Interpolating discrete advection–diffusion propagators at Leja sequences[☆]

M. Caliari[a], M. Vianello[b], L. Bergamaschi[c,*]

[a]Dip.to di Informatica, Università di Verona, Italy
[b]Dip.to di Matematica Pura ed Applicata, Università di Padova, Italy
[c]Dip.to di Metodi e Modelli Matematici per le Scienze Applicate, Università di Padova, via Belzoni 7, Padova 35131, Italy

Abstract

We propose and analyze the ReLPM (Real Leja Points Method) for evaluating the propagator $\varphi(\Delta tB)\boldsymbol{v}$ via matrix interpolation polynomials at spectral Leja sequences. Here B is the large, sparse, nonsymmetric matrix arising from stable 2D or 3D finite-difference discretization of linear advection–diffusion equations, and $\varphi(z)$ is the entire function $\varphi(z) = (e^z - 1)/z$. The corresponding stiff differential system $\dot{\boldsymbol{y}}(t) = B\boldsymbol{y}(t) + \boldsymbol{g}, \boldsymbol{y}(0) = \boldsymbol{y}_0$, is solved by the exact time marching scheme $\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \Delta t_i \varphi(\Delta t_i B)(B\boldsymbol{y}_i + \boldsymbol{g})$, $i = 0, 1, \ldots$, where the time-step is controlled simply via the variation percentage of the solution, and can be large. Numerical tests show substantial speed-ups (up to one order of magnitude) with respect to a classical variable step-size Crank–Nicolson solver.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Exponential operator; Advection–diffusion problem; Sparse matrix; Leja sequence; Polynomial interpolation

1. Introduction

Spatial discretization of linear autonomous advection–diffusion equations, like

$$\frac{\partial u}{\partial t}(x,t) = \Delta u(x,t) - \langle \theta, \nabla u(x,t) \rangle + g(x), \quad x \in \Omega, \ t > 0,$$

$$u(x,0) = u_0(x), \quad x \in \Omega,$$

$$u(x,t) = b(x), \quad x \in \partial\Omega, \tag{1}$$

where $\Omega \subset \mathbb{R}^p$ and $\theta = (\theta_i) \in \mathbb{R}^p$ ($p = 2, 3$), yields naturally large systems of ordinary differential equations of the form

$$\dot{y}(t) = By(t) + g, \quad t > 0,$$

$$y(0) = y_0, \tag{2}$$

where B is a sparse nonsymmetric $n \times n$ matrix: the solution of this system can be written as

$$y(t) = y_0 + t\varphi(tB)(By_0 + g), \tag{3}$$

where $\varphi(z)$ is the entire function

$$\varphi(z) = \frac{e^z - 1}{z} \tag{4}$$

leading to the numerical scheme

$$y_{i+1} = y_i + \Delta t_i \varphi(\Delta t_i B) v_i, \quad v_i = By_i + g, \ i = 0, 1, \dots \tag{5}$$

with $\Delta t_i = t_{i+1} - t_i$, which is *exact* for linear systems like (2) (cf. [12]). The solution of system (2) is best known in the form

$$y(t) = \exp(tB)(y_0 + B^{-1}g) - B^{-1}g$$

but with the use of the operator function φ, there is no linear system to solve.

Starting from the pioneering work in the 1980s, see, e.g., [6,9,35,37,38], in several papers explicit time integration schemes (exponential integrators) have been proposed for the solution of spatially discretized evolution problems like (2), which rest on the possibility of approximating efficiently the propagators $\exp(\Delta tB)v$ or $\varphi(\Delta tB)v$, where $v \in \mathbb{R}^n$ is a given vector, by *polynomial methods*; see, e.g., [2,3,9–12,23,24,26–28,34,35] and the recent survey paper [22]. All such schemes share the feature of being *explicit*, *exact* for linear autonomous problems, and superlinearly convergent in the approximation of the propagators, and thus represent an appealing alternative to classical finite-difference solvers. In particular, exactness entails that the time-steps Δt_i can be chosen, at least in principle, arbitrarily large with no loss of accuracy.

Polynomial methods can be subdivided in two classes. We have Krylov-like methods, which are based on the idea of projecting the propagator on a "small" Krylov subspace of the matrix via the Arnoldi process, and typically involve long-term recurrences in the nonsymmetric case; see, e.g., [11,34,36] and [29] for another (nonstandard) Krylov-like approach. The second class consists of methods based on polynomial interpolation or series expansion of the corresponding scalar analytic function on a suitable compact subset containing the spectrum (or in general an ε-pseudospectrum for some ε [39], or the field of values) of the matrix (e.g. Faber and Chebyshev series, interpolation at special points like Faber and Fejér points). They are conceived for approximating directly the matrix function, and typically require some initial estimate of the underlying (pseudo)spectral structure. For a survey and an extensive bibliography on polynomial methods for matrix functions, we refer to the Ph. D. thesis [26], which contains several original convergence estimates for the Faber and Chebyshev series methods and the Fejér points method. Despite the need of estimating (pseudo)

spectra, this second class of methods turned out to be competitive with Krylov-based approaches, especially on very large nonsymmetric problems, cf. [2,23–28].

In this paper, we propose and analyze the ReLPM (real Leja points method) for advection–diffusion problems (1)–(2), which rests on (pseudo)spectral estimates via families of confocal ellipses, and polynomial interpolation on *real* Leja sequences of the corresponding focal intervals. As known, (1) represents a simplified model, describing for example the transport of pollutants and suspended particles in shallow water. It is worth noting, however, that the ReLPM could be useful also for more complicated models, like nonautonomous advection–diffusion–reaction systems, for example within operator splitting approaches [13,15], or as building-block of nonlinear exponential integrators [12].

The paper is organized as follows. In Section 2, we briefly recall some basic results on the approximation of analytic functions by interpolation at Leja sequences, and their extension to matrix functions. In Section 3, we discuss the implementation of the ReLPM, with a special attention to the key feature concerning (pseudo)spectral estimates. In Section 4, finally, we present a wide set of tests on the numerical solution of 2D and 3D instances of the advection–diffusion problem (1)–(2). The exponential integrator obtained coupling the time marching scheme (5) and ReLPM, with a time stepping strategy based simply on the control of the variation percentage of the solution, shows substantial speed-ups (up to one order of magnitude) with respect to a classical variable step-size Crank–Nicolson solver.

2. Background on Leja points interpolation

In this section, we recall some basic classical results, concerning the approximation of analytic functions on compact sets of the complex plane based on polynomial interpolation, as well as their extension to the approximation of matrix functions.

Define $B[0, r] = \{w \in \mathbb{C} : |w| \leqslant r\}$ and $\bar{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$; if $K \subset \mathbb{C}$ is a compact set with more than one point, then there is a function $w = \phi(z)$ which maps $\bar{\mathbb{C}} \setminus K$ conformally onto $\bar{\mathbb{C}} \setminus B[0, 1]$ and satisfies the conditions

$$\phi(\infty) = \infty, \quad \lim_{z \to \infty} \frac{\phi(z)}{z} = \frac{1}{\gamma}, \quad \gamma = \mathrm{cap}(K), \tag{6}$$

where $\gamma > 0$ is called *capacity* of K (cf. [21, Theorem 3.14 and Collary]).

Sequences of Leja points $\{z_j\}_{j=0}^{\infty}$ for the compact K are defined recursively as follows [18]: if z_0 is an arbitrary fixed point in K (usually such as $|z_0| = \max_{z \in K} |z|$, cf. [32]), the z_j are chosen in such a way that

$$\prod_{k=0}^{j-1} |z_j - z_k| = \max_{z \in K} \prod_{k=0}^{j-1} |z - z_k|, \quad j = 1, 2, \dots . \tag{7}$$

By the maximum principle, the Leja points for K lie on ∂K.

For any $R > 0$, define now

$$\Gamma_R = \{z : |\phi(z)| = R/\gamma\}, \quad K_R \text{ the bounded domain with boundary } \Gamma_R, \tag{8}$$

observe that $K_\gamma = K$, the compact K_R has capacity R, and $K_{R_1} \subseteq K_{R_2}$ if $R_1 \leqslant R_2$. Let f be an analytic function on K: the extremal properties of Leja sequences make them suitable for polynomial

interpolation. In fact, it is well-known [32,42] that the sequence of polynomials p_m of degree m that interpolate f on the Leja points $z_0, z_1, \ldots, z_m$ for K converges maximally to f on K, that is

$$\limsup_{m \to \infty} \| f - p_m \|_K^{1/m} = \frac{\gamma}{R_{\max}} = \limsup_{m \to \infty} \| f - p_m^* \|_K^{1/m}, \tag{9}$$

where $\| \cdot \|_K$ denotes the maximum norm on K, $\{ p_m^* \}$ the sequence of best uniform approximation polynomials for f in K, and $R_{\max}$ is the largest constant such that f is analytic on the interior of $K_{R_{\max}}$. Observe that (9) implies superlinear convergence for entire functions ($R_{\max} = \infty$) and that it also holds if we replace γ and K with R and K_R, respectively, for any $R > 0$ such that K_R is defined (e.g. for any $\gamma \leqslant R < R_{\max}$, but possibly also for $R < \gamma$).

Moreover, Leja sequences are attractive for interpolation at high-degree, in view of the stability of the corresponding algorithm [32]. In fact, let $\boldsymbol{f}_m = (f(x_0), f(x_1), \ldots, f(x_m))$, where $\{ x_j \}_{j=0}^m$ are arbitrary points of K and p_m the interpolation polynomial in the Newton form of f at the $\{ x_j \}_{j=0}^m$. Then it is defined as map $T : \mathbb{C}^{m+1} \to \mathbb{P}_m$ with $T(\boldsymbol{f}_m) = p_m$. If we use the maximum norms

$$\| \boldsymbol{f}_m \|_\infty = \max_{0 \leqslant j \leqslant m} | f(x_j) |, \quad \| p_m \|_\infty = \max_{x \in K} | p_m(x) |,$$

the condition number for the operator T is $\mathrm{cond}(T) = \| T \| \cdot \| T^{-1} \|$, where $\| \cdot \|$ denotes the induced norm. For many sequence of interpolation points the condition number $\mathrm{cond}(T)$ grows exponentially with the number of points, while it can be proved (cf. [8,32]) that if $\{ x_j \}_{j=0}^m$ is a sequence of Leja points for K, then

$$\lim_{m \to \infty} (\mathrm{cond}(T))^{1/m} = 1.$$

From these properties, a stable and efficient polynomial approximation method for the matrix operator φ can be derived (cf. (3), (4)). In fact, we have in general

Proposition 1 (Novati [26, Section 3.1, Theorems 22 and 23]). *Let $A \in \mathbb{R}^{n \times n}$ and $\boldsymbol{v} \in \mathbb{R}^n$. If $\{ p_m \}$ converges maximally to f on K and if $\sigma(A) \subseteq K_R$ for some $R > 0$, then $\{ p_m(A)\boldsymbol{v} \}$ converges to $f(A)\boldsymbol{v}$. Moreover,*

$$\limsup_{m \to \infty} \| f(A)\boldsymbol{v} - p_m(A)\boldsymbol{v} \|_2^{1/m} \leqslant \frac{R}{R_{\max}}. \tag{10}$$

As in the scalar case, property (10) of maximal convergence of the sequence of matrix interpolation polynomials on Leja points can also be written as

$$\limsup_{m \to \infty} \| f(A)\boldsymbol{v} - p_m(A)\boldsymbol{v} \|_2^{1/m} = \limsup_{m \to \infty} \| f(A)\boldsymbol{v} - p_m^*(A)\boldsymbol{v} \|_2^{1/m}. \tag{11}$$

Observe that, when A is diagonalizable, $A = X^{-1}DX$, we have

$$\| f(A)\boldsymbol{v} - p_m(A)\boldsymbol{v} \|_2 \leqslant \mathrm{cond}_2(X) \cdot \| \boldsymbol{v} \|_2 \cdot \| f - p_m \|_{K_R}, \quad \sigma(A) \subseteq K_R,$$

which gives immediately (10) in view of (9).

In order to get more precise estimates on the rate of convergence of Leja points interpolation in the superlinear case (entire matrix functions, as $\exp(A)$ and $\varphi(A)$), we can resort to known results

for *Faber series*. In fact, the partial sums $F_m(f)$ of Faber series for analytic functions f provide another example of maximal convergence.

In the case when K is a convex compact subset of $\mathbb{C}$ with capacity γ, symmetric with respect to the real axis, for the exponential function it can be proved following [7,23] that:

$$\|\exp - F_m(\exp)\|_{K_R} \leqslant \frac{2}{e} \frac{1+\alpha}{\alpha} e^{d+\mathcal{O}(1/m)} \left(\frac{eR}{m}\right)^{m+1}, \quad m \geqslant (1+\alpha)R \tag{12}$$

for any $\alpha > 0$, where $\psi(w) = \gamma w + d + \mathcal{O}(1/w)$ is the inverse of ϕ in (6); see also [4]. We stress that symmetry required above is not restrictive in the application to functions of real matrices, whose (pseudo)spectra and fields of values have the same property (the latter being also convex).

Now, using the fact that $\varphi(z) = (e^z - 1)/z$ is $\mathcal{O}(1)$ for $\mathrm{Re}(z) \leqslant 0$ and $\mathcal{O}(\exp(z))$ for $\mathrm{Re}(z) > 0$, with a reasoning analogous to that used for (12) we can prove that

$$\|\varphi - F_m(\varphi)\|_{K_R} = \frac{1+\alpha}{\alpha} \|\varphi\|_{K_m} \mathcal{O}\left(\left(\frac{R}{m}\right)^{m+1}\right)$$

$$\cong \frac{1+\alpha}{\alpha} \mathcal{O}\left(\left(\frac{eR}{m}\right)^{m+1}\right), \quad m \geqslant (1+\alpha)R, \ \alpha > 0 \tag{13}$$

and thus, when A is diagonalizable, we can write

$$\|\varphi(A)\boldsymbol{v} - p_m(A)\boldsymbol{v}\|_2 \leqslant \mathrm{cond}_2(X)\|\boldsymbol{v}\|_2 \|\varphi - p_m\|_{K_R}$$

$$\approx \mathrm{cond}_2(X)\|\boldsymbol{v}\|_2 \|\varphi - F_m(\varphi)\|_{K_R}$$

$$= \mathrm{cond}_2(X) \frac{1+\alpha}{\alpha} \|\boldsymbol{v}\|_2 \mathcal{O}\left(\left(\frac{eR}{m}\right)^{m+1}\right), \quad m \geqslant (1+\alpha)R, \ \alpha > 0, \tag{14}$$

where $\sigma(A) \subseteq K_R$ and $\{p_m\}$ is any sequence of polynomials converging maximally to φ on K_R, or equivalently on K_R for any admissible R (in particular a sequence of interpolation polynomials on Leja points of K). Here and below, it is intended that the constants of the $\mathcal{O}$-symbols do not depend on the parameters displayed on the left.

For a general real matrix A, we can resort to the convergence estimates obtained in [26], using the *field of values* (or numerical range) of A, $W(A) = \{\langle \boldsymbol{u}, A\boldsymbol{u}\rangle/\|\boldsymbol{u}\|_2^2\}, \ \boldsymbol{u} \in \mathbb{C}^n \setminus \boldsymbol{0}\} \supseteq \mathrm{co}(\sigma(A))$ (the convex hull of $\sigma(A)$). By a slight modification in the proof of [26, Theorem 28], we obtain

$$\|\varphi(A)\boldsymbol{v} - F_m(A)\boldsymbol{v}\|_2 = \frac{1+\alpha}{\alpha} \|\boldsymbol{v}\|_2 \|\varphi\|_{K_{m+1}} (2m+3) \cdot \mathcal{O}\left(\left(\frac{R}{m}\right)^{m+1}\right), \quad m \geqslant (1+\alpha)R,$$

where $\alpha > 0$ is arbitrary and $W(A) \subseteq K_R$, and thus, in view of (10)–(11) and of the considerations above

$$\|\varphi(A)\boldsymbol{v} - p_m(A)\boldsymbol{v}\|_2 \approx \|\varphi(A)\boldsymbol{v} - F_m(A)\boldsymbol{v}\|_2$$

$$= \frac{1+\alpha}{\alpha} R\|\boldsymbol{v}\|_2 \mathcal{O}\left(\left(\frac{eR}{m}\right)^m\right), \quad m \geqslant (1+\alpha)R, \ \alpha > 0, \tag{15}$$

where again $\{p_m\}$ is any sequence converging maximally to φ on K_R. It is worth noting that the loss of a unit in the exponent (compare (15) and (14)), is intrinsic in the estimating technique, cf. [26, Theorem 28].

An "intermediate" convergence estimate can be easily obtained by resorting to the notion of *pseudospectrum* of the matrix A. Recall that the ε-pseudospectrum of A is defined as $\sigma_\varepsilon(A) = \{z \in \mathbb{C}: \|(zI - A)^{-1}\|_2 \geqslant \varepsilon^{-1}\}$, $\varepsilon > 0$, or equivalently $\sigma_\varepsilon(A) = \{z \in \sigma(A + \mathscr{E}), \text{ for some } \mathscr{E} \text{ with } \|\mathscr{E}\|_2 \leqslant \varepsilon\}$. As known, the spectrum and the field of values can be recovered as special cases from the limits $\varepsilon \to 0$ and (after peeling away an ε-border region) $\varepsilon \to \infty$, respectively; cf. the survey paper [39]. Now, in view of the following estimate which is given immediately by the Cauchy integral representation of the entire matrix function $\varphi(A) - F_m(A)$

$$\|\varphi(A) - F_m(A)\|_2 \leqslant \frac{1}{2\pi} \|\varphi - F_m(\varphi)\|_{K_R} \int_{\partial K_R} \|(zI - A)^{-1}\|_2 |\mathrm{d}z|$$

$$\leqslant \frac{\ell(\partial K_R)}{2\pi\varepsilon} \|\varphi - F_m(\varphi)\|_{K_R},$$

where $\sigma_\varepsilon(A) \subseteq K_R$, we get finally from (13)

$$\|\varphi(A)\boldsymbol{v} - p_m(A)\boldsymbol{v}\|_2 \approx \|\varphi(A)\boldsymbol{v} - F_m(A)\boldsymbol{v}\|_2$$

$$= \frac{1+\alpha}{\alpha} \|\boldsymbol{v}\|_2 \frac{\ell(\partial K_R)}{\varepsilon} \mathcal{O}\left(\left(\frac{eR}{m}\right)^{m+1}\right), \quad m \geqslant (1+\alpha)R, \ \alpha, \varepsilon > 0, \quad (16)$$

where once again $\{p_m\}$ is any polynomial sequence converging maximally to φ on K_R.

It is worth noting that estimates (14)–(16) are optimized in the cases when R is equal, respectively, to

$$R_\sigma = \inf\{\rho : \sigma(A) \subseteq K_\rho\},$$

$$R_w = \inf\{\rho : W(A) \subset K_\rho\},$$

$$R_\varepsilon = \inf\{\rho : \sigma_\varepsilon(A) \subseteq K_\rho\}. \tag{17}$$

3. Numerical implementation of the ReLPM interpolation method

In the sequel, the compact subsets used for estimating some pseudospectrum or the field of values of the matrix B in (2), and thus to construct the Leja points polynomial interpolation for the matrix operator

$$\varphi(A)\boldsymbol{v}, \quad A = \Delta t B$$

(see (3)) will be families of confocal ellipses. There are two main reasons for this choice.

For advection–diffusion matrices arising from stable FD discretizations (like e.g. central differences for grid-Peclet numbers strictly smaller than 1, or upwind method for Peclet numbers greater than 1, cf. [30]), the convex hulls of the numerically evaluated spectra turn out to be (possibly degenerating)
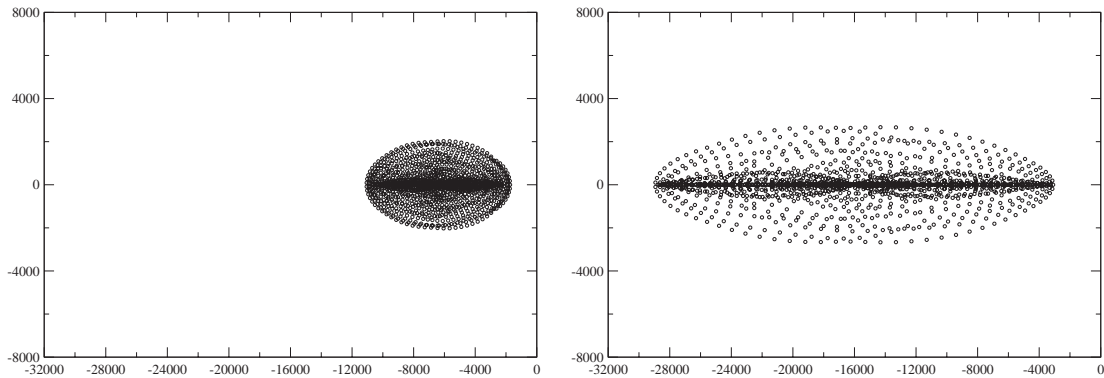
Fig. 1. Distribution of computed eigenvalues (pseudoeigenvalues) for advection–diffusion 2D matrices of dimension $n=1521$ with $\theta_1=60$, $\theta_2=60$ (Peclet=0.75), central discretization (left), and with $n=1521$, $\theta_1=120$, $\theta_2=120$ (Peclet=1.5), upwind discretization (right).

elliptical regions, with horizontal major axis; see Fig. 1 for an example (here the complete spectra have been computed by the subroutine dgeev of LA-PACK). This is not surprising, even though it is known, for example, that spectra of advection–diffusion matrices generated by central differences with grid-Peclet numbers < 1 are real (cf. [20]). In fact, due to strong nonnormality, the eigenvalue problem is ill-conditioned, and what we are really computing are pseudoeigenvalues of the matrix, i.e. a discrete ε-pseudospectrum where ε is related to the prescribed tolerance. It has been indeed already observed that pseudospectra of constant coefficients advection–diffusion operators are elliptical, and that this has to do with the underlying Toeplitz structure; see [31,33]. It is worth observing here that in our previous paper [2], some considerations made speaking of eigenvalues and spectra should be more properly intended in the sense of the corresponding pseudonotions. In particular, the figures showing ellipses are there also related to the computation of pseudoeigenvalues. Clearly this is not a problem as for convergence and superlinear convergence estimates, since we have seen in the previous section that there are clean convergence results also dealing with pseudospectra or with the field of values.

We now observe that if $K=E$ is the closure of the internal part of an ellipse of real center d and real foci $d-c$ and $d+c$, then its capacity is the half sum of the semi-axes of the ellipse; moreover, the family $\{K_R\}=\{E_R\}$ in (8) is a family of confocal ellipses. In particular, the ellipse of capacity $c/2$ degenerates into the segment $E_{c/2}=[d-c,d+c]$. Note that we restrict our attention to ellipses symmetric with respect to the real axis since in our application to advection–diffusion equations we deal with real matrices, whose spectrum, pseudospectra and field of values have the same property: in particular, the focal segment $[d-c,d+c]$ is a real interval.

The second reason for the choice of families of confocal ellipses follows immediately: if the spectrum $\sigma(A)$ is contained in E_R for some R, then, by Proposition 1, a sequence of polynomials converging maximally to φ on $E_{c/2} \subset \mathbb{R}$ will give a sequence of matrix polynomial operators converging (maximally) to $\varphi(A)\mathbf{v}$. The same applies to the convergence estimates (14)–(16), i.e. if the sequence $\{p_m\}$ converges maximally to φ on $E_{c/2} \subset \mathbb{R}$, then (14)–(16) hold on E_{R_σ}, E_{R_w} or E_{R_ε}, respectively (cf. (17)). Thus we are entitled to interpolate on Leja points of the focal interval

$[d - c, d + c] \subset \mathbb{R}$, working with real instead of complex arithmetic (as it would be required interpolating directly on the complex Leja points of some ellipse of the family).

We are now ready to describe our approximation procedure for advection–diffusion propagators. In view of the shape of the pseudospectra, the first step consists in evaluating the focal interval of the family of ellipses which will estimate such pseudospectra (and eventually the field of values); recall also, e.g., the classical papers [19,20] on ellipses-based "spectral" estimates for nonsymmetric matrices. To this aim, we could estimate by driver `dndrv1` (which does not solve linear systems) of ARPACK (ARnoldi PACKage, cf. [17]) the extreme eigenvalues of B. As observed above, in practice this routine provides a number of ε-pseudoeigenvalues, for a certain ε. Then, following [2], we could approximate the convex hull of the corresponding discrete pseudospectrum of $A = \Delta tB$ by the rectangle constructed using such (scaled) pseudoeigenvalues and select, among the ellipses circumscribing this rectangle, that, say $E = E_\gamma$, of smallest capacity γ, center d and foci $d + c$ and $d - c$ (for a discussion about this choice, cf. [2]). In practice, however, even a rough approximation of the focal interval $[d - c, d + c]$, given by the extreme real-parts of the spectrum estimated by Gershgorin's theorem (cf. also [35]), provides a polynomial approximation with essentially the same behavior (cf. Table 2). We stress that this approach, differently from that adopted in [2], makes completely negligible the preprocessing computational cost of extreme eigenvalue estimation. At this point, interpolating on Leja points of the focal interval, we obtain superlinear convergence of the corresponding matrix polynomial operators to the matrix operator φ; see (14)–(16).

An algorithm for the approximation of the advection–diffusion propagator $\varphi(\Delta tB)v$ can be now easily developed, by means of Newton interpolation at "spectral" Leja points: we sketch the algorithm in Table 1, and comment on it below.

Table 1
Algorithm ReLPM (Real Leja Points Method)

- INPUT: $B, v, \Delta t$, tol
- Estimate the spectral focal interval $[d - c, d + c]$ for $A = \Delta tB$, via Arnoldi approximation of extreme eigenvalues or by Gershgorin's theorem (see the discussion above)
- Let $\{x_j\}_{j=0}^N$ be a set of uniformly distributed points on $[d - c, d + c]$ for N sufficiently large (say $N = 10\,000$) take ξ_0 such that $|\xi_0| = \max_j |x_j| = \max\{|d - c|, |d + c|\}$, $d_0 := \varphi(\xi_0)$
- $A := \Delta tB$, $w_0 := v$, $p_0 := d_0\,w_0$, $m := 0$
- DO WHILE $e_m^{\text{Leja}} := |d_m| \cdot \|w_m\|_2 > \text{tol}$
  - $w_{m+1} := (A - \xi_m I)w_m$
  - $m := m + 1$
  - Compute ξ_m such that $\prod_{k=0}^{m-1} |\xi_m - \xi_k| = \max_j \prod_{k=0}^{m-1} |x_j - \xi_k|$ (avoiding overflow problems by scaling to capacity 1, as described in [32])
- $d_m := \varphi(\xi_m)$
- compute the next divided difference d_m as:
  DO $i = 1, m$
  $\quad d_m := (d_m - d_{i-1})/(\xi_m - \xi_{i-1})$
  END DO
- $p_m := p_{m-1} + d_m w_m$
END DO
- OUTPUT: the vector $p_m : \|p_m - \varphi(\Delta tB)v\|_2 \approx e_m^{\text{Leja}} \leqslant \text{tol}$

3.1. Comments on the implementation

The ReLPM algorithm is quite simple and efficient. Indeed, being based on *two-term* vector recurrences in *real arithmetic*, its storage occupancy and computational cost are very small. For $m \ll n$ (as in our application) it requires essentially only $(v + 3)n$ floats for the matrix and three vectors, and costs $\approx (2v + 3)mn$ flops (where v is the average number on nonzeros per row in the matrix).

The asymptotic performance of the ReLPM is that of the Chebyshev series method studied in [2] (in view of maximal convergence, see Section 2). For advection–diffusion matrices arising from stable spatial discretizations, the ReLPM turns out to be slightly more efficient than the Chebyshev method, and more efficient than Krylov method (see [2,23,26]). In fact, being based on a long-term recurrence, the performance of the Krylov method for exponential operators strongly depends on a careful choice of the threshold m for the Krylov subspace dimension; see the EXPPRO routine in [10,34], and the EXPOKIT package [36], As shown in [2, Tables VII–VIII], even with an optimal choice of m, the Krylov method is up to three times slower than the Chebyshev method on advection–diffusion problems.

We stress, finally, that ReLPM, being based on matrix vector multiplications like Krylov methods, is well-structured for a possible parallel implementation. One difference with respect to Krylov methods is that no inner products are needed, which as known represents an advantage within parallel sparse matrix computations (see, e.g., [16]).

Estimation of the focal interval, as well as computation of the Leja sequence, are displayed in the algorithm only for the sake of clarity. In practice, one can estimate once and for all the focal interval for B, and precompute a sufficiently large number of Leja points. Such data are passed as input parameters at each call of the algorithm, and scaled inside by Δt. This is the procedure to adopt when ReLPM is used for several values of Δt, as in the time marching scheme (5). Note, however, that the algorithm presented in Table 1 is general enough to be applied when matrix B is not constant in time, in which case spectral data would need to be evaluated at each time-step.

We have computed the Leja points via a discrete approximation of the focal interval. As discussed in [32] this discretization should be sufficiently dense compared with the maximum degree reached by the interpolating polynomial (at most of the order of tens in our examples). Alternatively, we could use the "fast Leja points" algorithm recently proposed in [1], which produces m Leja-like points with $\mathcal{O}(m^2)$ flops. We stress, however, that the computational cost of this part of the algorithm is in practice a negligible fraction of the overall cost, especially if the Leja points are produced once and for all (see above).

In practice, the exit test of the ReLPM is slightly more refined than that displayed in Table 1: in order to prevent fictitious convergence, instead of estimating the error with $|d_m| \cdot \|w_m\|_2$ (the norm of the last term in Newton interpolation), we use the mean of the norms of some consecutive terms (say the last 5). On the other hand, convergence may not take place when the smallest ellipse containing the spectrum and having as focal interval that numerically evaluated, has a large capacity. As already observed in [35] concerning Chebyshev series approximation, nonconvergence may be related to the effect of significant cancellation errors. There are two main strategies to overcome this problem: working in higher precision, or reducing the step-size (and thus the capacity and the approximating polynomial degree) when the size of the divided differences become close to the machine precision. Our implementation of the ReLPM for solving the discrete advection–diffusion equation (2) takes

into account this problem, and uses both strategies: it computes the divided differences in *quadruple* precision, and it automatically halves the time-steps when necessary, covering the original time-step Δt according to the time marching process (5).

4. Numerical tests

We have considered several n-dimensional systems of type (2), obtained by spatial discretization of the advection–diffusion equation (1) with $b(x) \equiv 0$ by finite differences (central or upwind discretization, depending on the value of the grid-Peclet numbers), with constant step-size $1/(n^{1/p}+1)$ on $\Omega = (0,1)^p$, $p=2$ (2D) or $p=3$ (3D), and various instances of the advection coefficients $\theta = (\theta_i)$, source-term $g(x)$ and initial datum $u_0(x)$. In particular, $u_0(x)$ has been chosen smooth (constant) or nonsmooth (strongly peaked), so that our tests correspond to the initial vectors $\boldsymbol{y}_0 = (1,\dots,1)^{\mathrm{T}}$ or $\boldsymbol{y}_0 = (1,\dots,1,100,1,\dots,1)^{\mathrm{T}}$ in (2).

We have compared the Leja points interpolation method ReLPM for the approximation of $\varphi(\Delta t_i B)\boldsymbol{v}_i$ within the time marching process (5), with the popular Crank–Nicolson method [5] (with variable step-size h_i)

$$\left(I - \frac{h_i B}{2}\right) \tilde{\boldsymbol{y}}_{i+1} = \left(I + \frac{h_i B}{2}\right) \tilde{\boldsymbol{y}}_i + h_i \boldsymbol{g}, \quad i = 0,1,\dots \ . \tag{18}$$

in the numerical solution of the stiff system (2) up to the steady state. Recall that the matrix B has eigenvalues with negative real-part, and thus the solution of (2) exhibits the steady-state $\lim_{t\to\infty} \boldsymbol{y}(t) = -B^{-1}\boldsymbol{g}$. Although Crank–Nicolson might not be considered the best choice for time integration of advection–diffusion problems (see, e.g., [13,41]), it is a robust method still widely used in engineering applications, and a sound baseline benchmark for any advection–diffusion solver (cf. [29]).

When the steady-state is null (e.g. in the homogeneous case $g(x) \equiv 0$), we have applied scheme (5) until

$$\|\boldsymbol{y}_i\|_2 \leqslant \varepsilon\|\boldsymbol{y}_0\|_2 \tag{19}$$

for small ε, i.e. a nearly steady-state is reached (we chose $\varepsilon = 10^{-4}$). Otherwise, (inhomogeneous case), we have computed the discrete solution until its "weighted derivative" becomes small

$$\frac{\|\boldsymbol{y}_{i+1} - \boldsymbol{y}_i\|_2/\Delta t_i}{\max\{\|\boldsymbol{y}_0\|_2, \|\boldsymbol{y}_{i+1}\|_2\}} \leqslant \tau$$

say with $\tau = 0.1$. The same stopping criteria have been applied to the computed solution $\tilde{\boldsymbol{y}}_i$ of the Crank–Nicolson scheme.

The Crank–Nicolson time-step h_i was chosen in such a way that the local truncation error e_{i+1}^{CN} (easily estimated by finite differences involving $\tilde{\boldsymbol{y}}_{i+1}$, $\tilde{\boldsymbol{y}}_i$ and $\tilde{\boldsymbol{y}}_{i-1}$) be smaller than a given tolerance

$$e_{i+1}^{\mathrm{CN}} \leqslant \mathrm{tol}_{i+1} = \varepsilon_1 \max\{\|\boldsymbol{y}_0\|_2, \|\tilde{\boldsymbol{y}}_{i+1}\|_2\}, \tag{20}$$

where ε_1 was chosen equal to 10^{-6} to have a local time accuracy at least of the same order as the spatial accuracy (i.e. $\mathcal{O}(n^{-2/p})$, p being the spatial dimension). When condition (20) is not satisfied, the time-step h_i is (proportionally) reduced and $\tilde{\boldsymbol{y}}_{i+1}$ recomputed, while when it is strictly satisfied, the next time-step h_{i+1} is (proportionally) enlarged. The linear system in (18) is solved by the

biconjugate gradient stabilized method (BiCGStab), preconditioned at each step (since the iteration matrix depends on h_i) with the incomplete triangular factor and no fill-in (ILU(0), cf. [14,40]), with a tolerance of $tol_i/10$. Recall that each iteration of the preconditioned BiCGStab requires the equivalent of four-sparse matrix–vector products, along with some scalar products and elementary vector operations.

As for scheme (5), since it is *exact* for autonomous linear systems of ODEs, there is no restriction on the choice of Δt_i, and we propose to select the local time-step in such a way that the relative variation of the solution be smaller than a given percentage, that is

$$\|\mathbf{y}_{i+1} - \mathbf{y}_i\|_2 \leqslant \eta \|\mathbf{y}_i\|_2, \quad 0 < \eta < 1. \tag{21}$$

The propagator $\varphi(\Delta t_i B)\mathbf{v}_i$ in (5) is approximated by the algorithm ReLPM described in Table 1, where the tolerance "tol" is selected similarly to (20), with the same ε_1. Recall that a subdivision of Δt_i is possible (see Section 3.1), but it is directly managed by the ReLPM. In order to avoid several (useless) time-steps near the possible null steady-state, we also add an absolute tolerance related to (19), getting

$$\|\mathbf{y}_{i+1} - \mathbf{y}_i\|_2 \leqslant \eta \|\mathbf{y}_i\|_2 + \varepsilon_2 \|\mathbf{y}_0\|_2. \tag{22}$$

In the numerical tests we have used $\varepsilon_2 = 10^{-3}$ and four different values of η, ranging from 0.1 to 0.75: with the smallest one, the solution is approximated at several time-points, in order to track with accuracy its evolution; with the largest one, only few large time-steps are sufficient to reach the steady-state (cf. Fig. 2). If condition (22) is not satisfied, the time step Δt_i is halved and $\mathbf{y}_{i+1}$ recomputed; on the other hand, if they are strictly satisfied (say with $\eta/2$ and $\varepsilon_2/2$ instead of η and ε_2), the next time-step Δt_{i+1} is doubled. In all the following numerical tests, we have chosen the same initial time-step for both methods, say $h_0 = \Delta t_0 = 10^{-5}$.

4.1. Comments on the numerical results

The experimental results, collected in Tables 2–19 and Figs. 2–6, show that scheme (5), implemented via ReLPM and the marching strategy (22), performs better than the variable step-size Crank–Nicolson method on several instances of the advection–diffusion problem (1)–(2). The superiority of other polynomial (Krylov and Faber-Chebyshev) methods with respect to Crank–Nicolson

Table 2
2D homogeneous, $n = 10\,000$, $\theta_1 = 100$, $\theta_2 = 100$, central discretization (Peclet number ≈ 0.5), smooth initial data, extreme eigenvalues computed by ARPACK with low tolerance or estimated by Gershgorin's theorem

η	ARPACK		Gershgorin	
	No. of time-steps	CPU (s)	No. of time-steps	CPU (s)
ReLPM				
0.1	95	4.18	95	4.22
0.25	41	2.26	43	2.34
0.5	24	1.95	25	1.81
0.75	18	2.11	19	1.52

Table 3
2D homogeneous, $n = 10\,000$, $\theta_1 = 100$, $\theta_2 = 100$, central discretization (Peclet number ≈ 0.5), smooth initial data

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	95	4.22	4.2
		0.25	43	2.34	7.6
375	17.77	0.5	25	1.81	9.8
		0.75	19	1.52	11.7

Table 4
Comparison of absolute and relative errors with respect to the "exact" solution, for the test case described in Table 3, at the "steady" state $t = 0.012$ (where $\|y(0.012)\|_2 \approx 10^{-4}\|y(0)\|_2$)

	Crank–Nicolson	ReLPM $\eta = 0.1$	ReLPM $\eta = 0.5$
Abs. err.	$6.5 \cdot 10^{-4}$	$1.8 \cdot 10^{-4}$	$1.8 \cdot 10^{-4}$
Rel. err.	$3.6 \cdot 10^{-2}$	$1.0 \cdot 10^{-2}$	$1.0 \cdot 10^{-2}$

Table 5
2D homogeneous, $n = 10\,000$, $\theta_1 = 100$, $\theta_2 = 100$, central discretization (Peclet number ≈ 0.5), nonsmooth initial data

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	106	4.02	5.1
		0.25	47	2.45	8.4
449	20.48	0.5	26	1.86	11.0
		0.75	18	1.50	13.7

Table 6
2D homogeneous, $n = 10\,000$, $\theta_1 = 500$, $\theta_2 = 500$, upwind discretization (Peclet number ≈ 2.48), smooth initial data

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	92	3.96	4.5
404	17.85	0.25	41	2.04	8.8
		0.5	23	1.51	11.8
		0.75	14	1.36	13.1

Table 7
2D homogeneous, $n = 10\,000$, $\theta_1 = 500$, $\theta_2 = 500$, upwind discretization (Peclet number ≈ 2.48), nonsmooth initial data

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	98	3.88	5.6
492	21.84	0.25	46	2.33	9.4
		0.5	25	1.70	12.8
		0.75	15	1.46	15.0

Table 8
2D homogeneous, $n = 10\,000$, $\theta_1 = 500$, $\theta_2 = 0$, upwind discretization (Peclet number ≈ 2.48), smooth initial data

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	89	3.53	5.3
403	18.88	0.25	39	1.92	9.8
		0.5	22	1.42	13.3
		0.75	15	1.31	14.4

Table 9
2D homogeneous, $n = 10\,000$, $\theta_1 = 500$, $\theta_2 = 0$, upwind discretization (Peclet number ≈ 2.48), nonsmooth initial data

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	102	3.61	6.0
461	21.48	0.25	46	2.18	9.9
		0.5	22	1.47	14.6
		0.75	16	1.41	15.2

Table 10
2D homogeneous, $n = 40\,000$, $\theta_1 = 100$, $\theta_2 = 100$, central discretization (Peclet number ≈ 0.25), smooth initial data

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	96	37.68	3.3
405	122.56	0.25	43	26.07	4.7
		0.5	25	20.57	6.0
		0.75	19	17.60	7.0

Table 11
2D homogeneous, $n = 40\,000$, $\theta_1 = 100$, $\theta_2 = 100$, central discretization (Peclet number ≈ 0.25), nonsmooth initial data

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	102	39.41	3.5
467	138.46	0.25	45	26.49	5.2
		0.5	25	21.01	6.6
		0.75	19	17.83	7.8

Table 12
2D inhomogeneous, $n = 10\,000$, $\theta_1 = 100$, $\theta_2 = 100$, central discretization (Peclet number ≈ 0.5), smooth initial data, postive source

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	41	2.24	8.0
375	17.91	0.25	20	1.86	9.6
		0.5	14	1.67	10.7
		0.75	13	2.87	6.2

Table 13
2D inhomogeneous, $n = 10\,000$, $\theta_1 = 100$, $\theta_2 = 100$, central discretization (Peclet number ≈ 0.5), nonsmooth initial data, positive source

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	55	2.61	7.7
450	20.04	0.25	25	2.05	9.8
		0.5	15	1.72	11.7
		0.75	13	2.85	7.0

Table 14
2D inhomogeneous, $n = 10\,000$, $\theta_1 = 100$, $\theta_2 = 100$, central discretization (Peclet number ≈ 0.5), smooth initial data, negative source

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	82	3.46	5.1
379	17.67	0.25	34	2.05	8.6
		0.5	21	1.90	9.3
		0.75	18	1.91	9.3

Table 15
2D inhomogeneous, $n = 10\,000$, $\theta_1 = 100$, $\theta_2 = 100$, central discretization (Peclet number ≈ 0.5), nonsmooth initial data, negative source

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	94	3.97	5.3
453	21.21	0.25	38	2.28	9.3
		0.5	22	2.04	10.4
		0.75	18	2.00	10.6

Table 16
3D homogeneous, $n = 27\,000$, $\theta_1 = 30$, $\theta_2 = 30$, $\theta_3 = 30$, central discretization (Peclet number ≈ 0.48), smooth initial data

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	82	17.38	4.1
238	71.42	0.25	41	10.93	6.5
		0.5	25	7.92	9.0
		0.75	19	6.80	10.5

Table 17
3D homogeneous, $n = 27\,000$, $\theta_1 = 30$, $\theta_2 = 30$, $\theta_3 = 30$, central discretization (Peclet number ≈ 0.48), nonsmooth initial data

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	86	18.22	4.4
288	79.62	0.25	41	10.91	7.3
		0.5	25	8.08	9.8
		0.75	18	6.70	11.9

Table 18
3D homogeneous, $n = 125\,000$, $\theta_1 = 50$, $\theta_2 = 50$, $\theta_3 = 50$, central discretization (Peclet number ≈ 0.49), smooth initial data

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	83	96.97	3.5
273	339.80	0.25	42	61.79	5.5
		0.5	23	41.98	8.1
		0.75	18	39.15	8.7

Table 19
3D homogeneous, $n = 125\,000$, $\theta_1 = 50$, $\theta_2 = 50$, $\theta_3 = 50$, central discretization (Peclet number ≈ 0.49), nonsmooth initial data

Crank–Nicolson		ReLPM			Speed-up
No. of time-steps	CPU (s)	η	No. of time-steps	CPU (s)	
		0.1	85	97.31	4.1
314	402.71	0.25	42	61.66	6.5
		0.5	23	42.02	9.6
		0.75	18	39.44	10.2
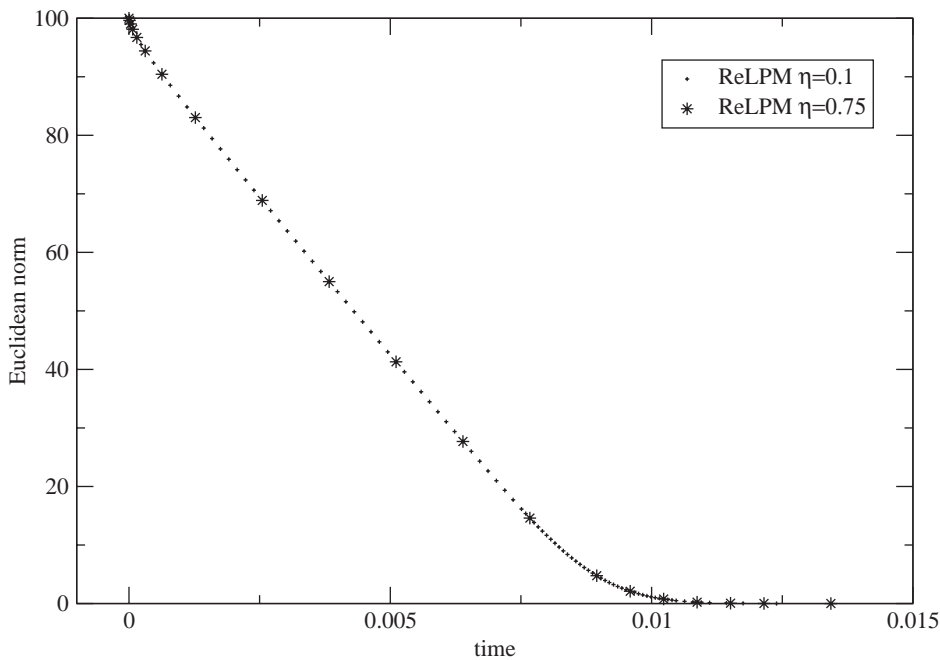
Fig. 2. Evolution of the norm of the solution computed by the ReLPM at several time-points ($\eta = 0.1$) and at few time-points ($\eta = 0.75$), for the test case described in Table 3 (2D, homogeneous, $n = 10\,000$, $\theta_1 = \theta_2 = 100$, smooth initial data).

for spatially discretized linear parabolic problems has been recognized in the numerical literature, but comparisons have been made with constant time-step marching for both approaches (see, e.g., [10,35]).

As already noticed, we have used ReLPM with a rough approximation of the focal interval, given by the extreme real-parts of the spectrum estimated by Gershgorin's theorem. Table 2 shows with a 2D example that efficiency of the method is not really affected by this choice. In fact, comparison with the more refined estimation of the focal interval adopted in [2] and recalled in Section 3 (extreme eigenvalues estimated by ARPACK $\rightarrow$ rectangle $\rightarrow$ optimal circumscribed ellipse $\rightarrow$ focal
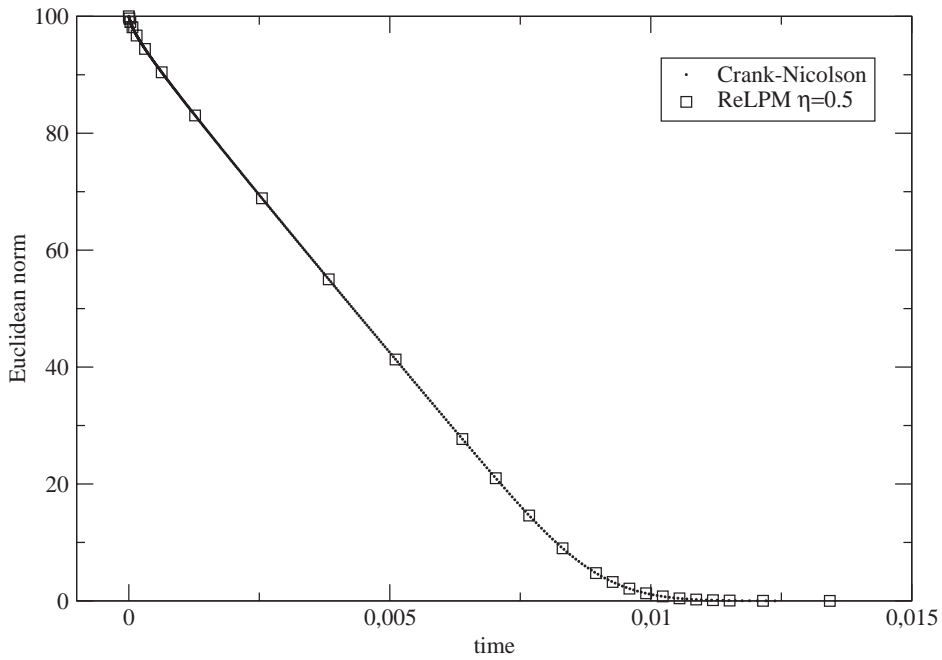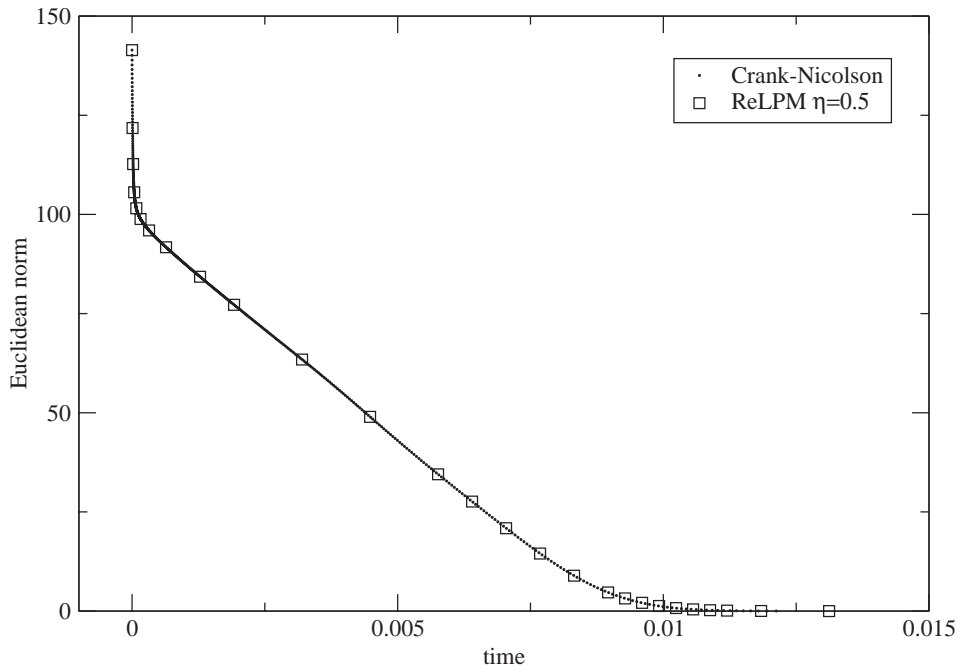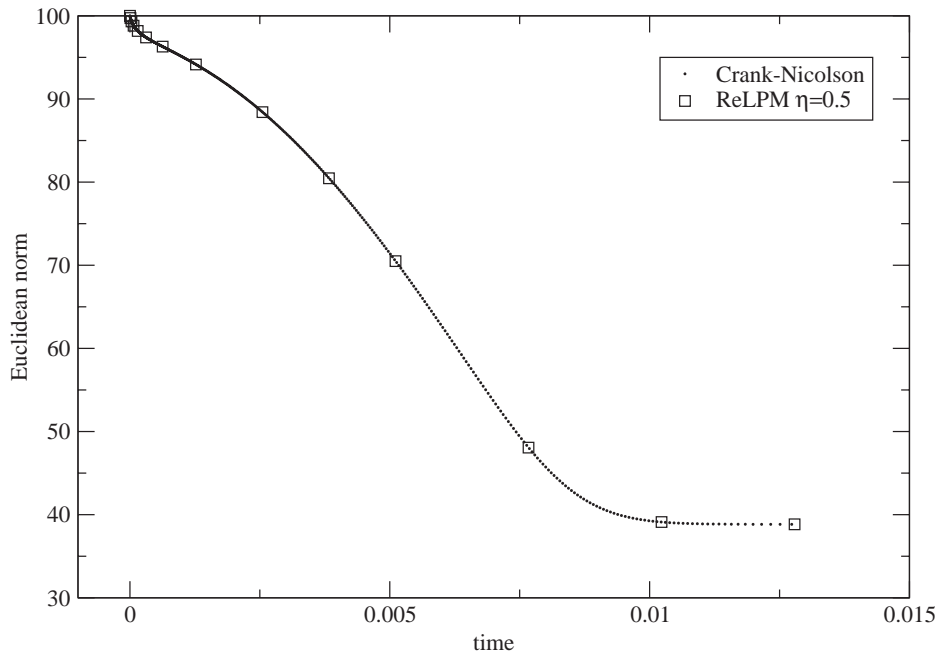
Fig. 3. Evolution of the norm of the solution computed by the Crank–Nicolson method and the ReLPM ($\eta = 0.5$), for the test case described in Table 3 (2D, homogeneous, $n = 10\,000$, $\theta_1 = \theta_2 = 100$, smooth initial data).

Fig. 4. Evolution of the norm of the solution computed by the Crank–Nicolson method and the ReLPM ($\eta = 0.5$), for the test case described in Table 5 (2D, homogeneous, $n = 10\,000$, $\theta_1 = \theta_2 = 100$, nonsmooth initial data).

Fig. 5. Evolution of the norm of the solution computed by the Crank–Nicolson method and the ReLPM ($\eta = 0.5$), for the test case described in Table 12 (2D, inhomogeneous, $n = 10\,000$, $\theta_1 = \theta_2 = 100$, smooth initial data, positive source).
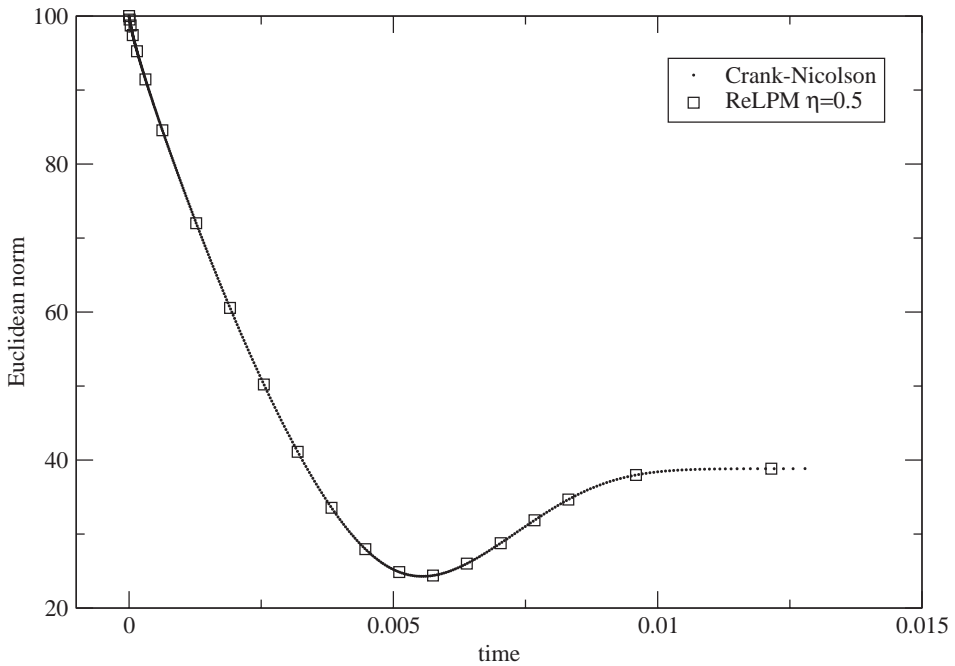

Fig. 6. Evolution of the norm of the solution computed by the Crank–Nicolson method and the ReLPM ($\eta = 0.5$), for the test case described in Table 14 (2D, inhomogeneous, $n = 10\,000$, $\theta_1 = \theta_2 = 100$, smooth initial data, negative source).

interval), shows that the number of time-steps and the total CPU time essentially do not depend on the estimation strategy.

Tables 3–11 refer to the homogeneous 2D case ($g \equiv 0$) of (1)–(2), where we adopted central differences or the upwind method depending on the Peclet number, and both smooth and nonsmooth initial data are considered. First, notice that the ReLPM approach is always faster than Crank–Nicolson, with speed-ups (ratios between CPU times) ranging from 3.3 to 14.4 for smooth, and from 3.5 to 15.2 for nonsmooth initial data. To this respect, we stress that Crank–Nicolson is more sensible to the smoothness of initial data, due to the effect of truncation errors; compare Tables 3 and 5, 6 and 7, 8 and 9, 10 and 11, and Figs. 3 and 4. Concerning the accuracy of the computed solutions, in Table 4 we have compared the absolute and relative errors with respect to the "exact" solution, for the test case described in Table 3, at the "steady" state $t = 0.012$ (where $\|\boldsymbol{y}(0.012)\|_2 \approx 10^{-4} \cdot \|\boldsymbol{y}(0)\|_2$). The reference solution has been computed by Crank–Nicolson with $\varepsilon_1 = 10^{-8}$ in (20), whereas we recall that the comparison of the errors is made using the same local tolerance for both methods, corresponding to $\varepsilon_1 = 10^{-6}$. Note that ReLPM is more accurate than Crank–Nicolson at the final time.

It is also worth noting that the choice of the spatial discretization scheme (central differences or upwind method) has a relatively small effect on the cost of both methods; see Tables 3, 6 and 8, and Tables 5, 7 and 9. Also an asymmetry in the advection terms does not seem to entail substantial differences at the same Peclet number, compare Tables 6 and 8, and Tables 7 and 9. On the other hand, comparing Tables 3 and 10 (or Tables 5 and 11), we see that the speed-ups become smaller when n increases (more accurate spatial discretization). This can be related to the fact that, while as expected the number of time-steps does not substantially depend on n, ReLPM faces increasing capacities of the estimated focal interval (cf. the qualitative convergence estimates (14)–(16); in view of Gershgorin's theorem, we expect roughly a capacity proportional to $n\Delta t$).

Similar considerations hold for Tables 12–15 (2D inhomogeneous instances with constant source-term) and Tables 16–19 (3D homogeneous instances). In particular, comparing Tables 16 and 18 (or Tables 17 and 19) we notice that the 3D speed-ups are less sensible to the increase of n with respect to 2D instances (indeed, in 3D we expect roughly a capacity proportional to $n^{2/3}\Delta t$). Observe also that in Tables 12 and 13 the CPU time for $\eta = 0.75$ is substantially greater than that for the smaller values of η, even if the number of time-steps is the smallest. This is due to the fact that the last time-step is too large to allow convergence (see Section 3.1), and the algorithm is forced to subdivide it, wasting a relatively large number of iterations.

Finally, Figs. 3–6 show that even the choice of the variation percentage $\eta = 0.5$ allows to track with some accuracy the evolution of the solution, with much less steps than Crank–Nicolson, exhibiting speed-ups ranging from 6.0 to 13.3. Moreover, we stress again that in any case (even with $\eta = 0.75$, which might be still acceptable for a qualitative recovery of the evolution, see Fig. 2) the computed solution vectors $\boldsymbol{y}_i$ at the time instants t_i are very accurate, since the time marching scheme (5) is exact and the tolerance required for ReLPM is small.

Acknowledgements

References

[1] J. Baglama, D. Calvetti, L. Reichel, Fast Leja points, Electron. Trans. Numer. Anal. 7 (1998) 124–140.

[2] L. Bergamaschi, M. Caliari, M. Vianello, Efficient approximation of the exponential operator for discrete 2D advection –diffusion problems, Numer. Linear Algebra Appl. 10 (3) (2003) 271–289.

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

[4] M. Caliari, Efficient implementation of exponential integrators for 2D and 3D advection–diffusion equations, Ph.D. Thesis in Computational Mathematics, University of Padova, advisors M. Vianello and L. Bergamaschi, 2003, URL http://www.math.unipd.it/~mcaliari/pdf/phd.pdf.

[5] J. Crank, P. Nicolson, A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type, Proc. Cambridge Philos. Soc. 43 (1947) 50–67.

[6] V.L. Druskin, L.A. Knizhnerman, Two polynomial methods for calculating functions of symmetric matrices, Comput. Math. Math. Phys. 29 (6) (1989) 112–121.

[7] S.W. Ellacott, Computation of Faber series with application to numerical polynomial approximation in the complex plane, Math. Comput. 40 (162) (1983) 575–587.

[8] B. Fischer, L. Reichel, Newton interpolation in Fejér and Chebyshev points, Math. Comput. 53 (187) (1989) 265–278.

[9] R.A. Friesner, L.S. Tuckerman, B.C. Dornblaser, T.V. Russo, A method for exponential propagation of large system of stiff nonlinear differential equations, J. Sci. Comput. 4 (4) (1989) 327–354.

[10] E. Gallopoulos, Y. Saad, Efficient solution of parabolic equations by Krylov subspace methods, SIAM J. Sci. Statist. Comput. 13 (5) (1992) 1236–1264.

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

[12] M. Hochbruck, C. Lubich, H. Selhofer, Exponential integrators for large systems of differential equations, SIAM J. Sci. Comput. 19 (5) (1998) 1552–1574.

[13] W. Hundsdorfer, J.G. Verwer, Numerical Solution of Time-Dependent Advection–Diffusion–Reaction Equations, Springer series in Computational Mathematics, Vol. 33, Springer, Berlin, 2003.

[14] D.S. Kershaw, The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations, J. Comput. Phys. 26 (1) (1978) 43–65.

[15] D. Lanser, J.G. Verwer, Analysis of operator splitting for advection–diffusion–reaction problems for air-pollution modelling, J. Comput. Appl. Math. 111 (1–2) (1999) 201–216.

[16] C. Le Calvez, Y. Saad, Modified Krylov acceleration for parallel environments, Appl. Numer. Math. 30 (2–3) (1999) 191–212.

[17] R.B Lehoucq, D.C. Sorensen, C. Yang, ARPACK Users' Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods, 1997, URL http://www.caam.rice.edu/software/ARPACK/UG/ug.html.

[18] F. Leja, Sur certaines suites liées aux ensembles plans et leur application à la représentation conforme, Ann. Polon. Math. 4 (1957) 8–13.

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

[20] T.A. Manteuffel, Adaptive procedure for estimating parameters for the nonsymmetric Tchebychev iteration, Numer. Math. 31 (2) (1978/1979) 183–208.

[21] A.I. Markushevic, Theory of Functions of a Complex Variable, Vol. 3, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1967.

[22] C. Moler, C. van Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, SIAM Rev. 45 (2003) 3–49.

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

[24] I. Moret, P. Novati, An interpolatory approximation of the matrix exponential based on Faber polynomials, J. Comput. Appl. Math. 131 (1–2) (2001) 361–380.

[25] I. Moret, P. Novati, Rational approximations of the matrix exponential. To appear in BIT (2003).

[26] P. Novati, Polynomial methods for the computation of functions of large unsymmetric matrices, Ph.D. Thesis in Computational Mathematics, University of Trieste, advisor I. Moret, 2000.

[27] P. Novati, A polynomial method based on Fejér points for the computation of functions of unsymmetric matrices, Appl. Numer. Math. 44 (1–2) (2003) 201–224.

[28] P. Novati, Solving linear initial value problems by Faber polynomials, Numer. Linear Algebra Appl. 10 (3) (2003) 247–270.

[29] G. Pini, G. Gambolati, Arnoldi and Crank–Nicolson methods for integration in time of the transport equation, Internat. J. Numer. Methods Fluids 35 (2001) 25–38.

[30] A. Quarteroni, A. Valli, Numerical Approximation of Partial Differential Equations, Springer Series in Computational Mathematics, Vol. 23, Springer, Berlin, 1994.

[31] S.C. Reddy, L.N. Trefethen, Pseudospectra of the convection–diffusion operator, SIAM J. Appl. Math. 54 (6) (1994) 1634–1649.

[32] L. Reichel, Newton interpolation at Leja points, BIT 30 (2) (1990) 332–346.

[33] L. Reichel, L.N. Trefethen, Eigenvalues and pseudo-eigenvalues of Toeplitz matrices, Linear Algebra Appl. 162/164 (1992) 153–185.

[34] Y. Saad, Analysis of some Krylov subspace approximations to the matrix exponential operator, SIAM J. Numer. Anal. 29 (1) (1992) 209–228.

[35] M.J. Schaefer, A polynomial based iterative method for linear parabolic equations, J. Comput. Appl. Math. 29 (1) (1990) 35–50.

[36] R.B. Sidje, Expokit, A software package for computing matrix exponentials, ACM Trans. Math. Software 24 (1) (1998) 130–156.

[37] H. Tal-Ezer, R. Kosloff, An accurate and efficient scheme for propagating the time dependend Schrödinger equation, J. Chem. Phys. 81 (9) (1984) 3967–3971.

[38] H. Tal-Ezer, R. Kosloff, C. Cerjen, Low-order polynomial approximation of propagators for the time-dependent Schrödinger equation, J. Comput. Phys. 100 (1) (1992) 179–187.

[39] L.N. Trefethen, Computation of pseudospectra, in: A. Iserles (Ed.), Acta Numerical, Vol. 8, Cambridge University Press, Cambridge, 1999, pp. 247–295.

[40] H.A. van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 13 (2) (1992) 631–644.

[41] C.B. Vreugdenhil, B. Koren (Eds.), Numerical methods for advection–diffusion problems, Notes on Numerical Fluid Mechanics, Vol. 45, Friedrivk, Vieweg and Sohn, Braunschweig, 1993.

[42] J.L. Walsh, Interpolation and approximation by rational functions in the complex domain, American Mathematical Society Colloquium Publications, Vol. XX, AMS, Providence, RI, 1935.