutely and uniformly on all sets K_ρ, $1 \le \rho < \rho^*$. For the truncated series

$$p_m(z) := c_0 + \sum_{\ell=1}^{m} c_\ell F_\ell(z)\,,$$

we have the estimate

$$\|f - p_{m-1}\|_{L^\infty(K)} \le 2\sum_{\ell=m}^{\infty} |c_\ell| \le 2\|f\|_{L^\infty(K_\rho)} \frac{\rho^{-m}}{1-\rho^{-1}}, \qquad \rho^* > \rho > 1\,,$$

where

$$\|f\|_{L^\infty(K)} := \sup_{z\in K} |f(z)|\,.$$

Proof. The existence of $\rho^* > 1$ follows from the fact that Ω is open and K is compact. Pick ρ such that $1 < \rho < \rho^*$. Let z be in the interior of K_ρ. Then, the Cauchy integral theorem gives

$$\begin{aligned} f(z) &= \frac{1}{2\pi i}\int_{\psi(\partial\mathbb{D}_\rho)} f(\xi)\frac{1}{\xi - z}\,d\xi = \frac{1}{2\pi i}\int_{\partial\mathbb{D}_\rho} f(\psi(w))\frac{\psi'(w)}{\psi(w)-z}\,dw \\ &= \frac{1}{2\pi i}\int_{\partial\mathbb{D}_\rho} \frac{f(\psi(w))}{w}\frac{w\psi'(w)}{\psi(w)-z}\,dw = \frac{1}{2\pi i}\int_{\partial\mathbb{D}_\rho} \frac{f(\psi(w))}{w}\left(1 + \sum_{\ell=1}^{\infty} F_\ell(z) w^{-\ell}\right) dw\,. \end{aligned}$$

Since the series is uniformly convergent on $\partial\mathbb{D}_\rho$, the series and the integral can be exchanged. This gives the series with the coefficients

$$c_\ell = \frac{1}{2\pi i}\int_{\partial\mathbb{D}_\rho} f(\psi(w)) w^{-\ell-1}\,dw\,.$$

Since $f\circ\psi$ is holomorphic for $\rho^* > |w| > \rho_*$, $\rho_* < 1$, the contour can be changed to $\partial\mathbb{D}$. For $\alpha \ne 0$ and $\beta \ne 0$, one might choose $\rho_* = \sqrt{|\alpha-\beta|/(\alpha+\beta)}$, for the remaining cases, one might choose $\rho_* = 1/\rho^*$. The coefficient c_ℓ can be estimated by

$$|c_\ell| = \left|\frac{1}{2\pi i}\int_{\partial\mathbb{D}_\rho} f(\psi(w)) w^{-\ell-1}\,dw\right| \le \frac{1}{\rho^\ell}\|f\circ\psi\|_{L^\infty(\partial\mathbb{D}_\rho)} = \frac{1}{\rho^\ell}\|f\|_{L^\infty(K_\rho)}\,,$$

where the last equality follows from the maximum modulus principle. By Bernstein's lemma (cf. Lemma 3.6.2 in [37]) and Theorem 2.1 in [3], we have that

$$\|F_\ell\|_{L^\infty(K_{\tilde\rho})} \le \tilde\rho^\ell \|F_\ell\|_{L^\infty(K)} \le 2\tilde\rho^\ell\,, \qquad \tilde\rho \ge 1\,.$$

Let now $1 \le \tilde\rho < \rho$, then

$$\|c_\ell F_\ell\|_{L^\infty(K_{\tilde\rho})} \le |c_\ell|\|F_\ell\|_{L^\infty(K_{\tilde\rho})} \le 2\left(\frac{\tilde\rho}{\rho}\right)^\ell \|f\|_{L^\infty(K_\rho)}\,.$$

From this, the absolute and uniform convergence on $K_{\tilde\rho}$ follows. For $\tilde\rho = 1$, the estimate of the truncated series on K is deduced. □

Theorem 2.1 can be transferred to matrix functions for matrices $\mathbf{S}$ with a field-of-values contained in the set K, i.e. $W(\mathbf{S}) \subset K$. The following result is a corollary of Theorem 2.1 and of Theorem 2.1 in [3].

Corollary 2.2. *Let F_ℓ be the Faber polynomials for the set K, cf. (2.2b). Let $\Omega \supset K$ be an open set and f be holomorphic on Ω. Further, let $\mathbf{S}$ be a matrix such that $W(\mathbf{S}) \subset K$. Then, the matrix series*

$$f(\mathbf{S}) = c_0 I + \sum_{\ell=1}^{\infty} c_\ell F_\ell(\mathbf{S}), \qquad c_\ell = \frac{1}{2\pi i} \int_{\partial\mathbb{D}} f(\psi(w)) w^{-\ell-1}\, dw\,,$$

converges in the operator norm. Here, I designates the identity matrix. For the truncated series

$$p_m(\mathbf{S}) := c_0 I + \sum_{\ell=1}^{m} c_\ell F_\ell(\mathbf{S})\,,$$

we have the estimate

$$\|f(\mathbf{S}) - p_{m-1}(\mathbf{S})\| \le 2 \sum_{\ell=m}^{\infty} |c_\ell|\,,$$

where the last series is convergent with the same bound as in Theorem 2.1.

Proof. By Theorem 2.1 in [3], we find the bound

$$\|F_\ell(\mathbf{S})\| \le 2, \qquad \ell = 1, 2, \dots\,,$$

for the Euclidean inner product and its corresponding operator norm. The proof of Theorem 2.1 in [3] can be extended straightforwardly to an arbitrary inner product in $\mathbb{C}^N$, by replacing A^T in the proof with the complex conjugate of the adjoint matrix with respect to the inner product, $\overline{A^\dagger}$, and by replacing the Euclidean inner product with the general one. Hence,

$$\|f(\mathbf{S}) - p_{m-1}(\mathbf{S})\| \le \sum_{\ell=m}^{\infty} |c_\ell| \|F_\ell(\mathbf{S})\| \le 2 \sum_{\ell=m}^{\infty} |c_\ell| \le 2 \|f\|_{L^\infty(K_\rho)} \frac{\rho^{-m}}{1-\rho^{-1}}\,,$$

where $1 < \rho < \rho^*$ has been chosen according to Theorem 2.1. □

For the Faber series of the φ_k-functions, cf. (1.1), we obtain the following theorem.

Theorem 2.3. *Let $\mathbf{S}$ satisfy (2.1) with $K = K_1$ defined in (2.2). Let F_ℓ be the Faber polynomials for the set K, cf. (2.2b). Then, the matrix series*

$$\varphi_k(\mathbf{S}) = c_0^{(k)} I + \sum_{\ell=1}^{\infty} c_\ell^{(k)} F_\ell(\mathbf{S}), \qquad c_\ell^{(k)} = \frac{1}{2\pi i} \int_{\partial\mathbb{D}} \varphi_k(\psi(w)) w^{-\ell-1}\, dw\,,$$

converges in the matrix norm. For the truncated series

$$p_m^{(k)}(\mathbf{S}) := c_0^{(k)} I + \sum_{\ell=1}^{m} c_\ell^{(k)} F_\ell(\mathbf{S})\,, \tag{2.6}$$

we have the estimates

$$\|\varphi_k(\mathbf{S}) - p_{m-1}^{(k)}(\mathbf{S})\| \le \frac{4\sqrt{\pi}}{\sqrt{2m}} \frac{m!}{(m+k)!} e^{\alpha+\gamma} \left(\frac{(\alpha+\beta)e}{2m}\right)^m, \qquad m \ge \alpha+\beta\,.$$

Proof. For the exponential φ_0, analogously to the proof of Corollary 4.1 in [3], we find the bound

$$|c_\ell^{(0)}| \le \frac{\sqrt{\pi}}{\sqrt{2\ell}} e^{\alpha+\gamma} \left(\frac{(\alpha+\beta)e}{2\ell}\right)^\ell, \qquad \ell > \alpha+\beta\,.$$

With the representation of the coefficients of the Faber series for the φ_k-functions in Example 4.3 in [3], we find for $k \ge 1$ the bounds

$$|c_\ell^{(k)}| \le \frac{\sqrt{\pi}}{\sqrt{2\ell}} \frac{\ell!}{(\ell+k)!} e^{\alpha+\gamma} \left(\frac{(\alpha+\beta)e}{2\ell}\right)^\ell, \qquad \ell > \alpha+\beta\,.$$

The last formula is the correct estimate for $k = 0$, too. For $k = 0, 1, 2, \ldots$ and $m > \alpha+\beta$, we therefore obtain the estimate

$$\begin{aligned}
\sum_{\ell=m}^{\infty} |c_\ell^{(k)}| &\le \frac{\sqrt{\pi}}{\sqrt{2m}} \frac{m!}{(m+k)!} e^{\alpha+\gamma} \sum_{\ell=m}^{\infty} \left(\frac{(\alpha+\beta)e}{2\ell}\right)^\ell \\
&\le \frac{\sqrt{\pi}}{\sqrt{2m}} \frac{m!}{(m+k)!} e^{\alpha+\gamma} \sum_{\ell=m}^{\infty} e^m \left(\frac{(\alpha+\beta)}{2m}\right)^\ell \\
&= \frac{\sqrt{\pi}}{\sqrt{2m}} \frac{m!}{(m+k)!} e^{\alpha+\gamma} \left(\frac{(\alpha+\beta)e}{2m}\right)^m \sum_{\ell=0}^{\infty} \rho^{-\ell} \\
&\le 2\frac{\sqrt{\pi}}{\sqrt{2m}} \frac{m!}{(m+k)!} e^{\alpha+\gamma} \left(\frac{(\alpha+\beta)e}{2m}\right)^m,
\end{aligned}$$

where $\rho = \frac{2m}{\alpha+\beta} > 2$. Corollary 2.2 applied to the φ_k-functions now proves the statement of Theorem 2.3. □

Theorem 2.3 is our main result for the error of the approximation of the φ_k-functions by Faber polynomials. Note that the coefficients of the Faber series are Fourier coefficients of the periodic smooth function $\varphi_k(\psi(\exp(i\theta))) = \varphi_k(\alpha\cos\theta + \mathrm{i}\,\beta\sin\theta + \gamma)$, $\theta \in [-\pi, \pi]$, which can be approximated efficiently by fast Fourier transforms (e.g. [47]). The evaluation of the Faber polynomials is efficient due to the three-term recursion (2.3).

3. Residual-based error estimator

To achieve the desired accuracy, we propose a strategy for estimating the degree m of the Faber approximation $p_m^{(k)}$ defined in (2.6). Here, the matrix functions $\varphi_k(\tau\mathbf{S})$ are considered for some time step $\tau > 0$. As in [7, 29], we consider the initial value problems

$$(Y^{(k)})'(\tau) = \begin{cases} \mathbf{S}Y^{(0)}(\tau), & Y^{(0)}(0) = I, \quad \text{if } k = 0, \\ \mathbf{S}Y^{(k)}(\tau) + \frac{\tau^{k-1}}{(k-1)!} I, & Y^{(k)}(0) = 0, \quad \text{if } k \ge 1, \end{cases}$$

whose solutions are given by

$$Y^{(k)}(\tau) = \tau^k \varphi_k(\tau\mathbf{S}), \qquad k \ge 0.$$

For the corresponding Faber approximations $Y_m^{(k)}(\tau) = \tau^k p_m^{(k)}(\tau\mathbf{S})$, $k \ge 0$ it can be shown that the residuals defined by

$$\begin{aligned}
R_m^{(0)}(\tau) &= \rho_m^{(0)}(\tau) && \text{if } k = 0, \\
R_m^{(k)}(\tau) &= \rho_m^{(k)}(\tau) + \frac{\tau^{k-1}}{(k-1)!} && \text{if } k \ge 1, \\
\rho_m^{(k)}(\tau) &= \mathbf{S}Y_m^{(k)}(\tau) - (Y_m^{(k)})'(\tau)
\end{aligned} \tag{3.1}$$

can be used to estimate the error $\|Y^{(k)}(\tau) - Y_m^{(k)}(\tau)\|$, and it holds

$$\begin{aligned} R_m^{(0)}(\tau) &= \Big(\mathbf{S}p_m^{(k)}(\tau\mathbf{S}) - \frac{d}{d\tau}p_m^{(k)}(\tau\mathbf{S})\Big), && \text{if } k = 0, \\ R_m^{(k)}(\tau) &= \tau^{k-1}\Big(\tau R_m^{(0)}(\tau) + \frac{I}{(k-1)!} - kp_m^{(k)}(\tau\mathbf{S})\Big), && \text{if } k \geq 1. \end{aligned} \tag{3.2}$$

(cf. [7, 29]). To compute the required quantities, we use the fact that Faber polynomials reduce to shifted and scaled Chebyshev polynomials when ellipses $\mathcal{E}_\gamma^{\alpha,\beta}$ with $\alpha \neq \beta$ are considered (cf. (2.4)). Consequently, the same technique as in [7] can be applied, where Chebyshev polynomials of first and second kind are used. We define a weighted version of these polynomials as

$$\widetilde{U}_j = \vartheta^{\frac{j}{2}} U_j, \qquad \vartheta = \frac{\alpha - \beta}{\alpha + \beta}, \tag{3.3}$$

where U_j is the standard jth Chebyshev polynomial of second kind (see [7, eq.(3.8)]). A simple calculation shows that $\widetilde{U}_j$ satisfies the three-term recurrence

$$\widetilde{U}_j(z) = 2z\sqrt{\vartheta}\widetilde{U}_{j-1}(z) - \vartheta\widetilde{U}_{j-2}(z), \quad j \geq 2$$

with

$$\widetilde{U}_1(z) = 2\sqrt{\vartheta}z, \quad \widetilde{U}_0(z) = 1, \quad \widetilde{U}_{-1}(z) = 0, \quad \widetilde{U}_{-2}(z) = -\tfrac{1}{\vartheta}.$$

Moreover, the following relations can be shown.

Lemma 3.1. *Let*

$$\widetilde{T}_j(z) = \vartheta^{\frac{j}{2}} T_j(z),$$

where T_j is the jth Chebyshev polynomial of first kind. For the weighted Chebyshev polynomials the following identities hold true:

$$(\widetilde{T}_j)'(z) = j\sqrt{\vartheta}\widetilde{U}_{j-1}(z), \tag{3.4a}$$

$$z\widetilde{T}_j(z) = \frac{1}{2}\big(\tfrac{1}{\sqrt{\vartheta}}\widetilde{T}_{j+1}(z) + \sqrt{\vartheta}\widetilde{T}_{j-1}(z)\big), \tag{3.4b}$$

$$z\widetilde{U}_j(z) = \frac{1}{2}\big(\tfrac{1}{\sqrt{\vartheta}}\widetilde{U}_{j+1}(z) + \sqrt{\vartheta}\widetilde{U}_{j-1}(z)\big), \tag{3.4c}$$

$$\widetilde{T}_j(z) = \frac{1}{2}\big(\widetilde{U}_j(z) - \vartheta\widetilde{U}_{j-2}(z)\big), \tag{3.4d}$$

Proof. Follows directly from [7, eq.(3.4)–(3.7)] and (3.3). □

The following Lemma shows that the residual (3.2) can be efficiently approximated using the short-term Chebyshev recurrences.

Lemma 3.2. *Let $p_m^{(k)}$ be defined as in Theorem 2.3 and*

$$\widehat{\tau\mathbf{S}} = \frac{\tau\mathbf{S} - \gamma I}{\sqrt{\alpha^2 - \beta^2}}, \qquad \alpha \neq \beta.$$

Then, we have

$$p_m^{(k)}(\tau\mathbf{S}) = c_0^{(k)} I + \sum_{j=1}^{m} c_j^{(k)} \Big(\widetilde{U}_j(\widehat{\tau\mathbf{S}}) - \frac{\alpha - \beta}{\alpha + \beta}\,\widetilde{U}_{j-2}(\widehat{\tau\mathbf{S}})\Big). \tag{3.5}$$

Further, the residual terms in (3.2) can be expressed by

$$\tau\frac{d}{d\tau}p_m^{(k)}(\tau\mathbf{S}) = \sum_{j=1}^{m} c_j^{(k)} j\Big(\widetilde{U}_j(\widehat{\tau\mathbf{S}}) + \frac{2\gamma}{\alpha+\beta}\widetilde{U}_{j-1}(\widehat{\tau\mathbf{S}}) + \frac{\alpha-\beta}{\alpha+\beta}\widetilde{U}_{j-2}(\widehat{\tau\mathbf{S}})\Big), \tag{3.6}$$

and

$$\begin{aligned}\tau\mathbf{S}p_m^{(k)}(\tau\mathbf{S}) = c_0^{(k)}\tau\mathbf{S} + \sum_{j=1}^{m}\frac{c_j^{(k)}}{2}\Big((\alpha+\beta)\widetilde{U}_{j+1}(\widehat{\tau\mathbf{S}}) + 2\gamma\Big(\widetilde{U}_j(\widehat{\tau\mathbf{S}}) - \frac{\alpha-\beta}{\alpha+\beta}\,\widetilde{U}_{j-2}(\widehat{\tau\mathbf{S}})\Big)\\ + \frac{(\alpha-\beta)^2}{\alpha+\beta}\,\widetilde{U}_{j-3}(\widehat{\tau\mathbf{S}})\Big).\end{aligned} \tag{3.7}$$

Proof. By (2.4) we have $2\widetilde{T}_j(\widehat{\tau\mathbf{S}}) = F_j(\tau\mathbf{S})$. Then, (3.5) follows directly from (3.4d). To derive (3.6) we use (3.4a) and (3.4c):

$$\begin{aligned}\tau\frac{d}{d\tau}p_m^{(k)}(\tau\mathbf{S}) &= 2\sum_{j=1}^{m} c_j^{(k)}\tau\frac{d}{d\tau}\widetilde{T}_j(\widehat{\tau\mathbf{S}})\\ &= 2\sum_{j=1}^{m} c_j^{(k)}\frac{\tau\mathbf{S}}{\sqrt{\alpha^2-\beta^2}}(\widetilde{T}_j)'(\widehat{\tau\mathbf{S}})\\ &= 2\sum_{j=1}^{m} c_j^{(k)} j\Big(\sqrt{\vartheta}\widehat{\tau\mathbf{S}}\widetilde{U}_{j-1}(\widehat{\tau\mathbf{S}}) + \frac{\gamma}{\sqrt{\alpha^2-\beta^2}}\sqrt{\vartheta}\widetilde{U}_{j-1}(\widehat{\tau\mathbf{S}})\Big)\\ &= 2\sum_{j=1}^{m} c_j^{(k)} j\Big(\frac{1}{2}\widetilde{U}_j(\widehat{\tau\mathbf{S}}) + \frac{\gamma}{\alpha+\beta}\widetilde{U}_{j-1}(\widehat{\tau\mathbf{S}}) + \frac{1}{2}\vartheta\widetilde{U}_{j-2}(\widehat{\tau\mathbf{S}})\Big).\end{aligned}$$

We obtain (3.7) by writing

$$\tau\mathbf{S}p_m^{(k)}(\tau\mathbf{S}) = c_0^{(k)}\tau\mathbf{S} + 2\sum_{j=1}^{m} c_j^{(k)}\sqrt{\alpha^2-\beta^2}\frac{\tau\mathbf{S}}{\sqrt{\alpha^2-\beta^2}}\widetilde{T}_j(\widehat{\tau\mathbf{S}})v$$

and using (3.4b) and (3.4d) to express

$$\begin{aligned}\frac{\tau\mathbf{S}}{\sqrt{\alpha^2-\beta^2}}\widetilde{T}_j(\widehat{\tau\mathbf{S}}) &= \widehat{\tau\mathbf{S}}\widetilde{T}_j(\widehat{\tau\mathbf{S}}) + \frac{\gamma}{\sqrt{\alpha^2-\beta^2}}\widetilde{T}_j(\widehat{\tau\mathbf{S}})\\ &= \frac{1}{2}\frac{1}{\sqrt{\vartheta}}\widetilde{T}_{j+1}(\widehat{\tau\mathbf{S}}) + \frac{\gamma}{\sqrt{\alpha^2-\beta^2}}\widetilde{T}_j(\widehat{\tau\mathbf{S}}) + \frac{1}{2}\sqrt{\vartheta}\widetilde{T}_{j-1}(\widehat{\tau\mathbf{S}})\\ &= \frac{1}{4}\Big(\frac{1}{\sqrt{\vartheta}}\widetilde{U}_{j+1}(\widehat{\tau\mathbf{S}}) - \sqrt{\vartheta}\widetilde{U}_{j-1}(\widehat{\tau\mathbf{S}})\Big)\\ &\quad + \frac{\gamma}{2\sqrt{\alpha^2-\beta^2}}\Big(\widetilde{U}_j(\widehat{\tau\mathbf{S}}) - \vartheta\widetilde{U}_{j-2}(\widehat{\tau\mathbf{S}})\Big)\\ &\quad + \frac{1}{4}\Big(\sqrt{\vartheta}\widetilde{U}_{j-1}(\widehat{\tau\mathbf{S}}) - \vartheta^{\frac{3}{2}}\widetilde{U}_{j-3}(\widehat{\tau\mathbf{S}})\Big)\\ &= \frac{1}{4}\frac{1}{\sqrt{\vartheta}}\widetilde{U}_{j+1}(\widehat{\tau\mathbf{S}}) + \frac{\gamma}{2\sqrt{\alpha^2-\beta^2}}\Big(\widetilde{U}_j(\widehat{\tau\mathbf{S}}) - \vartheta\widetilde{U}_{j-2}(\widehat{\tau\mathbf{S}})\Big)\\ &\quad - \frac{1}{4}\vartheta^{\frac{3}{2}}\widetilde{U}_{j-3}(\widehat{\tau\mathbf{S}}),\end{aligned}$$

which leads to

$$
\begin{aligned}
&2\sum_{j=1}^{m} c_j^{(k)} \sqrt{\alpha^2-\beta^2}\, \frac{\tau\mathbf{S}}{\sqrt{\alpha^2-\beta^2}} \widetilde{T}_j(\widehat{\tau\mathbf{S}}) \\
=&2\sum_{j=1}^{m} \frac{c_j^{(k)}}{4} \sqrt{\alpha^2-\beta^2}\Big(\frac{1}{\sqrt{\vartheta}} \widetilde{U}_{j+1}(\widehat{\tau\mathbf{S}}) + \frac{2\gamma}{\sqrt{\alpha^2-\beta^2}} \Big(\widetilde{U}_j(\widehat{\tau\mathbf{S}}) - \vartheta \widetilde{U}_{j-2}(\widehat{\tau\mathbf{S}}) \Big) - \vartheta^{\frac{3}{2}} \widetilde{U}_{j-3}(\widehat{\tau\mathbf{S}}) \Big) \\
=&\sum_{j=1}^{m} \frac{c_j^{(k)}}{2} \Big((\alpha+\beta) \widetilde{U}_{j+1}(\widehat{\tau\mathbf{S}}) + 2\gamma \big(\widetilde{U}_j(\widehat{\tau\mathbf{S}}) - \frac{\alpha-\beta}{\alpha+\beta} \widetilde{U}_{j-2}(\widehat{\tau\mathbf{S}}) \big) - \frac{(\alpha-\beta)^2}{\alpha+\beta} \widetilde{U}_{j-3}(\widehat{\tau\mathbf{S}}) \Big).
\end{aligned}
$$

This completes the proof. □

In order to estimate the error of the Faber approximation to $\varphi_k(\tau\mathbf{S})v$ for a vector $v \in \mathbb{C}^N$, we compute $R_m^{(k)}(\tau)v$ defined in (3.2) using the short-term recurrence of Lemma 3.2. Note that although more vectors must be stored, the additional computational effort increases only moderately, since these vectors can also be reused for the evaluation of (3.5). Finally, [7] provides pseudocode for a related but simpler case, in which only the matrix exponential is considered and where $\mathbf{S}$ is (skew-)symmetric.

4. Efficient calculation of an optimal ellipse

We seek a *best-fitting* ellipse $\mathcal{E}$ that encloses the field-of-values $W(\mathbf{S})$ in order to provide a reliable approximation, but which is also "small" enough to guarantee convergence with optimal order. In this section we restrict ourselves to real matrices $\mathbf{S}$. Since the field-of-values is generally difficult to compute, we instead approximate it by the convex hull of a few "outer" Ritz values which are computed by a preconditioned Arnoldi iteration. Note that this is a preprocessing step which is necessary for all semiiterative methods, see, e.g., [19]. If $\mathbf{S}$ is nearly normal, using an ellipse computed from appropriate Ritz values in the method of Section 2 provides a reliable approximation, as demonstrated in the numerical experiments in Section 5.

4.1. Construction of the best-fitting ellipse from a discrete complex set

We begin by constructing a best-fitting ellipse for a finite set of points M in the complex plane. Following the approach of [34], we assume w.l.o.g. that $M \subset \mathbb{C}_+ = \{z \in \mathbb{C} \mid \mathrm{Re} z > 0\}$. Although our primary interest is in the case where all points lie in the left half-plane, the method can be easily adapted by first reflecting the points across the imaginary axis, computing the ellipse $\mathcal{E}_\gamma^{\alpha,\beta}$ in the right half-plane, and return $\mathcal{E}_{-\gamma}^{\alpha,\beta}$.

4.1.1. Notation and assumptions

We assume that M is symmetric with respect to the real axis. The linear eccentricity of the ellipse $\mathcal{E}_\gamma^{\alpha,\beta}$ is defined as $\epsilon = \sqrt{\alpha^2-\beta^2} \in \mathbb{C}$. Typically, the linear eccentricity is defined as a real quantity. However, in our setting we allow ϵ to be complex in order to distinguish between the two possible cases that may arise: the ellipse is *horizontal*, with the major semi-axis α parallel to the real axis, in which case $\epsilon \in \mathbb{R}_{\geq 0}$, or the ellipse is *vertical*, with the major semi-axis β parallel to the imaginary axis, in which case $\epsilon \in i\,\mathbb{R}_{>0}$.

Note that these ellipses can be parametrized either by the α, β, and γ or ϵ, γ, and an arbitrary point λ on the ellipse. Therefore, we define the family of ellipses (see Figure 1) with the same

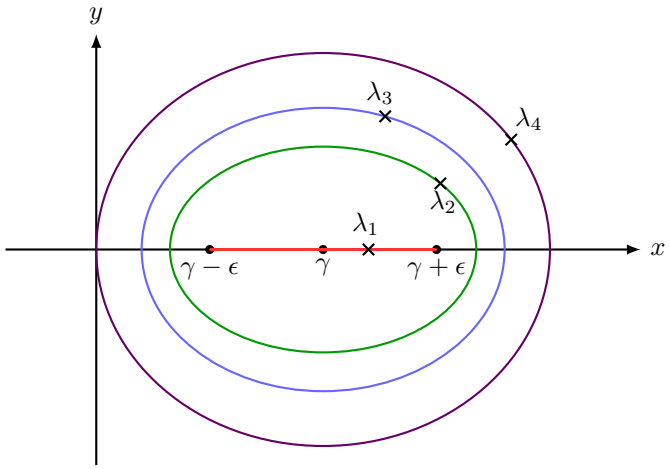


Figure 1: Ellipses from $\mathcal{F}_\gamma^\epsilon$, each uniquely determined by the corresponding λ_i. The ellipse determined by λ_4 is the member of the family that passes through the origin.

linear eccentricity and center as

$$\mathcal{F}_\gamma^\epsilon = \left\{ \mathcal{E}_\gamma^{\alpha,\beta} \mid \epsilon = \sqrt{\alpha^2 - \beta^2} \right\}.$$

4.1.2. The Min-Max problem

As proposed in [34] we consider the map

$$s(\epsilon, \gamma, \lambda) = \left| \frac{(\gamma - \lambda) + ((\gamma - \lambda)^2 - \epsilon^2)^{1/2}}{\gamma + (\gamma^2 - \epsilon^2)^{1/2}} \right|, \tag{4.1}$$

which satisfies the following properties: For two ellipses $\mathcal{E}_1, \mathcal{E}_2 \in \mathcal{F}_\gamma^\epsilon$ and two points $\lambda_1 \in \mathcal{E}_1$ and $\lambda_2 \in \mathcal{E}_2$ it holds that

$$\begin{aligned} s(\epsilon, \gamma, \lambda_1) > s(\epsilon, \gamma, \lambda_2) \quad &\iff \mathcal{E}_1 \text{ encloses } \mathcal{E}_2, \\ s(\epsilon, \gamma, \lambda_1) = s(\epsilon, \gamma, \lambda_2) \quad &\iff \mathcal{E}_1 = \mathcal{E}_2, \\ s(\epsilon, \gamma, \lambda_1) = 1 \quad &\iff 0 \in \mathcal{E}_1, \end{aligned} \tag{4.2}$$

(cf. [34, eq. (2.13) and (3.1)]). These properties lead two important observations. First, let ϵ and γ be fixed, and let s be maximized over $\lambda \in M$. Then the resulting ellipse encloses all points in M.

Second, we can rewrite (4.1) as follows. Let $\mathcal{E}_\gamma^{\alpha,\beta} \in \mathcal{F}_\gamma^\epsilon$ be an ellipse with $\lambda \in \mathcal{E}_\gamma^{\alpha,\beta}$ for some $\lambda \in M$. Furthermore, let $\mathcal{E}_\gamma^{\alpha_0,\beta_0} \in \mathcal{F}_\gamma^\epsilon$ the ellipse that passes through the origin., i.e., $0 \in \mathcal{E}_\gamma^{\alpha_0,\beta_0}$ (see Figure 1). By (4.2) and [25, Eq. (7)], we have

$$s(\epsilon, \gamma, \lambda) = s(\epsilon, \gamma, \alpha + \gamma) = \frac{\alpha + \beta}{\alpha_0 + \beta_0}, \tag{4.3}$$

where

$$\beta = \sqrt{\alpha^2 - \epsilon^2}, \quad \alpha_0 = |\gamma|, \quad \beta_0 = \sqrt{\alpha_0^2 - \epsilon^2}.$$

Thus, minimizing (4.1) with respect to ϵ and γ leads, heuristically speaking, to an ellipse with a small value of $\alpha + \beta$ and a large value of $\alpha_0 + \beta_0$. The latter means that the ellipse is as far as possible into the right half-plane, while still keeping λ on the ellipse.

This motivates the following definition of a *best-fitting* ellipse for M (cf. [34, Eq. (3.2)]).

Definition 4.1 (Best-fitting ellipse)**.** Let M be a finite set in $\mathbb{C}_+$. Then a *best-fitting* ellipse is a solution $\mathcal{E}_{\gamma^\star}^{\alpha^\star,\beta^\star} \in \mathcal{F}_{\gamma^\star}^{\epsilon^\star}$ of the min-max problem

$$\min_{(\epsilon,\gamma)\in R} \max_{\lambda\in M} s(\epsilon,\gamma,\lambda) = s(\epsilon^\star,\gamma^\star,\lambda^\star), \quad \text{for all } \lambda^\star \in \mathcal{E}_{\gamma^\star}^{\alpha^\star,\beta^\star}, \tag{4.4}$$

where $R := \Big((\mathbb{R}_{\geq 0} \cup \mathrm{i}\,\mathbb{R}_{>0}) \times \mathbb{R}\Big) \setminus \{(0,0)\}$.

Lemma 4.2. *Let M be a finite set of points in $\mathbb{C}_+$ for which the assumptions of Section 4.1.1 hold. If there exists a solution $\mathcal{E}_{\gamma^\star}^{\alpha^\star,\beta^\star}$ of the min-max problem (4.1), then $\mathcal{E}_{\gamma^\star}^{\alpha^\star,\beta^\star}$ encloses all points of M. Further, $\mathcal{E}_{\gamma^\star}^{\alpha^\star,\beta^\star} \subset \mathbb{C}_+$ is optimal in the sense that no ellipse enclosing all points of M attains a smaller value for (4.3).*

Proof. By the assumption, there exist $\gamma > 0$ and $\epsilon \in \mathbb{C}$ such that the ellipse $\tilde{\mathcal{E}} \in \mathcal{F}_\gamma^\epsilon$ which is contained in the right half-plane and encloses all points in M. We can assume that at least one point $\widetilde{\lambda} \in M$ lies on the boundary of this ellipse. For all $\lambda \in M$ there exists $\mathcal{E}_\lambda \in \mathcal{F}_\gamma^\epsilon$ with $\lambda \in \mathcal{E}_\lambda$ given by (cf. Lemma A.2),

$$\mathcal{E}_\lambda = \begin{cases} \{\epsilon \cosh\omega_\lambda \cos\psi_\lambda + \mathrm{i}\,\epsilon \sinh\omega_\lambda \sin\psi_\lambda + \gamma \quad | \, \omega_\lambda \in \mathbb{R}_{\geq 0}, \psi_\lambda \in [0,2\pi]\}, & \text{if } \epsilon \in \mathbb{R}, \\ \{|\epsilon| \sin\omega_\lambda \cos\psi_\lambda + \mathrm{i}\,|\epsilon| \cosh\omega_\lambda \sin\psi_\lambda + \gamma \mid \omega_\lambda \in \mathbb{R}_{\geq 0}, \psi_\lambda \in [0,2\pi]\}, & \text{if } \epsilon \in \mathrm{i}\,\mathbb{R}. \end{cases}$$

Since all λ are contained in $\tilde{\mathcal{E}} = \mathcal{E}_{\widetilde{\lambda}}$, it holds that $\omega_\lambda \leq \omega_{\widetilde{\lambda}}$. Therefore, all $\mathcal{E}_\lambda$ are enclosed by $\widetilde{\mathcal{E}}$. By (4.2) it follows that

$$s(\epsilon,\gamma,\lambda) \leq s(\epsilon,\gamma,\widetilde{\lambda}) < 1$$

for all $\lambda \in M$. Then, for $\mathcal{E}_{\gamma^\star}^{\alpha^\star,\beta^\star}$ with eccentricity $\epsilon^\star$ and center $\gamma^\star$ and $\lambda^\star \in M$ with $\lambda^\star \in \mathcal{E}_{\gamma^\star}^{\alpha^\star,\beta^\star}$ it holds

$$s(\epsilon^\star,\gamma^\star,\lambda) \leq s(\epsilon^\star,\gamma^\star,\lambda^\star) \leq s(\epsilon,\gamma,\lambda^\star) < 1 \tag{4.5}$$

for all $\lambda \in M$. Using (4.2) again, $\mathcal{E}_{\gamma^\star}^{\alpha^\star,\beta^\star}$ enclosed all $\lambda \in M$ by the first inequality in (4.5) and $\mathcal{E}_{\gamma^\star}^{\alpha^\star,\beta^\star}$ lies in the right half-plane.

Finally, for any ellipse $\widehat{\mathcal{E}} \in \mathcal{F}_\gamma^\epsilon$ with $\widehat{\lambda} \in \widehat{\mathcal{E}}$ that encloses M, we have

$$s(\epsilon^\star,\gamma^\star,\lambda^\star) \leq s(\epsilon,\gamma,\lambda^\star) \leq s(\epsilon,\gamma,\widehat{\lambda}).$$

Since $\widehat{\mathcal{E}}$ was arbitrary $\mathcal{E}_{\gamma^\star}^{\alpha^\star,\beta^\star}$ is the ellipse with the smallest value (4.3) that enclose M. □

In [34] it was shown that there exists a solution to the min–max problem (4.4) on a compact set if we minimize with respect to

$$\widetilde{R} = \{(\epsilon,\gamma) \in R \mid \gamma > 0,\ \gamma^2 > \epsilon^2\},$$

(see also [33, Chp. 4] for more details).

4.1.3. Finding the ellipse

In the following, we present how to construct the *best-fitting* ellipses by explicitly solving the corresponding min-max problems. The calculations can be found in [33]. Because the ellipses are symmetric with respect to the real axis, λ is contained in the ellipse if and only if $\overline{\lambda}$ is also contained in the ellipse. Consequently, the min-max problem reduces to a minimization with respect to

$$M_+ := \{\lambda \in M \mid \mathrm{Im}\lambda \geq 0\}.$$

As described in [34], the following three cases must be considered.

Case 1: Exactly one point $\lambda^\star \in M_+$ is on the boundary of the best-fitting ellipse:
In this case, we have

$$\min_{(\epsilon,\gamma)\in R} \max_{\lambda\in M_+} s(\epsilon,\gamma,\lambda) = \min_{(\epsilon,\gamma)\in R} s(\epsilon,\gamma,\lambda^\star).$$

The *best-fitting* (degenerate) ellipse is

$$\alpha = 0, \quad \beta^2 = \mathrm{Im}^2\lambda^\star, \quad \gamma = \mathrm{Re}\lambda^\star.$$

(cf. [34]). This means that all points in M have the same real part and $\lambda^\star$ is the point with the largest imaginary part.

Case 2: Exactly two points $\lambda_1, \lambda_2 \in M_+$ are on the boundary of the best-fitting ellipse:
In this case, we have

$$\min_{(\epsilon,\gamma)\in R} \max_{\lambda\in M_+} s(\epsilon,\gamma,\lambda) = \min_{(\epsilon,\gamma)\in R} s(\epsilon,\gamma,\lambda_1) = \min_{(\epsilon,\gamma)\in R} s(\epsilon,\gamma,\lambda_2).$$

We assume that $\mathrm{Re}\lambda_1 < \mathrm{Re}\lambda_2$. Further, we define

$$A = \frac{\mathrm{Re}\lambda_2 - \mathrm{Re}\lambda_1}{2}, \quad B = \frac{\mathrm{Re}\lambda_2 + \mathrm{Re}\lambda_1}{2}, \quad V = \frac{\mathrm{Im}\lambda_2 - \mathrm{Im}\lambda_1}{2}, \quad W = \frac{\mathrm{Im}\lambda_2 + \mathrm{Im}\lambda_1}{2}. \tag{4.6}$$

By the assumptions of this section, we have $A, B, W \geq 0$.

i) If $V = 0$, according to [34], we obtain

$$\gamma = B, \qquad \epsilon^2 = \frac{\alpha^2(\alpha^2 - (A^2 + W^2))}{\alpha^2 - A^2}, \tag{4.7}$$

by inserting λ_1 and λ_2 into equation

$$\frac{(\mathrm{Re}\lambda - \gamma)^2}{\alpha^2} + \frac{\mathrm{Im}^2\lambda}{\alpha^2 - \epsilon^2} = 1. \tag{4.8}$$

It remains to calculate α by minimizing the constraint function (4.1). By [34] we obtain that α^2 is the only real root of $q(z)$ with

$$q(z) = q_1 z^3 + q_2 z^2 + q_3 z + q_4,$$

where the coefficient q_i, $i = 1 \dots 4$ can be found in [34, eq.(4.8)].

ii) Else if $V \neq 0$, then using again (4.8) [34] obtained

$$\begin{aligned} \alpha^2 &= \Big(\gamma - \big(B - \tfrac{AW}{V}\big)\Big)\Big(\gamma - \big(B - \tfrac{AV}{W}\big)\Big), \\ \epsilon^2 &= \frac{\Big(\gamma - \big(B + \tfrac{VW}{A}\big)\Big)\Big(\gamma - \big(B - \tfrac{AW}{V}\big)\Big)\Big(\gamma - \big(B - \tfrac{AV}{W}\big)\Big)}{\gamma - B}. \end{aligned} \tag{4.9}$$

Note that there is a small typo in [34] in the last equation. It remains to minimize the constraint function (4.1) with respect to γ. This leads to

$$z = \gamma - B \text{ is the root of } q(z) \text{ in } \begin{cases} (0, A) & \text{for } V > 0, \\ (-A, 0) & \text{for } V < 0, \end{cases}$$

where

$$q(z) = q_1 z^5 + q_2 z^4 + q_3 z^3 + q_4 z^2 + q_5 z + q_6.$$

Again the coefficients can be found in [34, eq.(4.9)].

Case 3: Three or more pairs in M_+ are on the boundary
We start with the case where we only have three points λ_1, λ_2, and λ_3, where the construction from case 2 does not yield the *best-fitting* ellipse. In this case, we have

$$\min_{(\epsilon,\gamma)\in R} \max_{\lambda\in M_+} s(\epsilon,\gamma,\lambda) = \min_{(\epsilon,\gamma)\in R} s(\epsilon,\gamma,\lambda_1) = \min_{(\epsilon,\gamma)\in R} s(\epsilon,\gamma,\lambda_2) = \min_{(\epsilon,\gamma)\in R} s(\epsilon,\gamma,\lambda_3).$$

We assume that $\mathrm{Re}\lambda_1 < \mathrm{Re}\lambda_2 < \mathrm{Re}\lambda_3$ and

$$(\mathrm{Re}\lambda_2 - \mathrm{Re}\lambda_1)(\mathrm{Im}\lambda_3^2 - \mathrm{Im}\lambda_1^2) < (\mathrm{Re}\lambda_3 - \mathrm{Re}\lambda_1)(\mathrm{Im}\lambda_2^2 - \mathrm{Im}\lambda_1^2). \tag{4.10}$$

Condition (4.10) ensures that there exists an ellipse such that $\lambda_1, \lambda_2, \lambda_3$ lie on its boundary. By inserting the three points into equation (4.8) we obtain a system of three equations with three unknowns α, ϵ, γ. The details can be found in [34, eq. (4.11) and (4.12)]. In the general case with more than three pairs, where the construction from Case 2 does not yield the *best-fitting* ellipse, we must consider all combinations of three pairs of points that satisfy (4.10) and selecting the ellipse that minimizes (4.1). The algorithm for calculating the method for determining the *best-fitting* ellipse consists of the following steps (see also [34] for more details).

Algorithm 1 Best-fitting ellipse

Input: Set of points $M_+ = \{\lambda_i\}_i$ with $\mathrm{Im}\lambda_i \geq 0$
Output: ellipse parameters α, β, γ

Set $\mathcal{P}_n = \{\Pi \in \mathcal{P}(M_+) \mid |\Pi| = n\}$ for $n = 2, 3$ with $\mathcal{P}(M_+) = \{U \mid U \subset M_+\}$
if All pairs lie on a vertical line **then**
    Set $\lambda = \mathrm{argmax}_{\lambda\in M_+} |\mathrm{Im}\lambda|$
    Calculate α, β, γ by Case 1 w.r.t λ
    return α, β, γ
end if
for $\{\lambda_i, \lambda_j\} \in \mathcal{P}_2$ **do**
    Calculate α, β, γ by Case 2 w.r.t $\{\lambda_i, \lambda_j\}$
    if $\nexists \lambda \in M_+$ outside $\mathcal{E}_\gamma^{\alpha,\beta}$ **then**
        return α, β, γ
    end if
end for
Set $s_{best} = \infty$
for $\{\lambda_i, \lambda_j, \lambda_l\} \in \mathcal{P}_3$ **do**
    if (4.10) is true for $\{\lambda_i, \lambda_j, \lambda_l\}$ **then**
        Calculate $\alpha, \beta, \gamma, \epsilon$ by Case 3 w.r.t $\{\lambda_i, \lambda_j, \lambda_l\}$ and set $s_{tmp} = s(\epsilon, \gamma, \lambda_i)$
        if $\nexists \lambda \in M_+$ outside $\mathcal{E}_\gamma^{\alpha,\beta}$ **and** $s_{tmp} < s_{best}$ **then**
            $s_{best} = s_{tmp}$
            $\alpha^{best} = \alpha,\ \beta^{best} = \beta,\ \gamma^{best} = \gamma$
        end if
    end if
end for
return $\alpha^{best}, \beta^{best}, \gamma^{best}$

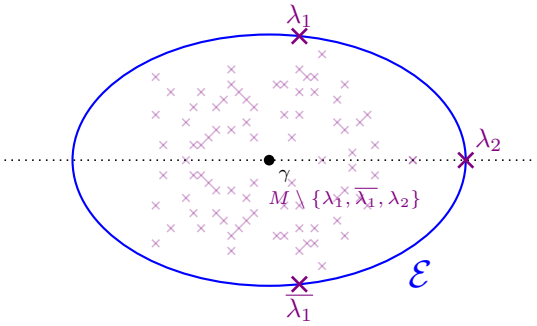


Figure 2: Best-fitting ellipse for a set M. For its calculation only λ_1 and λ_2 are required.

It is clear that, for the computation of a best-fitting ellipse for a set M, only the outer points are relevant (see Figure 2). In the following, we therefore derive a method for approximating the best-fitting ellipse of the set of eigenvalues by approximating the "outer" eigenvalues with suitable Ritz values.

4.2. Arnoldi method

The Arnoldi method is used to construct an orthonormal basis for the Krylov subspace

$$\mathcal{K}_m(\mathbf{S}, \nu) = \operatorname{span}\{\nu, \mathbf{S}\nu, \mathbf{S}^2\nu, \ldots, \mathbf{S}^{m-1}\nu\}.$$

The basis vectors $\{\nu_i\}_i$ of $\mathcal{K}_m(\mathbf{S}, \nu)$ are then stored columnwise in the matrix V_m. The eigenvalues of the projection matrix $H_m = \big((\nu_i, \mathbf{S}\nu_j)\big)_{i,j}$ are Ritz values (w.r.t. $\mathcal{K}_m(\mathbf{S}, \nu)$) and approximate the eigenvalues of the original matrix $\mathbf{S}$ (see, e.g., [10, 43]). Here, $(\cdot, \cdot)$ denotes the scalar product used in the method (see Section 2). We are interested in approximating eigenvalues, which lie on the outmost part of the spectrum (see Figure 2). The Arnoldi algorithm naturally approximates these eigenvalues (see, e.g. [43, p.156 ff]). However, the convergence speed of the Arnoldi method strongly depends on the distribution of the eigenvalues. If the eigenvalues are clustered, convergence can be very slow. In addition, we are looking for a stopping criterion that will allow us to determine whether we have found enough Ritz values for a good approximation of the best-fitting ellipse. We will address this issues in the next sections.

4.3. Transformation of the spectrum

We follow [40] to improve the convergence of the Arnoldi method. Suppose we are given an initial estimate of the best-fitting ellipse $\mathcal{E}$ that encloses all eigenvalues of $\mathbf{S}$ except for one eigenvalue λ. Our goal is to construct a polynomial p such that $|p(\lambda)|$ is large and $|p|$ is small on all other eigenvalues. We then apply the Arnoldi method to the polynomial $p(\mathbf{S})$.

By the Arnoldi method w.r.t. $p(\mathbf{S})$, we obtain a basis $\widehat{V}_m = \{\widehat{\nu}_i\}_i$ of the Krylov subspace $\mathcal{K}_m(p(\mathbf{S}), \nu)$ and an approximation of the corresponding eigenvalue of $\mathbf{S}$ from the projected matrix

$$G_m = \big((\widehat{\nu}_i, \mathbf{S}\widehat{\nu}_j)\big)_{i,j}, \tag{4.11}$$

to recompute $\mathcal{E}$. A widely used polynomial in this context is the following (see, e.g., [17, 25, 35, 40, 44]).

Definition 4.3. Let $\mathcal{E}_\gamma^{\alpha,\beta}$ be an ellipse and $z_0 \in \mathcal{E}_\gamma^{\alpha,\beta}$. For $\alpha \neq \beta$ and $z_0 \neq \gamma$ we define

$$r_\ell^{\alpha,\beta,\gamma}(z, z_0) := \frac{T_\ell(\iota(z))}{T_\ell(\iota(z_0))}, \qquad \iota(z) = \frac{z-\gamma}{\sqrt{\alpha^2-\beta^2}}. \tag{4.12}$$

We obtain by [40] the following relation for efficiently computing $r_\ell^{\alpha,\beta,\gamma}$:

$$\begin{aligned} &r_{\ell+1}^{\alpha,\beta,\gamma}(z, z_0) = 2\sigma_{\ell+1}\iota(z) r_\ell^{\alpha,\beta,\gamma}(z, z_0) - \sigma_\ell \sigma_{\ell+1} r_{\ell-1}^{\alpha,\beta,\gamma}(z, z_0), \quad \ell \geq 1, \\ &\sigma_{\ell+1} = \frac{1}{\iota(z_0) - \sigma_\ell}, \qquad \sigma_1 = \frac{1}{\iota(z_0)}, \qquad r_0^{\alpha,\beta,\gamma}(z, z_0) = 1, \qquad r_1^{\alpha,\beta,\gamma}(z, z_0) = \frac{z-\gamma}{z_0-\gamma}. \end{aligned} \tag{4.13}$$

The contour lines of $|r_\ell^{\alpha,\beta,\gamma}(z, z_0)|$ are confocal and concentric ellipses, in particular, the larger ellipses approach circles (see the discussion in [35, p. 307 ff.] and [34, p. 312 ff.]). The reference point z_0 in (4.12) determines the behavior of $r_\ell^{\alpha,\beta,\gamma}$, in particular whether $|r_\ell^{\alpha,\beta,\gamma}(z, z_0)|$ is large or small for $z \in W(\mathbf{S})$. This leads to the following result, (see again [35, p. 307 ff.]).

Lemma 4.4. *Let $\mathcal{E}_\gamma^{\alpha,\beta}$ be an ellipse with $z_0 \in \mathcal{E}_\gamma^{\alpha,\beta}$ and $\alpha, \beta > 0$. Then,*

$$\lim_{\ell\to\infty} \left| r_\ell^{\alpha,\beta,\gamma}(z, z_0) \right|^{1/\ell} \begin{cases} > 1 & \text{if } z \text{ is outside of } \mathcal{E}_\gamma^{\alpha,\beta}, \\ = 1 & \text{if } z \in \mathcal{E}_\gamma^{\alpha,\beta}, \\ < 1 & \text{if } z \text{ is inside of } \mathcal{E}_\gamma^{\alpha,\beta}, \end{cases}$$

for all $z \in \mathbb{C}$.

Remark 4.5. A similar result holds in the degenerate cases $\alpha = 0$ or $\beta = 0$, which means that z is outside of the degenerate ellipse if and only if $\iota(z) \notin [-1, 1]$. We choose z_0 such that $\iota(z_0) \in \{-1, 1\}$. Then

$$|r_\ell^{\alpha,\beta,\gamma}(z, z_0)| = |T_\ell(\iota(z))| \quad \text{for all } z \in \mathbb{C}.$$

It is well-known that $T_\ell(z) = \frac{1}{2}\big(\chi(z)^\ell + \chi(z)^{-\ell}\big)$, where χ is the (inverse) *Joukowski* transformation [31, p.86 ff], which maps the exterior of $[-1, 1]$ to the exterior of the unit disk. Consequently, $\lim_{\ell\to\infty} |r_\ell^{\alpha,\beta,\gamma}(z, z_0)|^{1/\ell} > 1$ if $\iota(z) \notin [-1, 1]$, whereas $|r_\ell^{\alpha,\beta,\gamma}(z, z_0)| \leq 1$ for all $\ell \in \mathbb{N}$ otherwise.

4.4. Find best-fitting ellipse for eigenvalues of S

Using the ideas from Section 4.3, we propose the following algorithm for calculating the best-fitting ellipse. We begin with an initial guess of the ellipse $\mathcal{E}_\gamma^{\alpha,\beta}$. This can be obtained by approximating a small set of eigenvalues, for example those with the largest magnitude, by standard methods. If eigenvalues with small magnitudes are expected, the origin can also be used to construct the ellipse.

Next, we apply the Arnoldi method to an initial vector ν and $r_\ell^{\alpha,\beta,\gamma}(\mathbf{S}, z)$, where $z \in \mathcal{E}_\gamma^{\alpha,\beta}$ and $\ell \in \mathbb{N}$ and calculate the Ritz values of this matrix. To assess the convergence of the Ritz values, we employ a standard residual-based error estimator [41, Proposition 6.8]. Specifically, at each iteration, we check whether the residual of the Ritz pair corresponding to the Ritz value of the largest magnitude is smaller than some tolerance. Additionally, a maximum number of iterations is enforced.

Once the Arnoldi process terminates, we check whether the ℓth root of norm of the Ritz value of the largest magnitude is less than one. If this condition is satisfied, the current ellipse is accepted as the best-fitting ellipse.

Otherwise, we calculate the Ritz values of $\mathbf{S}$ w.r.t. $\mathcal{K}_m(r_\ell^{\alpha,\beta,\gamma}(\mathbf{S}, z), \nu)$ by calculating the eigenvalues of G_m (see (4.11)) and determine whether they approximate eigenvalues of $\mathbf{S}$ (see Algorithm 3 line 8). If so, these are added to the current set of approximated eigenvalues, and a new

ellipse is constructed based on the updated set. If no Ritz value converges, we choose as new starting vector for the Arnoldi method the Ritz vector ν corresponding to the maximum Ritz value of this matrix with respect to the function

$$\varrho(\lambda) = \begin{cases} \frac{(\mathrm{Re}\lambda-\gamma)^2}{\alpha^2} + \frac{\mathrm{Im}\lambda^2}{\beta^2}, & \text{if } \alpha, \beta > 0, \\ \frac{(\mathrm{Re}\lambda-\gamma)^2}{\alpha^2}, & \text{if } \beta = 0 \text{ and } \mathrm{Im}\lambda = 0, \\ \frac{\mathrm{Im}\lambda^2}{\beta^2}, & \text{if } \alpha = 0 \text{ and } \mathrm{Re}\lambda = \gamma, \\ 2, & \text{else}, \end{cases} \tag{4.14}$$

which indicates how "far" a point λ lies outside the ellipse $\mathcal{E}_\gamma^{\alpha,\beta}$. If two Ritz values have the same value, we choose the one with the Ritz values of larger magnitude. Otherwise, if some Ritz values converge, we choose a random initial vector that is orthogonal to last converged Ritz vector with respect to $(\cdot,\cdot)$, see Algorithm 3, line 19. Since we consider real matrix $\mathbf{S}$, we can restart the Arnoldi algorithm with $\mathrm{Re}(\nu)$ (see [35, p. 303]). The algorithm is summarized in Algorithm 2 and 3.

Algorithm 2 Arnoldi method with spectral preconditioning

Input: Matrix $\mathbf{S} \in \mathbb{R}^{N\times N}$ and initial vector $\widehat{\nu}_1 \in \mathbb{R}^N$ with $\|\widehat{\nu}_1\| = 1$,
Point $z \in \mathbb{R}$, degree $\ell \in \mathbb{N}$, ellipse parameters α, β, γ
Maximum iterations $m_{\max} \in \mathbb{N}$, tolerance $\mathrm{TOL}_{\mathrm{res}} > 0$
Output: $\mu_{\max}$ and Ritz pairs (λ_i, w_i)

1: Initialize $R = 1$, $m = 1$
2: **while** $R > \mathrm{TOL}_{\mathrm{res}}$ **and** $m < m_{\max}$ **do**
3: Compute $\eta_m = r_\ell^{\alpha,\beta,\gamma}(\mathbf{S}, z)\widehat{\nu}_m$ by (4.13)
4: **for** $j = 1, \ldots, m$ **do**
5: $\widehat{h}_{j,m} = (\widehat{\nu}_j, \eta_m)$
6: **end for**
7: $v = \eta_m - \sum_{j=1}^m \widehat{h}_{j,m}\widehat{\nu}_j$
8: $\widehat{h}_{m+1,m} = \|v\|$
9: **if** $\widehat{h}_{m+1,m} = 0$ **then**
10: **break**
11: **end if**
12: $\widehat{\nu}_{m+1} = v/\widehat{h}_{m+1,m}$
13: Compute eigenpairs $(\mu_i^{(m)}, y_i^{(m)})$, $i = 1, \ldots, m$ of $\widehat{H}_m = (\widehat{h}_{i,j})_{i,j=1}^m$
14: Select $(\mu_{\max}^{(m)}, y_{\max}^{(m)})$ such that $|\mu_{\max}^{(m)}| = \max_i |\mu_i^{(m)}|$
15: $R = \widehat{h}_{m+1,m}\,|e_m^T y_{\max}^{(m)}|$ and $m \leftarrow m + 1$
16: **end while**
17: Form $\widehat{V}_m = (\widehat{\nu}_1, \ldots, \widehat{\nu}_m)$
18: Compute eigenpairs $(\lambda_i^{(m)}, x_i^{(m)})$, $i = 1, \ldots, m$ of $G_m = \left((\widehat{\nu}_i, \mathbf{S}\widehat{\nu}_j)\right)_{i,j}$
19: **return** $\mu_{\max}^{(m)}$ and $(\lambda_i^{(m)}, \widehat{V}_m x_i^{(m)})$, $i = 1, \ldots, m$

Remark 4.6. A more detailed discussion of implementation aspects, in particular the choice of the parameter ℓ_n in Algorithm 3, and additional strategies for improving the convergence speed can be found in Section 5.2.

Algorithm 3 Method for calculating the ellipse

Input: Matrix $\mathbf{S} \in \mathbb{R}^{N\times N}$ and initial vector $\nu_1 \in \mathbb{R}^N$ with $\|\nu_1\| = 1$,
degree $\ell_1 \in \mathbb{N}$, initial set M^1, maximum iterations $m_{\max}, n_{\max} \in \mathbb{N}$
Tolerance $\mathrm{TOL}_{\mathrm{eig}}, \mathrm{TOL}_{\mathrm{res}} > 0$, user-defined parameter $\mathrm{EPS} > 0$
Output: Ellipse $\mathcal{E}_\gamma^{\alpha,\beta}$

Compute $\mathcal{E}_{\gamma_1}^{\alpha_1,\beta_1}$ with M^1 and Algorithm 1
Set $z_1 = \max(\mathcal{E}_{\gamma_1}^{\alpha_1,\beta_1} \cap \mathbb{R})$ and $n = 1$
while $n < n_{\max}$ **do**
  $\mu_n, (\lambda_i^{(n)}, w_i^{(n)}) \leftarrow$ Algorithm 2 with $\left(\mathbf{S}, \nu_n, z_n, \ell_n, \alpha_n, \beta_n, \gamma_n, m_{max}, \mathrm{TOL}_{\mathrm{res}}\right)$
  if $\left|\mu_n\right|^{1/\ell_n} \leq 1$ **then**
    return $\mathcal{E}_{\gamma_n}^{\alpha_n,\beta_n}$
  end if
  for each $(\lambda_i^{(n)}, w_i^{(n)})$ **do**
    if $\|\lambda_i^{(n)} w_i^{(n)} - \mathbf{S} w_i^{(n)}\| \leq \mathrm{TOL}_{\mathrm{eig}} \cdot \|\mathbf{S} w_i^{(n)}\|$ **then**
      $M^{n+1} \leftarrow M^n \cup \{\lambda_i^{(n)}\}$
      $\omega = w_i^{(n)}$
    end if
  end for
  Compute $\mathcal{E}_{\gamma_{n+1}}^{\alpha_{n+1},\beta_{n+1}}$ with M^{n+1} and Algorithm 1
  if $\alpha_{n+1} = \beta_{n+1}$ **then**
    $\beta_{n+1} \leftarrow \beta_{n+1} + \mathrm{EPS}$
  end if
  if $|M^{n+1}| > |M^n|$ **then**
    ν_{n+1} is a random unit vector that is orthogonal to ω with respect to $(\cdot,\cdot)$
  else
    $\omega = w_i^{(n)}$ with $i = \mathrm{argmax}_i \, |\varrho(\lambda_i^{(n)})|$
    $\nu_{n+1} = \mathrm{Re}(\omega)/\|\omega\|$
  end if
  Determine new ℓ_{n+1} and set $z_{n+1} = \max(\mathcal{E}_{\gamma_{n+1}}^{\alpha_{n+1},\beta_{n+1}} \cap \mathbb{R})$ and $n \leftarrow n+1$
end while

5. Application for discrete semilinear damped wave equation

This section describes how the theory from the previous sections can be applied to systems of ordinary differential equations (ODEs) arising from space discretizations of semilinear damped wave equations. First, in Section 5.1, we introduce a second-order model problem. Rewriting this in first-order form leads to a system matrix $\mathbf{S}$, to which we apply the proposed methods to approximate the matrix φ_k-functions, as required when applying exponential integrators to the model problem. We focus here on the matrix exponential. In Section 5.2, we provide details of the implementation. Furthermore, we compare our method with other approaches described in Section 5.3 and present numerical experiments in Section 5.4.

5.1. Model problem

We consider the following semilinear damped wave equation in a domain $\Omega \subset \mathbb{R}^d, d = 1, 2, 3$ with Dirichlet boundary conditions

$$\begin{aligned} \partial_{tt} u(t,x) - a\Delta u(t,x) + \mathcal{B}\partial_t u(t,x) = G(u, \partial_t u(t,x)), \quad t > 0, \\ u(0) = u_0, \qquad \partial_t u(0) = v_0, \end{aligned} \tag{5.1}$$

where $a > 0$, $\mathcal{B}$ is a linear differential operator and G is nonlinear and locally Lipschitz continuous form $H^1(\Omega) \times L^2(\Omega) \to L^2(\Omega)$ and u_0, v_0 sufficiently smooth. We consider two different operators $\mathcal{B}$ in the following

$$\mathcal{B} = -\delta\Delta, \tag{5.2a}$$

$$\mathcal{B} = \delta b^T \cdot \nabla \tag{5.2b}$$

with $\delta > 0$ and $b(x) \in (C^1(\mathbb{R}^d))^d$. The operator (5.2a) appears in, e.g., viscoelasticity, structural mechanics, and thermoelasticity and represents strong/structural damping (cf. [6, 32]). On the other hand (5.2b) arises in aeroacoustics [5, 11, 13], and moving-medium wave propagation [2]. Discretizing (5.1) using linear finite elements yields a system of ODEs of the form

$$\begin{aligned} \mathbf{M}\boldsymbol{\mu}''(t) + \mathbf{A}\boldsymbol{\mu}(t) + \mathbf{B}\boldsymbol{\mu}'(t) = \mathbf{G}(\boldsymbol{\mu}(t), \boldsymbol{\mu}'(t)), \quad t > 0, \\ \boldsymbol{\mu}(0) = \boldsymbol{\mu}^0, \quad \boldsymbol{\mu}'(0) = \boldsymbol{\nu}^0, \end{aligned} \tag{5.3}$$

where $\mathbf{M}, \mathbf{A}, \mathbf{B} \in \mathbb{R}^{N\times N}$, $\mathbf{M}, \mathbf{A}$ are symmetric and positive definite, and $\mathbf{G} : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ is locally Lipschitz continuous w.r.t both arguments. Rewriting (5.3) in first-order formulation gives

$$\begin{aligned} \boldsymbol{x}' - \mathbf{S}\boldsymbol{x} = \mathbf{F}(\boldsymbol{x}), \\ \boldsymbol{x}(0) = \boldsymbol{x}^0, \end{aligned} \tag{5.4}$$

where

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\nu} \end{bmatrix}, \qquad \mathbf{S} = \begin{bmatrix} 0 & I \\ -\mathbf{M}^{-1}\mathbf{A} & -\mathbf{M}^{-1}\mathbf{B} \end{bmatrix},$$

$$\boldsymbol{x}^0 = \begin{bmatrix} \boldsymbol{\mu}^0 \\ \boldsymbol{\nu}^0 \end{bmatrix}, \qquad \mathbf{F}(\boldsymbol{x}) = \begin{bmatrix} 0 \\ \mathbf{M}^{-1}\mathbf{G}(\boldsymbol{\mu}, \boldsymbol{\nu}) \end{bmatrix}.$$

Note that if $\mathbf{B} = 0$ the matrix $\mathbf{S}$ is skew-symmetric w.r.t. the inner product

$$\langle \cdot, \cdot \rangle_{\mathbf{A}\times\mathbf{M}} = \left(\cdot, \begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{M} \end{bmatrix} \cdot \right), \tag{5.5}$$

where $\langle \cdot, \cdot \rangle$ denotes here the Euclidean inner product in $\mathbb{C}^{2N}$. This choice is natural, as it represents a discrete analogue of the $H^1 \times L^2$ inner product, which is commonly used in this setting [18, 26].

When multiplying with $\mathbf{S}$, we solve a linear system involving $\mathbf{M}$ using the CG method with a tolerance of $\mathrm{TOL}_{\mathrm{sol}} = 0.01 \cdot \mathrm{TOL}_R$, where TOL_R is a tolerance used for the error estimator (Section 3). This tolerance should be determined by the ODE solver, for example through local error estimators. We use an incomplete Cholesky factorization without fill-in (IC(0)) factorization as a preconditioner (cf. [42, Chp. 10]). Although the mass matrix $\mathbf{M}$ is generally well-conditioned, the frequent multiplications of $\mathbf{S}$, when applying our method, requires repeatedly solving such linear systems. It is therefore advantageous to compute the preconditioner once and reuse it throughout the computation. In our numerical experiments, we find that the number of CG iterations is reduced to only 1 or 2 when using IC(0) factorization, whereas without preconditioning it ranges from 5 to 20 depending on the chosen tolerance.

5.2. Implementation details of Algorithms 2 and 3

We now describe the details of the Algorithms proposed in Section 4.4. We begin by selecting a random initial vector ν_1 with $\|\nu_1\| = 1$, generated using a uniform distribution on $(0, 1)$. As the first set M^1, we choose $\{\lambda^\star, 0\}$, where $\lambda^\star$ is an approximation of the eigenvalue with the largest magnitude of $\mathbf{S}$, calculated with the C++ library *Spectra* https://spectralib.org/. Based on this set, the initial ellipse is constructed using Algorithm 1. We stop the algorithm if $\left|\mu_n\right|^{1/\ell_n} \leq 1 + \mathrm{TOL}_{\mathrm{stop}}$ (see Algorithm 3, line 4), where $\mathrm{TOL}_{\mathrm{stop}}$ is a small tolerance that ensures numerical errors in computation do not destabilize the algorithm.

In the following experiments we choose $\mathrm{TOL}_{\mathrm{stop}} = \mathrm{TOL}_{\mathrm{eig}} = 10^{-4}$. The initial polynomial degree is set to $\ell_0 = \sqrt{\lfloor \epsilon \rfloor}$, where ϵ denotes the linear eccentricity of the initial ellipse. Based on numerical experiments, this appears to be a good choice in many cases. $n_{\max}$ is set to 30.

Error estimator of Arnoldi method For the error estimator in the Arnoldi method, we use a moderate tolerance $\mathrm{TOL}_{\mathrm{res}} = 10^{-2}$ to keep the number of iterations low while still achieving good convergence. We set the maximum number of iterations to $m_{\max} = 200$.

Degree ℓ of $r_\ell(z, \cdot)$ We calculate the degree ℓ_n by the following procedure, which is inspired by [44, eq.(23),(24)]. In the first steps, we take into account that our ellipses may not be good, which means that there are eigenvalues that are likely to lie far outside. Therefore, we do not need a high polynomial degree, as these converge relatively quickly. This is because the convergence rate to an eigenvalue λ outside of the current ellipse $\mathcal{E}$ depends on the "gap" between $\mathcal{E}$ and an ellipse, which passes through λ and has the same center and linear eccentricity as $\mathcal{E}$ (see [35, Theorem 4.1]). A larger gap leads to better convergence. Later, the polynomial degree must be increased to ensure good convergence of eigenvalues that are not so far outside. We therefore choose

$$\ell_{n+1} = \max\Big\{ \Big| \frac{\ln(\mathrm{TOL}_{\mathrm{stop}})}{2\ln(\mu_n)} \Big| + 10, \ell_n \Big\},$$

where μ_n is computed in Algorithm 2 line 4. Close to convergence, $\ln(\mu_n)$ can become very small, causing ℓ_n to grow unnecessarily large. For this reason, we propose that if $\mu_n - 1 < 100 \cdot \mathrm{TOL}_{\mathrm{stop}}$ then

$$\ell_{n+1} = 1 + \Big(0.1 \Big| \ln\Big(\frac{\mu_n}{1 + \mathrm{TOL}_{\mathrm{stop}}}\Big) \Big| + 1\Big)\ell_n,$$

which ensures moderate growth while avoiding an excessive number of matrix-vector multiplications. We also set a maximum degree $\ell_{\max}$ to prevent excessive computational effort. In the

numerical experiments, we choose $\ell_{\max} = 1000$.

Further strategies To improve the convergence, we propose several strategies. Firstly, if the maximum degree $\ell_{\max}$ is reached and the algorithm has not yet converged, we reduce the tolerance $\mathrm{TOL}_{\mathrm{res}}$ by the factor 0.1.

In practice, the ellipse can become quite large when the eigenvalues lie close to the imaginary axis. To avoid this, we shift all computed eigenvalues by 1. After completing the calculation, we shift the resulting ellipse back.

Since numerical instabilities can arise when the semi-axes have nearly equal lengths, we slightly increase one of the semi-axes in this case. In the numerical experiments, we increase β by 10^{-9} if $\left|\alpha - \beta\right| < 10^{-10}$.

5.3. Krylov subspace methods

We compare our method with two classical approaches for computing matrix functions: the standard Krylov subspace method and the shift-and-invert method with shift $s \in \mathbb{R}$.

Standard Krylov subspace method Similar as in Section 4.2 we can use the Arnoldi method to construct an orthonormal basis $V_m \in \mathbb{R}^{2N\times m}$ (w.r.t. $(\cdot,\cdot)_{\mathbf{A}\times\mathbf{M}}$) and a projection matrix $H_m \in \mathbb{R}^{m\times m}$ of the Krylov subspace $\mathcal{K}_m(\mathbf{S}, \nu)$, such that

$$\mathbf{S}V_m = V_m H_m + h_{m+1,m}\nu_{m+1}e_m^T, \qquad V_m^T \begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{M} \end{bmatrix} V_m = I_m.$$

Here, $e_m = (0,\dots,0,1)^T \in \mathbb{R}^m$ and I_m denotes the $m \times m$ identity matrix. An approximation of the exponential matrix function is computed by

$$\exp(\tau\mathbf{S})\nu \approx \|\nu\|_{\mathbf{A}\times\mathbf{M}} V_m \exp(\tau H_m)e_1. \tag{5.6}$$

(see e.g., [7, eq.(2.2)]). By [7, eq.(2.5)] we get the following expression for the residual

$$|R_m(\tau)| = \|\nu\|_{\mathbf{A}\times\mathbf{M}} |h_{m+1,m} e_m^T \exp(\tau H_m)e_1| < \mathrm{TOL}_R, \tag{5.7}$$

which provides an error estimate controlled by a user-defined tolerance TOL_R.

Shift-and-Invert Arnoldi method Here the Krylov subspace is constructed with respect to the matrix $(I - s\mathbf{S})^{-1}$ with $s > 0$ being the shift parameter. Then, the Krylov relation reads

$$(I - s\mathbf{S})^{-1}V_m = V_m\widetilde{H}_m + \widetilde{h}_{m+1,m}\nu_{m+1}e_m^T. \tag{5.8}$$

An approximation of the matrix function is then calculated using (5.6), where H_m is defined by

$$H_m = \frac{1}{s}(I - \widetilde{H}_m^{-1}),$$

c.f. [48]. Using [7, eq.(2.10)] gives $\|R_m(\tau)\|_{\mathbf{A}\times\mathbf{M}} = |\beta_m(\tau)|\|w_{m+1}\|_{\mathbf{A}\times\mathbf{M}} < \mathrm{TOL}_R$ with

$$|\beta_m(\tau)| = \|v\|_{\mathbf{A}\times\mathbf{M}} \left|\frac{h_{m+1,m}}{s} e_m^T \widetilde{H}_m^{-1} \exp(\tau H_m)e_1\right|, \qquad w_{m+1} = (I - s\mathbf{S})\nu_{m+1},$$

which provides an error estimate controlled by a user-defined tolerance TOL_R. Note, that we never calculate the inverse $(I - s\mathbf{S})^{-1}$ explicitly. Instead we solve

$$(I - s\mathbf{S})x = y \Leftrightarrow \left(\begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{M} \end{bmatrix} - s\begin{bmatrix} 0 & \mathbf{A} \\ -\mathbf{A} & \mathbf{B} \end{bmatrix}\right)\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{M} \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

for $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ and $y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ for $x_1, x_2, y_1, y_2 \in \mathbb{R}^N$. This together with the Schur complement lead to following system of equations

$$\begin{aligned} y_1 &= x_1 + sy_2, \\ \mathbf{T}y_2 &= \mathbf{M}x_2 - s\mathbf{A}x_1 \end{aligned}$$

with $\mathbf{T} = \mathbf{M} + s^2\mathbf{A} + s\mathbf{B}$. We solve this system using the CG method with tolerance $\text{TOL}_{\text{sol}} = 0.01 \cdot \text{TOL}_R$ if $\mathbf{T}$ is symmetric. Otherwise, we use the GMRES method with the same tolerance TOL_{sol}. In the symmetric case, we use the IC(0) as a preconditioner, whereas in the nonsymmetric case, we employ the incomplete LU factorization without fill-in (ILU(0)) (cf. [42, Chp. 10]).

5.4. Numerical examples

In this section we present numerical experiments for the model problem (5.3) using the Chebyshev method from Section 2. All experiments are performed in the C++-library *deal.II* [1] and run on an Intel i5-12500K CPU with 32 GB RAM.

Example 1 In order to show how the Arnoldi method with spectral preconditioning (Algorithm 3) works in practice, we first consider a small-scale example. We set $\Omega = (0, 1)$ and $\mathcal{B}$ as in (5.2a) with $\delta \in \{0.01, 0.03, 0.4\}$ and $a = 0.5$. For the discretization we use an uniform grid with grid width $h = 2^{-4}$ which lead to $\mathbf{A}, \mathbf{M}, \mathbf{B} \in \mathbb{R}^{17\times 17}$.

We first compute the eigenvalues of the matrix $\mathbf{S}$ using a standard dense eigenvalue solver from LAPACK++ (`https://lapackpp.sourceforge.net/html/index.html`) and calculate the *best-fitting* ellipse $\mathcal{E}_{\text{best}}$ containing all eigenvalues using the algorithm from Section 4.1 for reference.

In Figure 3, we show $\mathcal{E}_{\text{best}}$ and the eigenvalues of $\mathbf{S}$ together with the result of the Arnoldi method with spectral preconditioning, which was applied to $\mathbf{S}$ for different δ. Note that the axes are scaled differently for better visualization.

The first Figure 3a shows the case, where the method performed several steps to find a best-fitting ellipse. Each detected eigenvalue is marked with a cross in the color corresponding to the step in which it was found. It can be seen that after a few steps, the method finds the best-fitting ellipse.

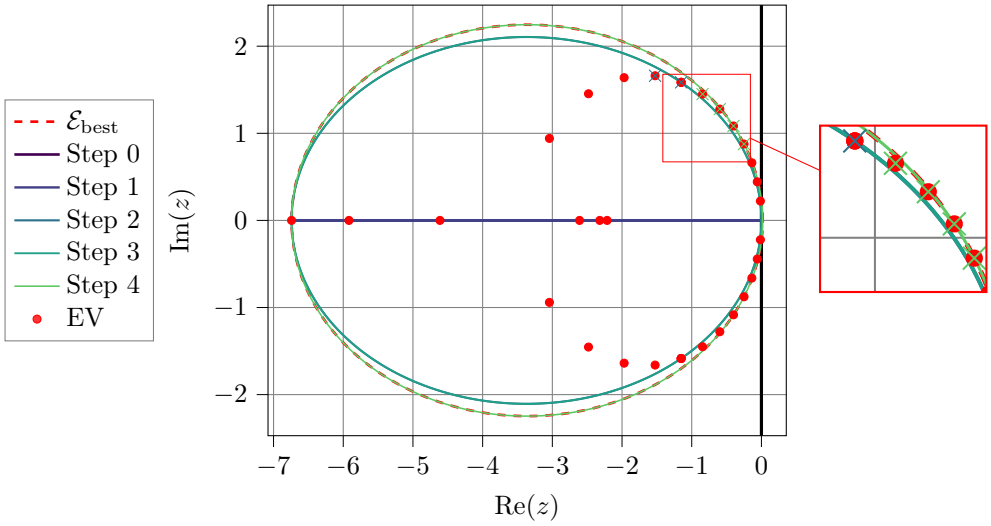
The second Figure 3b shows the case where the largest eigenvalue is sufficient to obtain the best-fitting ellipse, so that the algorithm converges in one step.

The last Figure 3c shows the case where there are eigenvalues that lie slightly outside the computed ellipse, as they were not found with the chosen tolerance and ℓ_0. Theoretically, this can lead to problems, but in practice we have not encountered any issues with the selected tolerance and initial degree, as the computed ellipse differs relatively little from the best-fitting ellipse.

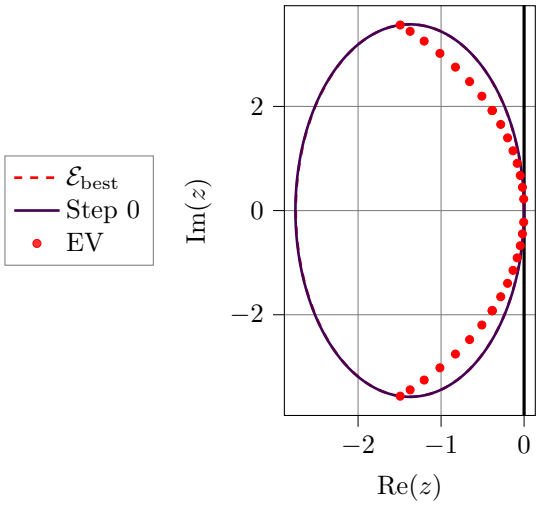
Example 2 We set $\Omega = (0, 1)^2$ and $\mathcal{B}$ with $\delta \in \{0.1, 0.01, 0.001\}$ and $a = 0.5$ as in (5.2a). We use a 2D-finite element discretization on a regular rectangular grid with grid width $h = 2^{-7}$, which lead to $\mathbf{A}, \mathbf{M}, \mathbf{B} \in \mathbb{R}^{N\times N}$ with $N = 16641$.

Further, we set ν_c as the coefficient vector that is obtained from the interpolation of the function $g(x, y) = \sin(\pi x^2)\sin(\pi y^2)$ on the grid. As initial vector we choose

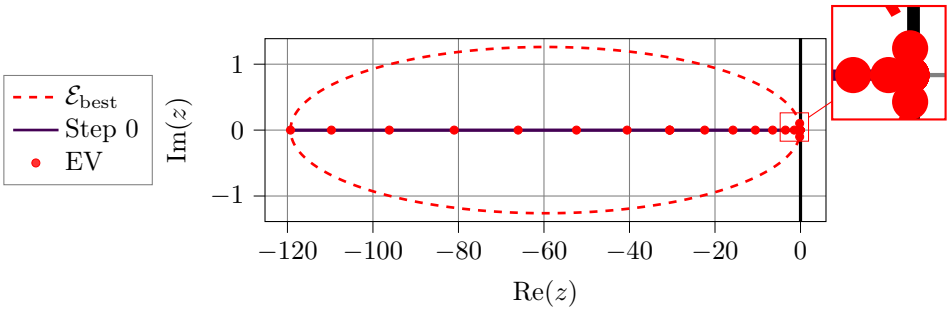
$$\nu = \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix} \quad \text{with} \quad \nu_1 = \nu_2 = \nu_c \in \mathbb{R}^N.$$



(a) $\delta = 0.03$



(b) $\delta = 0.01$



(c) $\delta = 0.4$

Figure 3: Eigenvalues of $\mathbf{S}$ in Example 1, best-fitting reference ellipse, and the calculated ellipse of the Arnoldi method with spectral preconditioning for each step and for different coefficients δ.

In the following we compute the matrix function $\exp(\tau\mathbf{S})\nu$ with $\tau = 0.1$. We compare the Chebyshev method with the standard Krylov subspace method and the shift-and-invert method. Since the computational cost is dominated by sparse matrix–vector multiplications, we measure the computational cost in terms of matrix–vector multiplications with $\mathbf{A}, \mathbf{B}, \mathbf{M}$ or similar matrices. In CG or GMRES with IC(0)/ILU(0) preconditioning, each iteration requires one matrix–vector multiplication, and one forward and one backward substitution with the triangular matrices obtained from the IC(0)/ILU(0) factorization. Since, for suitable sparse storage formats and memory layouts, the cost of these two substitutions is comparable to one sparse matrix–vector multiplications (see, e.g., [45] for ILU(0)), we approximate the cost of one preconditioned CG/GMRES iteration by two sparse matrix–vector multiplications.

In Figure 4, we examine the error with respect to the number of matrix-vector multiplications for different δ values and different tolerances TOL_R applied to the respective residual error estimators.

We calculate a reference solution using the standard Krylov subspace method for the matrix function with a very low tolerance 10^{-10} and the residual calculated by (5.7). Furthermore, we set $s = \frac{\tau}{2}$.

We observe that the Chebyshev method performs significantly better than the standard Krylov subspace method. Compared to the shift-and-invert method, the Chebyshev method is less efficient for relatively large tolerances. However, for smaller tolerances, the Chebyshev method benefits from the exponential decay of the error combined with the low computational cost per iteration, which considerably improves its efficiency compared to the shift-and-invert method. This advantage diminishes for larger δ (Figure 4b), so that the shift-and-invert method performs slightly better (Figure 4c) if δ is sufficiently large. This behavior can be explained by the fact that for small δ, the eigenvalues are clustered which is advantageous for the Chebyshev method because the ellipses stay relatively small (see Figure 3). For larger δ, the eigenvalues are more dispersed, which makes the advantage of an ellipse-based approach less significant.

Example 3 We use the same setting as in Example 2, but $\mathcal{B}$ with $\delta \in \{1.0, 0.1, 0.01\}$ and

$$b(x_1, x_2) = \begin{bmatrix} -x_1 \\ 0 \end{bmatrix}$$

is defined as in (5.2b).

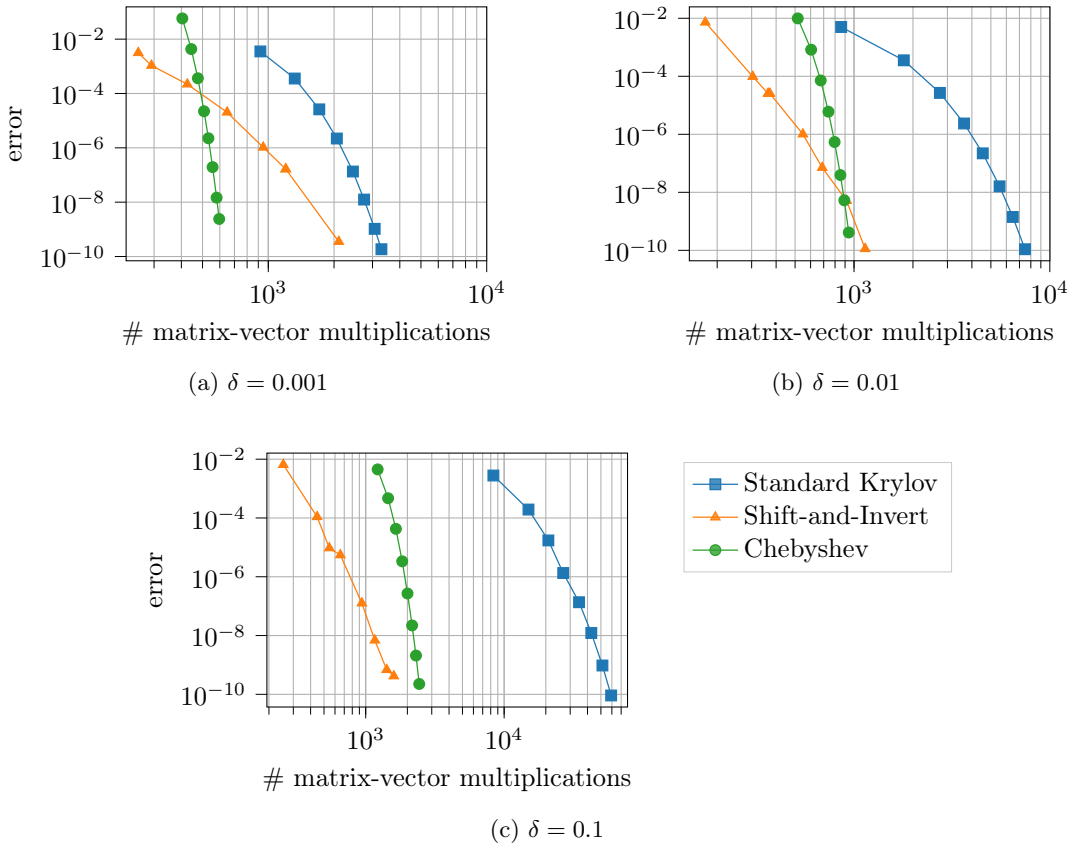
Again, we compute the matrix function $\exp(\tau\mathbf{S})\nu$ with $\tau = 0.1$. In Figure 5, we examine the error with respect to the number of matrix-vector multiplications for different δ values and different tolerances TOL_R applied to the respective residual error estimators. Again, we calculate a reference solution using the standard Krylov subspace method for the matrix function with a very low tolerance 10^{-10} and choose $s = \frac{\tau}{2}$.

We observe that the Chebyshev method performs significantly better than the standard Krylov subspace method. Compared to the shift-and-invert method, it can be seen, as in Example 2, that the shift-and-invert method is particularly efficient for relatively large tolerances, while the Chebyshev method is more advantageous for smaller tolerances.

Conflict of interest The authors declare that they have no conflict of interest to this work.

Data availability The codes to reproduce the experiments are available on https://github.com/danieleckhardt/chebychev

Declaration This manuscript is part of the PhD thesis of Daniel Eckhardt.



(a) $\delta = 0.001$

(b) $\delta = 0.01$

(c) $\delta = 0.1$

Figure 4: Approximation error in $\|\cdot\|_{\mathbf{A}\times\mathbf{M}}$ of $\exp(\tau\mathbf{S})v$ with $\tau = 0.1$ in Example 2 for different δ. The error is shown w.r.t. the number of matrix-vector multiplications, for $tol = 10^{-k}$, $k = 1, \dots 8$ as tolerance for the residual error estimators.

References

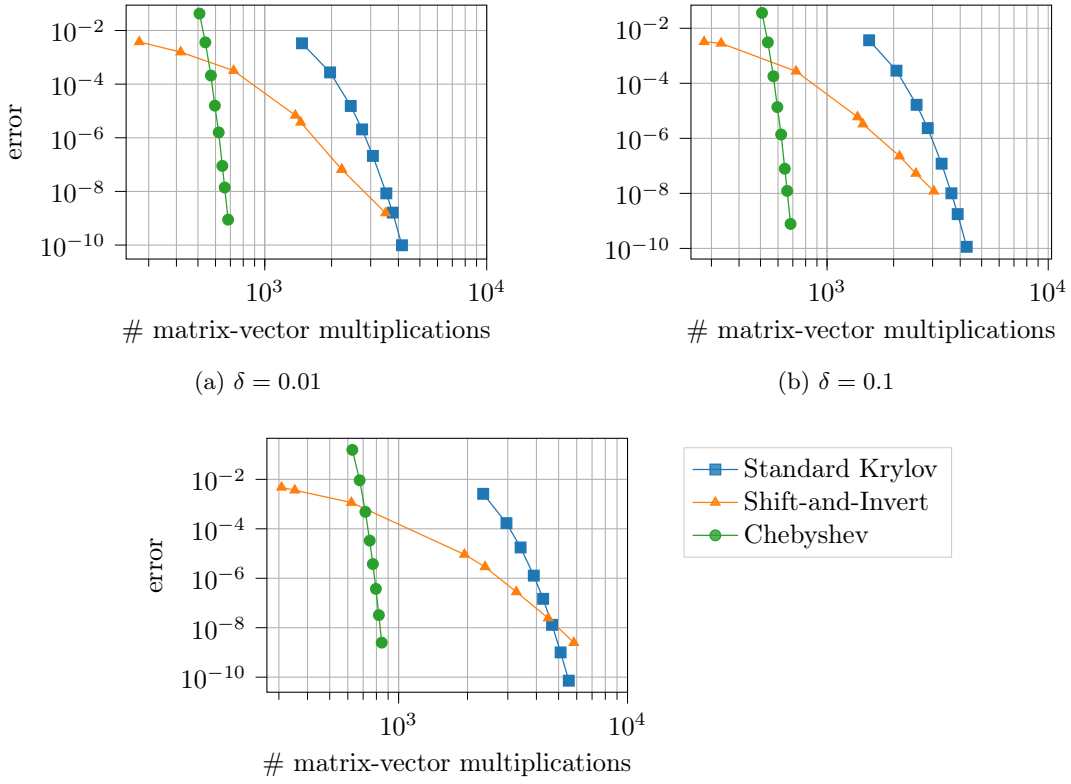
[1] D. Arndt, W. Bangerth, M. Bergbauer, B. Blais, M. Fehling, R. Gassmöller, T. Heister, L. Heltai, M. Kronbichler, M. Maier, P. Munch, S. Scheuerman, B. Turcksin, S. Uzunbajakau, D. Wells, and M. Wichrowski. The deal.ii library, version 9.7. *Journal of Numerical Mathematics*, 33(4):403–415, 2025.

[2] N. Banichuk, J. Jeronen, P. Neittaanmäki, T. Saksa, and T. Tuovinen. *Mechanics of Moving Materials*, volume 207 of *Solid Mechanics and Its Applications*. Springer International Publishing, 2014. First edition.

[3] B. Beckermann and L. Reichel. Error estimates and evaluation of matrix functions via the Faber transform. *SIAM J. Numer. Anal.*, 47(5):3849–3883, 2009.

[4] L. Bergamaschi and M. Vianello. Efficient computation of the exponential operator for large, sparse, symmetric matrices. *Numer. Linear Algebra Appl.*, 7(1):27–45, 2000.

[5] R. L. Bisplinghoff and H. Ashley. *Principles of aeroelasticity*. John Wiley & Sons, Inc., New York-London, 1962.



(a) $\delta = 0.01$

(b) $\delta = 0.1$

(c) $\delta = 1.0$

Figure 5: Approximation error in $\|\cdot\|_{\mathbf{A}\times\mathbf{M}}$ of $\exp(\tau\mathbf{S})v$ with $\tau = 0.1$ in Example 3 for different δ. The error is shown w.r.t. the number of matrix-vector multiplications, for $tol = 10^{-k}$, $k = 1, \dots 8$ as tolerance for the residual error estimators.

[6] E. Bonetti, E. Rocca, R. Scala, and G. Schimperna. On the strongly damped wave equation with constraint. *Comm. Partial Differential Equations*, 42(7):1042–1064, 2017.

[7] M. A. Botchev, V. Grimm, and M. Hochbruck. Residual, restarting, and Richardson iteration for the matrix exponential. *SIAM J. Sci. Comput.*, 35(3):A1376–A1397, 2013.

[8] M. Caliari, P. Kandolf, A. Ostermann, and S. Rainer. The Leja method revisited: backward error analysis for the matrix exponential. *SIAM J. Sci. Comput.*, 38(3):A1639–A1661, 2016.

[9] M. Caliari, M. Vianello, and L. Bergamaschi. Interpolating discrete advection-diffusion propagators at Leja sequences. *J. Comput. Appl. Math.*, 172(1):79–99, 2004.

[10] F. Chatelin. *Eigenvalues of matrices*, volume 71 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2012. With exercises by M. Ahués and the author, Translated with additional material by W. Ledermann, Revised reprint of the 1993 edition.

[11] I. Chueshov, I. Lasiecka, and J. Webster. Flow-plate interactions: well-posedness and long-time behavior. *Discrete Contin. Dyn. Syst. Ser. S*, 7(5):925–965, 2014.

[12] J. H. Curtiss. Faber polynomials and the Faber series. *Amer. Math. Monthly*, 78:577–596, 1971.

[13] E. H. Dowell, K. C. Hall, J. P. Thomas, and D. A. Dowell, editors. *A Modern Course in Aeroelasticity*, volume 293 of *Solid Mechanics and Its Applications*. Springer, Cham, 6 edition, 2022.

[14] V. Druskin and L. Knizhnerman. Krylov subspace approximation of eigenpairs and matrix functions in exact and computer arithmetic. *Numer. Linear Algebra Appl.*, 2(3):205–217, 1995.

[15] V. L. Druskin and L. A. Knizhnerman. Two polynomial methods for calculating functions of symmetric matrices. *Zh. Vychisl. Mat. i Mat. Fiz.*, 29(12):1763–1775, 1989.

[16] V. L. Druskin and L. A. Knizhnerman. Error estimates in the simple Lanczos process in the calculation of functions of symmetric matrices and eigenvalues. *Zh. Vychisl. Mat. i Mat. Fiz.*, 31(7):970–983, 1991.

[17] I. S. Duff and J. A. Scott. Computing selected eigenvalues of sparse unsymmetric matrices using subspace iteration. *ACM Trans. Math. Software*, 19(2):137–159, 1993.

[18] D. Eckhardt, M. Hochbruck, and B. Verfürth. Error analysis of an implicit–explicit time discretization scheme for semilinear wave equations with application to multiscale problems. *IMA Journal of Numerical Analysis*, 2025.

[19] M. Eiermann. On semiiterative methods generated by Faber polynomials. *Numer. Math.*, 56(2-3):139–156, 1989.

[20] G. Faber. Über polynomische Entwicklungen. *Math. Ann.*, 57:389–408, 1903.

[21] G. Faber. Über polynomische Entwicklungen II. *Math. Ann.*, 64(1):116–135, 1907.

[22] D. Gaier. *Lectures on complex approximation*. Birkhäuser Boston, Inc., Boston, MA, 1987.

[23] S. Güttel. Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection. *GAMM-Mitt.*, 36(1):8–31, 2013.

[24] N. J. Higham. *Functions of Matrices*. Society for Industrial and Applied Mathematics, 2008.

[25] D. Ho. Tchebychev acceleration technique for large scale nonsymmetric matrices. *Numer. Math.*, 56(7):721–734, 1990.

[26] M. Hochbruck and J. Leibold. An implicit-explicit time discretization scheme for second-order semilinear wave equations with application to dynamic boundary conditions. *Numer. Math.*, 147(4):869–899, 2021.

[27] M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 34(5):1911–1925, 1997.

[28] M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numer.*, 19:209–286, 2010.

[29] P. Kandolf, A. Ostermann, and S. Rainer. A residual based error estimate for Leja interpolation of matrix functions. *Linear Algebra Appl.*, 456:157–173, 2014.

[30] J. S. Kole. Solving seismic wave propagation in elastic media using the matrix exponential approach. *Wave Motion*, 38(4):279–293, 2003.

[31] C. Lubich. *From quantum to classical molecular dynamics: reduced models and numerical analysis*, volume 12. European Mathematical Society, 2008.

[32] J. Lubliner and P. Papadopoulos. *Introduction to Solid Mechanics*. Springer, Cham, 2nd edition, 2017.

[33] T. A. Manteuffel. *An Iterative Method for Solving Nonsymmetric Linear Systems with Dynamic Estimation of Parameters*. PhD thesis, University of Illinois at Urbana-Champaign, Ann Arbor, MI, 1975. Ph.D. thesis.

[34] T. A. Manteuffel. The Tchebychev iteration for nonsymmetric linear systems. *Numer. Math.*, 28(3):307–327, 1977.

[35] K. Meerbergen and C. Roose. Matrix transformations for computing rightmost eigenvalues of large sparse non-symmetric eigenvalue problems. *IMA J. Numer. Anal.*, 16(3):297–346, 1996.

[36] I. Moret and P. Novati. The computation of functions of matrices by truncated Faber series. *Numer. Funct. Anal. Optim.*, 22(5-6):697–719, 2001.

[37] O. Nevanlinna. *Convergence of iterations for linear equations*. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, 1993.

[38] J. Niesen and W. M. Wright. Algorithm 919: a Krylov subspace algorithm for evaluating the ϕ-functions appearing in exponential integrators. *ACM Trans. Math. Software*, 38(3):Art. 22, 19, 2012.

[39] F. V. Ravelo, P. S. Peixoto, and M. Schreiber. An explicit exponential integrator based on Faber polynomials and its application to seismic wave modeling. *J. Sci. Comput.*, 98(2):Paper No. 32, 39, 2024.

[40] Y. Saad. Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems. *Math. Comp.*, 42(166):567–588, 1984.

[41] Y. Saad. *Numerical methods for large eigenvalue problems*. Algorithms and Architectures for Advanced Scientific Computing. Manchester University Press, Manchester; Halsted Press [John Wiley & Sons, Inc.], New York, 1992.

[42] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.

[43] Y. Saad. *Numerical methods for large eigenvalue problems*, volume 66 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, revised edition, 2011.

[44] J. A. Scott. An Arnoldi code for computing selected eigenvalues of sparse, real, unsymmetric matrices. *ACM Trans. Math. Software*, 21(4):432–475, 1995.

[45] B. Smith and H. Zhang. Sparse triangular solves for ILU revisited: data layout crucial to better performance. *The International Journal of High Performance Computing Applications*, 25(4):386–391, 2011.

[46] P.K. Suetin. The basic properties of Faber poynomials. *Uspehi Mat. Nauk*, 19(4(118)):121–149, 1964.

[47] L. N. Trefethen, editor. *Spectral methods in MATLAB*, volume 10 of *Software, Environments, and Tools*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[48] J. van den Eshof and M. Hochbruck. Preconditioning Lanczos approximations to the matrix exponential. *SIAM J. Sci. Comput.*, 27(4):1438–1457, 2006.

[49] J. L. Walsh. *Approximation by polynomials in the complex domain*. Number 73 in Mémorial des sciences mathématiques. Gauthier-Villars, 1935.

A. Useful identities

Lemma A.1. *For $\theta_\psi^\omega = \psi - \mathrm{i}\,\omega$, $\psi \in [0, 2\pi]$ and $\omega \geq 0$ it holds that*

$$\cos(\theta_\psi^\omega) = \cosh\omega\cos\psi + \mathrm{i}\sinh\omega\sin\psi.$$

Proof.

$$\begin{aligned}\cos(\theta_\psi^\omega) &= \frac{1}{2}(\mathrm{e}^{\mathrm{i}\,\psi+\omega} + \mathrm{e}^{-\mathrm{i}\,\psi-\omega}) = \frac{1}{2}((\cos\psi + \mathrm{i}\sin\psi)\,\mathrm{e}^{\omega} + (\cos\psi - \mathrm{i}\sin\psi)\,\mathrm{e}^{-\omega}) \\ &= \cosh\omega\cos\psi + \mathrm{i}\sinh\omega\sin\psi.\end{aligned}$$

□

Lemma A.2. *Let*

$$z = \alpha\cos\psi + \mathrm{i}\,\beta\sin\psi + \gamma \in \mathcal{E}_\gamma^{\alpha,\beta}.$$

Then it holds that if $\alpha \neq \beta$

$$\frac{z-\gamma}{\epsilon} = \cos\theta_\psi^{\alpha,\beta} = \frac{1}{2}(\mathrm{e}^{\mathrm{i}\,\theta_\psi^{\alpha,\beta}} + \mathrm{e}^{-\mathrm{i}\,\theta_\psi^{\alpha,\beta}}),$$

where $\epsilon = \sqrt{\alpha^2 - \beta^2}$ and

$$\theta_\psi^{\alpha,\beta} = \psi - \mathrm{i}\log\Big(\sqrt{\frac{\alpha+\beta}{\alpha-\beta}}\Big) \qquad \text{if } \epsilon \in \mathbb{R}$$

$$\theta_\psi^{\alpha,\beta} = \psi - \frac{\pi}{2} - \mathrm{i}\log\Big(\sqrt{\frac{\beta+\alpha}{\beta-\alpha}}\Big) \qquad \text{if } \epsilon \in \mathrm{i}\,\mathbb{R}$$

with $\psi \in [-\pi, \pi]$.

Proof. Follows directly from Lemma A.1. □