

The ReLPM Exponential Integrator for FE Discretizations of Advection-Diffusion Equations^{*}

Luca Bergamaschi¹, Marco Caliari², and Marco Vianello³

¹ Dip.to di Met. e Mod. Mat. per le Scienze Appl., Università di Padova

² Dip.to di Informatica, Università di Verona

³ Dip.to di Mat. Pura ed Appl., Università di Padova, marcov@math.unipd.it

Abstract. We implement an exponential integrator for large and sparse systems of ODEs, generated by FE (Finite Element) discretization with mass-lumping of advection-diffusion equations. The relevant exponential-like matrix function is approximated by polynomial interpolation, at a sequence of real Leja points related to the spectrum of the FE matrix (ReLPM, Real Leja Points Method). Application to 2D and 3D advection-dispersion models shows speed-ups of one order of magnitude with respect to a classical variable step-size Crank-Nicolson solver.

1 The Advection-Diffusion Model

We consider the classical evolutionary advection-diffusion problem

$$\begin{cases} \frac{\partial c}{\partial t} = \operatorname{div}(D\nabla c) - \operatorname{div}(cv) + \phi & x \in \Omega, t > 0 \\ c(x, 0) = c_0(x) & x \in \Omega \\ c(x, t) = g_D(x, t) & x \in \Gamma_D, t > 0 \\ \langle D\nabla c(x, t), \nu \rangle = g_N(x, t) & x \in \Gamma_N, t > 0 \end{cases} \quad (1)$$

with mixed Dirichlet and Neumann boundary conditions on $\Gamma_D \cup \Gamma_N = \partial\Omega$, $\Omega \subset \mathbb{R}^d$, $d = 2, 3$; cf. [1]. Equation (1) represents, e.g., a simplified model for solute transport in groundwater flow (advection-dispersion), where c is the solute concentration, D the hydrodynamic dispersion tensor, $D_{ij} = \alpha_T |v| \delta_{ij} + (\alpha_L - \alpha_T) v_i v_j / |v|$, $1 \leq i, j \leq d$, v the average linear velocity of groundwater flow and

^{*} Work supported by the research project CPDA028291 “Efficient approximation methods for nonlocal discrete transforms” of the University of Padova, and by the subproject “Approximation of matrix functions in the numerical solution of differential equations” (co-ordinator M. Vianello, University of Padova) of the MIUR PRIN 2003 project “Dynamical systems on matrix manifolds: numerical methods and applications” (co-ordinator L. Lopez, University of Bari). Thanks also to the numerical analysis group at the Dept. of Math. Methods and Models for Appl. Sciences of the University of Padova, for having provided FE matrices for our numerical tests, and for the use of their computing resources.

ϕ the source (cf. [2,3]). The standard Galerkin FE discretization of (1) nodes $\{x_i\}_{i=1}^N$ and linear basis functions $\{\varphi_i\}_{i=1}^N$ [4], gives the linear system of ODEs

$$\begin{cases} P\dot{\mathbf{z}} = H\mathbf{z} + \mathbf{b} + \mathbf{q}, & t > 0 \\ \mathbf{z}(0) = \mathbf{c}_0 \end{cases} \tag{2}$$

where $\mathbf{z} = [z_1(t), \dots, z_N(t)]^T$, $\mathbf{c}_0 = [c_0(x_1), \dots, c_0(x_N)]^T$, P is the symmetric positive-definite mass matrix, H the (nonsymmetric) stiffness matrix and

$$b_i \approx \int_{\text{supp}(\varphi_i)} \phi \varphi_i \, d\Omega, \quad q_i \approx \int_{\Gamma_N \cap \text{supp}(\varphi_i)} g_N \varphi_i \, d\Gamma_N, \quad i = 1, \dots, N \tag{3}$$

As it is known, such a basic FE discretization makes sense numerically only on sufficiently fine grids (small grid Péclet numbers). Otherwise, special stabilization techniques should be adopted, like Petrov-Galerkin upwinding; see [1].

Actually, system (2) does not take into account yet of possible Dirichlet conditions: we discuss in detail their treatment within the next sections.

2 An Exponential Integrator via Mass-Lumping

In the sequel we consider stationary velocity, source and boundary conditions in (1), which give a system of ODEs like (2) with constant H , \mathbf{b} and \mathbf{q} . Such a system can be rewritten in the form

$$\begin{cases} \dot{\mathbf{z}} = P^{-1}H\mathbf{z} + \boldsymbol{\phi} + P^{-1}\mathbf{q}, & t > 0 \\ \mathbf{z}(0) = \mathbf{c}_0 \end{cases}$$

which is suitable for the application of exponential integrators (cf. [5,6,7]). Observe that $\boldsymbol{\phi} = P^{-1}\mathbf{b}$ since we chose $\mathbf{b} = P[\phi(x_1), \dots, \phi(x_N)]^T$ in (3). In order to impose the lacking Dirichlet conditions we make vanishing the i -th row of P^{-1} and of $\boldsymbol{\phi}$ (recall that g_D is stationary) and substitute the i -th component of the initial vector \mathbf{c}_0 with $g_D(x_i)$, if $x_i \in \Gamma_D$, obtaining \widehat{P}^{-1} , $\widehat{\boldsymbol{\phi}}$ and $\widehat{\mathbf{c}}_0$. In practice, the system above is modified into

$$\begin{cases} \dot{\mathbf{c}} = \widehat{P}^{-1}H\mathbf{c} + \widehat{\boldsymbol{\phi}} + \widehat{P}^{-1}\mathbf{q}, & t > 0 \\ \mathbf{c}(0) = \widehat{\mathbf{c}}_0 \end{cases} \tag{4}$$

System (4) is now the discrete approximation of the PDE (1), where the Dirichlet conditions have been artificially imposed also to the initial data \mathbf{c}_0 .

As known, the solution can be written explicitly in the exponential form

$$\mathbf{c}(t) = \widehat{\mathbf{c}}_0 + t\varphi(t\widehat{P}^{-1}H)[\widehat{P}^{-1}H\widehat{\mathbf{c}}_0 + \widehat{\boldsymbol{\phi}} + \widehat{P}^{-1}\mathbf{q}] \ ,$$

where $\varphi(z)$ is the *entire* function $\varphi(z) = (e^z - 1)/z$ if $z \neq 0$, $\varphi(0) = 1$. Clearly, availability of matrix P^{-1} (and thus \widehat{P}^{-1}) is a computationally expensive task:

so, we can apply the well known *mass-lumping* technique (sum on the diagonal of all the row elements) to P in order to obtain a diagonal matrix P_L (and, immediately, P_L^{-1} and $\widehat{P_L^{-1}}$); cf. [1,8]. Now we can consider the transformed system (4) with $\widehat{P^{-1}}$ replaced by $\widehat{P_L^{-1}}$, and apply the *exact* and *explicit* time-marching scheme (see the work by Schaefer [5] for FD spatial discretization)

$$\mathbf{c}_{k+1} = \mathbf{c}_k + \Delta t_k \varphi(\Delta t_k H_L) \mathbf{v}_k, \quad k = 0, 1, \dots, \quad \mathbf{c}_0 = \widehat{\mathbf{c}}_0, \quad (5)$$

where we have defined $H_L = \widehat{P_L^{-1}} H$ and $\mathbf{v}_k = H_L \mathbf{c}_k + \widehat{\boldsymbol{\phi}} + \widehat{P_L^{-1}} \mathbf{q}$. Exactness of the exponential integrator (5) entails that the time-steps Δt_k can be chosen, at least in principle, arbitrarily large with no loss of accuracy, making it an appealing alternative to classical finite-difference integrators (cf. [6,7,9]).

However, the practical application of (5) rests on the possibility of approximating efficiently the *exponential operator* $\varphi(\Delta t H_L) \mathbf{v}$, where $\mathbf{v} \in \mathbb{R}^N$. To this aim, two classes of *polynomial* methods are currently used. We have Krylov-like methods, which are based on the idea of projecting the operator 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., [10,11,12], and [13, 14] for other (nonstandard) Krylov-like approaches. The second class consists of methods based on polynomial interpolation or series expansion of the entire function φ on a suitable compact subset containing the spectrum (or in general the field of values) of the matrix (e.g. Faber and Chebyshev series, interpolation at special points like Faber, Fejér and Leja points). They typically require some preliminary estimate of the underlying spectral structure, but, despite of this, this second class of methods turned out to be competitive with Krylov-based approaches, especially on very large nonsymmetric matrices, cf. [15,16,17, 18,19]. In this work we adopt the Real Leja Points Method (shortly ReLPM), recently proposed in the frame of FD spatial discretization of advection-diffusion equations [9].

3 Computing the Exponential Operator by the ReLPM (Real Leja Points Method)

Sequences of Leja points $\{z_j\}_{j=0}^{\infty}$ for the compact $K \subset \mathbb{C}$ are defined recursively as follows: if z_0 is an arbitrary fixed point in K (usually such as $|z_0| = \max_{z \in K} |z|$, cf. [20]), the z_j are chosen in such a way that

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

By the maximum principle, the Leja points for K lie on ∂K . We recall that an efficient algorithm for the computation of a sequence Leja points, the so-called Fast Leja Points, has been recently proposed in [21].

Now, for any fixed compact set K with more than one point, there is a function Φ which maps the exterior of K conformally onto the exterior of the unit

disk, and satisfies in particular the condition $\lim_{z \rightarrow \infty} \Phi(z)/z = \gamma$, where $\gamma > 0$ is called *capacity* of K (cf. [22]). For any $R > 0$, define $\Gamma_R = \{z : |\Phi(z)| = R/\gamma\}$ and K_R the bounded domain with boundary Γ_R ; observe that $K_\gamma = K$, K_R has capacity R , and $K_{R_1} \subseteq K_{R_2}$ if $R_1 \leq R_2$. Let f be an entire function: it is well known [22,20] that the sequence of polynomials p_m of degree m that interpolate f on the Leja points $\{z_j\}_{j=0}^m$ for K converges *maximally* to f on every K_R , i.e. asymptotically like the best uniform approximation polynomials, and thus *superlinearly*, that is

$$\limsup_{m \rightarrow \infty} \|f - p_m\|_{K_R}^{1/m} = 0 .$$

Moreover, Leja sequences are attractive for interpolation at high-degree, in view of the stability of the corresponding algorithm in the Newton form (cf. [20]).

From these properties, we can derive a stable and efficient polynomial approximation method for the matrix operator φ in (5). In fact, cf. [17], if $\{p_m\}$ converges maximally to the entire function f on a compact K , then

$$\limsup_{m \rightarrow \infty} \|f(A)\mathbf{v} - p_m(A)\mathbf{v}\|_2^{1/m} = 0 . \tag{6}$$

In particular, if R is such that the spectrum $\sigma(A)$ of the matrix $A \in \mathbb{R}^{N \times N}$ is contained in K_R , and if $A = X^{-1}\Lambda X$ is diagonalizable, we have the estimate

$$\|f(A)\mathbf{v} - p_m(A)\mathbf{v}\|_2 \leq \text{cond}_2(X) \cdot \|f - p_m\|_{K_R} \cdot \|\mathbf{v}\|_2 = \mathcal{O} \left(\left(\frac{e \cdot R}{m} \right)^{m+1} \right) , \tag{7}$$

for $m \geq m_0 > R$. More refined convergence estimates (even in the nondiagonalizable case) can be obtained by resorting to the notions of pseudospectra and field of values of A ; cf. [17,9].

Following [9], an algorithm for the approximation of the advection-diffusion FE propagator $\varphi(\Delta t H_L)\mathbf{v}$ can be now easily developed, by means of Newton interpolation at “spectral” Leja points. In the sequel, the compact subset used for estimating the spectrum of H_L in (5) will be an ellipse in a suitable family of *confocal ellipses* $\{K_R\}$, where $K = K_{c/2} = [d - c, d + c] \subset \mathbb{R}$ is the common focal interval. Note that we restrict our attention to ellipses symmetric with respect to the real axis, since in our application we deal with real matrices. Moreover, it makes sense to consider a real focal interval, since the numerically evaluated spectra of H_L with small grid Péclet number have an horizontal projection larger than the vertical projection. The advantage of working with such confocal ellipses stems from complex approximation theory: if the spectrum $\sigma(\Delta t H_L)$ is contained in K_R for some R , then a sequence of polynomials converging maximally to φ on $K_{c/2} = [d - c, d + c]$ converges maximally also on K_R (cf. [22]), and thus the corresponding sequence of matrix polynomial operators converges maximally to $\varphi(\Delta t H_L)\mathbf{v}$; cf. (6)–(7). 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). Clearly, a key step in this procedure is given by

estimating at low cost the reference focal interval for the spectrum of $\Delta t H_L$. Following [5] and [9], which deal with FD discretizations, we adopt the simplest estimate given directly by Gershgorin's theorem.

We can now sketch the algorithm for Leja points interpolation (ReLPM) of the advection-diffusion propagator $\varphi(\Delta t H_L)\mathbf{v}$ in Table 1. The ReLPM algorithm

Table 1. Algorithm ReLPM (Real Leja Points Method)

-
- INPUT: H_L , \mathbf{v} , Δt , tol
 - Estimate the spectral focal interval $[d-c, d+c]$ for $\Delta t H_L$, by Gershgorin's theorem
 - Compute a sequence of Fast Leja Points $\{\xi_j\}$ in $[d-c, d+c]$ as in [21]
 - $d_0 := \varphi(\xi_0)$, $\mathbf{w}_0 := \mathbf{v}$, $\mathbf{p}_0 := d_0 \mathbf{w}_0$, $m := 0$
 - WHILE $e_m^{\text{Leja}} := |d_m| \cdot \|\mathbf{w}_m\|_2 > \text{tol}$
 - $\mathbf{w}_{m+1} := (\Delta t H_L - \xi_m I) \mathbf{w}_m$
 - $m := m + 1$
 - compute the next divided difference d_m
 - $\mathbf{p}_m := \mathbf{p}_{m-1} + d_m \mathbf{w}_m$
 - OUTPUT: the vector \mathbf{p}_m : $\|\mathbf{p}_m - \varphi(\Delta t H_L)\mathbf{v}\|_2 \approx e_m^{\text{Leja}} \leq \text{tol}$
-

turns out to be 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 implementation details not reported in Table 1, we refer to [9]. We only stress that ReLPM is very well structured for a possible *parallel implementation*, since it uses only sparse matrix-vector multiplications and basic vector operations, but no inner product.

4 Application: 2D and 3D Advection-Dispersion Models

We present now three examples (cf. [13]), concerning application of the ReLPM exponential integrator (5) to advection-dispersion models like (1), together with the comparison with the classical variable step-size Crank-Nicolson solver.

Example 1. As first numerical test, we consider problem (1) on the 2D rectangular domain $\Omega = [0, 1] \times [0, 0.5]$, with a regular grid of $N = 161 \times 81 = 13041$ nodes and $M = 25600$ triangular elements. Here, $\phi \equiv 0$ and $c_0 \equiv 1$. Dirichlet boundary conditions $c = 1$ for $0.2 \leq y \leq 0.3$ and $c = 0$ elsewhere are imposed on $\Gamma_D = \{0\} \times [0, 0.5]$; the Neumann condition $\partial c / \partial \nu = 0$ is prescribed on $\Gamma_N = \partial\Omega \setminus \Gamma_D$. The velocity is $v = (v_1, v_2) = (1, 0)$, and $\alpha_L = \alpha_T = 0.00625$.

Example 2. The second numerical test is the extension of the first on a 3D domain $\Omega = [0, 1] \times [0, 0.5] \times [0, 1]$, with a regular grid of $N = 81 \times 41 \times 9 = 29889$

nodes and $M = 153600$ tetrahedral elements. The boundary conditions of the previous test are extended along the z axis, while we have taken the peaked initial data $\mathbf{c}_0 = [1, \dots, 1, 100, 1, \dots, 1]^T$. The velocity is $v = (v_1, v_2, v_3) = (1, 0, 0)$, and $\alpha_L = \alpha_T = 0.0125$.

Example 3. In the last numerical test, the domain is the unit circle with a mesh consisting of $N = 35313$ nodes and $M = 245997$ triangular elements, generated by the `triangle` software of Shewchuk (available at www.netlib.org/voronoi). Dirichlet boundary condition $c = 0$ are imposed on the whole boundary. The velocity is $v = (v_1, v_2) = (1, 1)$, and $\alpha_L = \alpha_T = 0.00625$; moreover $c_0 \equiv 1$, and the source is constant and negative, $\phi \equiv -1$.

4.1 Crank-Nicolson (CN) Method

Although Crank-Nicolson (CN) method might not be considered the best choice for time integration of advection-diffusion problems, it is a robust method still widely used in engineering applications, and a sound baseline benchmark for any advection-diffusion solver (cf. [13]). In the case of the relevant ODEs system (2) (with stationary \mathbf{b} and \mathbf{q}), its variable step-size version writes as

$$\left(P - \frac{h_k}{2}H\right)\mathbf{u}_{k+1} = \left(P + \frac{h_k}{2}H\right)\mathbf{u}_k + h_k(\mathbf{b} + \mathbf{q}), \quad k = 0, 1, \dots, \quad \mathbf{u}_0 = \mathbf{c}_0 .$$

In order to impose Dirichlet conditions, we change the i -th row of the system matrix above with the basis vector \mathbf{e}_i and the i -th component of the right hand side with the value of $g_D(x_i)$, if $x_i \in \Gamma_D$. The linear system is solved by the biconjugate gradient stabilized method, preconditioned at each step (since the system matrix depends on h_k) with the incomplete triangular factor and no fill-in. As for estimation of the local truncation error $\mathcal{O}(\|\ddot{\mathbf{c}}(t_k)\|_2 h_k^3)$ and step-size control, we have used standard finite-difference approximation of the third derivatives.

4.2 Numerical Tests and Comparisons

In Table 2 we have compared the absolute and relative errors with respect to the “exact” solution for Example 1 at the “steady” state $t = 1.3$ (where $\|\dot{\mathbf{c}}(1.3)\|_2 \leq 10^{-2} \cdot \|\mathbf{c}_0\|_2$). The reference solution has been computed by CN with a local tolerance equal to 10^{-6} , whereas the comparison of the errors is made using a local tolerance of 10^{-4} for both methods (namely “tol” for the ReLPM algorithm in Table 1), which guarantees an error of the order of the spatial discretization error. Note that ReLPM is more accurate than CN at the final time, which shows that the mass-lumping technique does not significantly degrade the accuracy of the exponential integrator (5).

While for CN the local time-step is selected adaptively, in order to guarantee a local error below the given tolerance, for scheme (5) there is no restriction on the choice of Δt_k , since it is *exact* for autonomous linear systems of ODEs. To

Table 2. Comparison of absolute and relative errors for Example 1

	CN	ReLPM $\eta = 0.1$	ReLPM $\eta = 0.5$
abs. err.	$3.5 \cdot 10^{-2}$	$4.7 \cdot 10^{-3}$	$4.8 \cdot 10^{-3}$
rel. err.	$1.0 \cdot 10^{-3}$	$1.4 \cdot 10^{-4}$	$1.4 \cdot 10^{-4}$

follow with some accuracy the evolution of the solution, we propose as in [9] to select the local time-step in (5) in such a way that the relative variation of the solution be smaller than a given percentage η , that is

$$\|\mathbf{c}_{k+1} - \mathbf{c}_k\|_2 \leq \eta \cdot \|\mathbf{c}_k\|_2, \quad 0 < \eta < 1. \quad (8)$$

If condition (8) is not satisfied, the time step Δt_k is halved and \mathbf{c}_{k+1} recomputed; if it is satisfied with $\eta/2$ instead of η , the next time-step Δt_{k+1} is doubled.

Tables 3–5 show that the ReLPM exponential integrator is always faster than CN method, with speed-ups (ratio between CPU times) ranging from 5.1 to 16.2.

Table 3. Comparison of CN and ReLPM for Example 1 (constant initial data)

CN		ReLPM			
# time-steps	CPU s.	η	# time-steps	CPU s.	Speed-up
451	37.47	0.1	41	7.33	5.1
		0.25	17	4.62	8.1
		0.5	11	4.02	9.3
		0.75	8	3.68	10.2

Table 4. Comparison of CN and ReLPM for Example 2 (peaked initial data)

CN		ReLPM			
# time-steps	CPU s.	η	# time-steps	CPU s.	Speed-up
385	198.24	0.1	44	27.53	7.2
		0.25	22	19.94	9.9
		0.5	11	15.37	12.9
		0.75	9	12.26	16.2

Table 5. Comparison of CN and ReLPM for Example 3 (constant initial data)

CN		ReLPM			
# time-steps	CPU s.	η	# time-steps	CPU s.	Speed-up
1079	507.36	0.1	90	66.24	7.7
		0.25	38	41.53	12.2
		0.5	21	36.09	14.0
		0.75	15	32.09	15.8

Note that the local tolerance for Examples 1 and 2 has been chosen equal to 10^{-4} , that is the order of spatial discretization error, whereas in Example 3

equal to 10^{-5} , to match the finer accuracy of the FE discretization. Finally, Fig. 1 shows that even the choice of the variation percentage $\eta = 0.5$ allows to track with sufficient accuracy the evolution of the solution, with much less steps than CN, exhibiting speed-ups of one order of magnitude.

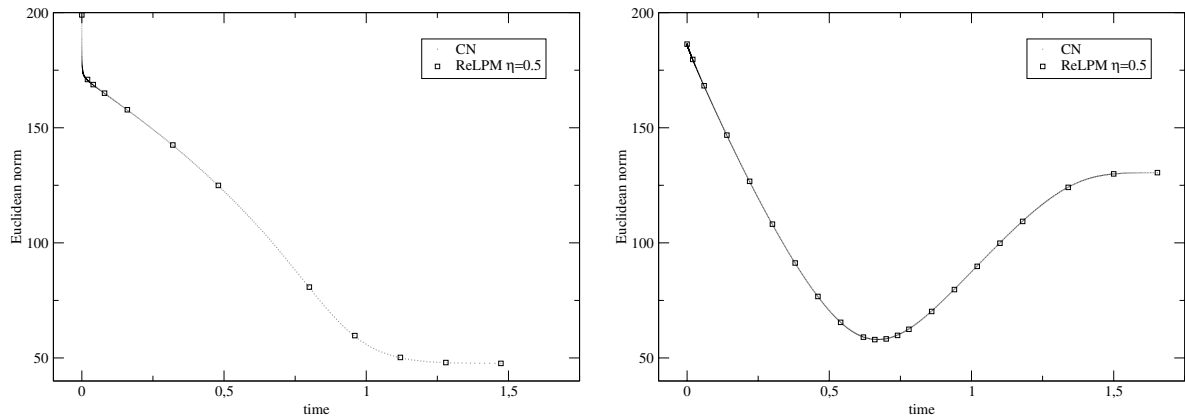


Fig. 1. Evolution of the norm of the solution computed by the CN method and the ReLPM ($\eta = 0.5$) for Example 2 (left) and Example 3 (right)

References

1. Hundsdorfer, W., Verwer, J.G.: Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations. Volume 33 of Springer series in Computational Mathematics. Springer-Verlag, Berlin (2003)
2. Wood, W.L.: Introduction to numerical methods for water resources. Oxford Science Publications. The Clarendon Press, Oxford University Press, New York (1993)
3. Fetter, C.W.: Applied Hydrogeology. Prentice Hall (2000)
4. Thomée, V.: Galerkin finite element methods for parabolic problems. Springer-Verlag, Berlin (1997)
5. Schaefer, M.J.: A polynomial based iterative method for linear parabolic equations. *J. Comput. Appl. Math.* **29** (1990) 35–50
6. Gallopoulos, E., Saad, Y.: Efficient solution of parabolic equations by Krylov subspace methods. *SIAM J. Sci. Statist. Comput.* **13** (1992) 1236–1264
7. Hochbruck, M., Lubich, C., Selhofer, H.: Exponential integrators for large systems of differential equations. *SIAM J. Sci. Comput.* **19** (1998) 1552–1574
8. Vanselow, R.: Error estimates of a FEM with lumping for parabolic PDEs. *Computing* **68** (2002) 131–141
9. Caliari, M., Vianello, M., Bergamaschi, L.: Interpolating discrete advection-diffusion propagators at Leja sequences. Preprint, to appear in *J. Comput. Appl. Math.* (2003)
10. Hochbruck, M., Lubich, C.: On Krylov subspace approximations to the matrix exponential. *SIAM J. Numer. Anal.* **34** (1997) 1911–1925
11. Saad, Y.: Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.* **29** (1992) 209–228

12. Sidje, R.B.: Expokit. A Software Package for Computing Matrix Exponentials. *ACM Trans. Math. Software* **24** (1998) 130–156
13. Pini, G., Gambolati, G.: Arnoldi and Crank-Nicolson methods for integration in time of the transport equation. *Int. J. Numer. Meth. Fluids* **35** (2001) 25–38
14. Moret, I., Novati, P.: RD rational approximation of the matrix exponential. Preprint, to appear in *BIT* (2003)
15. Bergamaschi, L., Vianello, M.: Efficient computation of the exponential operator for large, sparse, symmetric matrices. *Numer. Linear Algebra Appl.* **7** (2000) 27–45
16. Bergamaschi, L., Caliari, M., Vianello, M.: Efficient approximation of the exponential operator for discrete 2D advection-diffusion problems. *Numer. Linear Algebra Appl.* **10** (2003) 271–289
17. Moret, I., Novati, P.: The computation of functions of matrices by truncated Faber series. *Numer. Funct. Anal. Optim.* **22** (2001) 697–719
18. Moret, I., Novati, P.: An interpolatory approximation of the matrix exponential based on Faber polynomials. *J. Comput. Appl. Math.* **131** (2001) 361–380
19. Novati, P.: A polynomial method based on Fejér points for the computation of functions of unsymmetric matrices. *Appl. Numer. Math.* **44** (2003) 201–224
20. Reichel, L.: Newton interpolation at Leja points. *BIT* **30** (1990) 332–346
21. Baglama, J., Calvetti, D., Reichel, L.: Fast Leja points. *Electron. Trans. Numer. Anal.* **7** (1998) 124–140
22. Walsh, J.L.: Interpolation and approximation by rational functions in the complex domain. Volume XX of *Amer. Math. Soc. Colloq. Publ.* AMS (1935)