Numerische Mathematik manuscript No.
(will be inserted by the editor)

Fast Runge-Kutta approximation of inhomogeneous parabolic equations

María López-Fernández[1], Christian Lubich[2], Cesar Palencia[1], and Achim Schädle[3]

[1] Departamento de Matemática Aplicada y Computación, Universidad de Valladolid, Valladolid, Spain. E-mail: {marial, palencia}@mac.cie.uva.es
[2] Mathematisches Institut, Universität Tübingen, Auf der Morgenstelle 10, D–72076 Tübingen, Germany. E-mail: lubich@na.uni-tuebingen.de
[3] ZIB Berlin, Takustr. 7, D-14195 Berlin, Germany. E-mail: schaedle@zib.de

January 24, 2005

Summary The result after N steps of an implicit Runge-Kutta time discretization of an inhomogeneous linear parabolic differential equation is computed, up to accuracy ε, by solving only

$$O\Big(\log N \, \log \frac{1}{\varepsilon}\Big)$$

linear systems of equations. We derive, analyse, and numerically illustrate this fast algorithm.

Mathematics Subject Classification (2000): 65M20

1 Introduction

In the method of lines, semi-discretization in space turns a linear parabolic differential equation into a large, stiff system of ordinary differential equations

$$u'(t) + Au(t) = g(t), \qquad u(0) = u_0, \tag{1}$$

possibly with a mass matrix multiplying the time derivative. This system is subsequently discretized in time, e.g., by the implicit Euler method with step size h,

$$(I + hA)u_n = u_{n-1} + hg(t_n), \qquad n = 1, \ldots, N.$$

The approximation u_N for a prescribed step number N is thus obtained by solving a sequence of N linear systems with a matrix of the form $\lambda + A$, where $\lambda = 1/h$ in the implicit Euler method. For N steps with a higher-order, m-stage Runge-Kutta method, there are mN such linear systems, possibly with complex λ as in the excellent Radau IIA methods. Even if fast techniques such as multi-grid methods are used, solving the linear systems of equations typically constitutes the main computational cost, in particular for problems in complicated spatial geometries.

In this paper we propose an algorithm to compute the implicit Runge-Kutta approximation u_N at a fixed time $T = Nh$, up to an arbitrary accuracy ε, by doing N Runge-Kutta steps for differential equations of the form $y'(t) = \lambda y(t) + g(t)$, each step in parallel for $O(\log(1/\varepsilon))$ complex parameters λ, and by solving only

$$O(\log N \, \log \tfrac{1}{\varepsilon}) \text{ linear systems}$$

with matrices of the form $\lambda + A$, all of which can be solved in parallel. The constant in this work estimate is moderate: for a relative accuracy of 10^{-5} and $N \leq 10^5$ time steps we need to solve less than 100 linear systems! For large step numbers N, the number of linear systems is thus dramatically reduced, both in a sequential and in a parallel computational setting.

The algorithm is highly efficient for computing Runge-Kutta approximations to the solution of (1) at a relatively small number of selected time points or of short subintervals, but it is not useful for computing *all* values $u_1, \ldots, u_N$.

Basic ingredients of the algorithm are the following:

- the discrete variation-of-constants formula for the Runge-Kutta method;
- the Cauchy integral representation of the approximations to the operator exponential;
- the discretization of the contour integrals, using $O(\log N)$ contours with $O(\log(1/\varepsilon))$ quadrature points each;
- the discrete semigroup property, which permits us to reinterpret the split sums as Runge-Kutta approximations to solutions of equations of the form $y'(t) = \lambda y(t) + g(t)$.

The algorithm given here is closely related to the fast convolution algorithms developed in [11,13]. The error analysis for the discretized contour integrals follows the analysis of inverse Laplace transform approximations in [8].

Discretized contour integrals have been used previously in several instances in the numerical solution of parabolic equations: for homogeneous problems ($g \equiv 0$) in [14] similarly to Talbot's method [19] for the inversion of the Laplace transform $(s+A)^{-1}u_0$, and more recently for inhomogeneous problems [15, 5] using the Laplace transform of the inhomogeneity g or assuming special properties, in particular analyticity, of g. In contrast, the present algorithm works directly with the discrete values $g(t)$ that are used in the Runge-Kutta discretization of (1). No smoothness conditions for g are needed. This is because the algorithm approximates the *discrete* result of the Runge-Kutta method, with an error that does not depend on the smoothness of either the inhomogeneity or the solution. Of course, to make sense, the Runge-Kutta discretization of (1) with the considered step size h should be sufficiently accurate, which in turn does depend on the smoothness of g (see [10] for Runge-Kutta error bounds for parabolic equations in terms of the data).

About the differential equation (1) we assume that A is *sectorial*: there exist real constants M and σ and an angle $\varphi < \frac{\pi}{2}$ such that the resolvent is bounded by

$$\|(\lambda + A)^{-1}\| \leq \frac{M}{|\lambda - \sigma|}, \qquad \text{for} \quad |\arg(\lambda - \sigma)| \leq \pi - \varphi. \tag{2}$$

Here $\|\cdot\|$ is the operator norm corresponding to a vector norm, also denoted by $\|\cdot\|$. Clearly, for a symmetric positive semi-definite matrix A the bound (2) holds in the Euclidean norm with $\sigma = 0$ and $M = 1/\sin\varphi$ for any positive angle φ. More generally, condition (2) includes also non-symmetric operators such as those arising in convection-diffusion equations. In many situations resolvent bounds (2) in L^p norms are known to be inherited from the continuous problem by finite differences or finite elements, uniformly in the spatial discretization parameter (see, e.g., [1, 2]).

In Section 2 we review the discrete variation-of-constants formula for implicit Runge-Kutta methods, and in Section 3 we describe the discretization of the contour integrals for the rational approximations to the matrix exponential. The fast algorithm is given in Section 4, including an extension to systems with a mass matrix. A numerical example illustrates the performance of the algorithm in Section 5. Finally, Section 6 analyses the error of the contour integral discretization, which is the only error source in the algorithm.

2 The discrete variation-of-constants formula

In this preparatory section we recall the discrete variation-of-constants formula for implicit Runge-Kutta methods; cf., e.g., [3].

An implicit m-stage Runge-Kutta method applied to (1) yields, at $t_n = nh$, an approximation u_n to $u(t_n)$, given recursively by

$$v_{ni} = u_n + h \sum_{j=1}^{m} a_{ij} \Big(-Av_{nj} + g(t_n + c_j h) \Big), \quad 1 \le i \le m, \tag{3}$$

$$u_{n+1} = u_n + h \sum_{j=1}^{m} b_j \Big(-Av_{nj} + g(t_n + c_j h) \Big). \tag{4}$$

The method is determined by its coefficients a_{ij}, b_j, c_i $(i, j = 1, \dots, m)$. We denote the Runge-Kutta matrix by $\mathcal{A} = (a_{ij})$ and the row vector of the weights by $b^T = (b_j)$. Eliminating the internal stages v_{ni} results in

$$u_{n+1} = r(-hA)u_n + h \sum_{i=1}^{m} q_i(-hA)\, g(t_n + c_i h)\,, \qquad n \ge 0, \tag{5}$$

where the rational approximation $r(z)$ to e^z is defined by

$$r(z) = 1 + z b^T (I - z\mathcal{A})^{-1} \mathbb{1} \tag{6}$$

with $\mathbb{1} = (1, \dots, 1)^T$, and where the rational functions $q_i(z)$ are the entries of the row vector[1]

$$q(z) = \big(q_1(z), \dots, q_m(z) \big) = b^T (I - z\mathcal{A})^{-1}. \tag{7}$$

We assume that the eigenvalues of the Runge-Kutta matrix $\mathcal{A}$ have positive real part, and that the method is L-stable, i.e.,

$$|r(z)| \le 1 \quad \text{for} \quad \operatorname{Re} z \le 0, \qquad \text{and} \qquad r(\infty) = 0. \tag{8}$$

These conditions are in particular satisfied by the Radau IIA family of Runge-Kutta methods [6].

The discrete analogue of the variation-of-constants formula

$$u(t) = e^{-tA} u_0 + \int_0^t e^{-(t-\tau)A}\, g(\tau)\, d\tau$$

[1] Instead of taking $r(z)$ and $q_i(z)$ as rational functions originating from a Runge-Kutta method, another suitable choice would be $r(z) = e^z$ and $q_i(z) = \int_0^1 e^{(1-\theta)z}\, \ell_i(\theta)\, d\theta$, where ℓ_i is the ith Lagrange polynomial corresponding to the Gauss nodes c_j. This could be used similarly in the algorithm below.

is obtained by solving the recurrence relation (5). With the column vector $g_j = \big(g(t_j + c_i h)\big)_{i=1}^m$, this becomes

$$u_n = r(-hA)^n u_0 + h \sum_{j=0}^{n-1} r(-hA)^{n-1-j}\, q(-hA)\, g_j\,, \qquad n \geq 1. \tag{9}$$

3 Discretization of the contour integrals

We now discretize the Cauchy integral representation

$$r(-hA)^n q(-hA) = \frac{1}{2\pi \mathrm{i}} \int_\Gamma (\lambda + A)^{-1}\, r(h\lambda)^n\, q(h\lambda)\, d\lambda \tag{10}$$

along suitable contours Γ in the resolvent set of $-A$. The numerical integration in (10) is done by applying the trapezoidal rule with equidistant steps to a parametrization of a hyperbola [8]. With one contour and one set of quadrature points on this contour, we do not have a uniformly good approximation for all $n = 0, \ldots, N$, but we can instead obtain a uniform approximation locally on a sequence of geometrically growing intervals

$$I_\ell = [B^{\ell-1}h, B^\ell h), \qquad \ell \geq 1, \tag{11}$$

where the base $B > 1$ is an integer, e.g., $B = 10$. For $nh \in I_\ell$ we approximate the contour integrals (10) as

$$\begin{aligned} &r(-hA)^n q(-hA) \\ &\qquad \approx \sum_{k=-K}^{K} w_k^{(\ell)} (\lambda_k^{(\ell)} + A)^{-1} r(h\lambda_k^{(\ell)})^n\, q(h\lambda_k^{(\ell)})\,, \quad nh \in I_\ell, \end{aligned} \tag{12}$$

with the quadrature points $\lambda_k^{(\ell)}$ lying on a hyperbola Γ_ℓ and with the corresponding weights $w_k^{(\ell)}$. The number of quadrature points on Γ_ℓ, $2K+1$, is chosen independent of ℓ. The contour Γ_ℓ is chosen as a hyperbola given by

$$\mathbb{R} \to \Gamma_\ell : \ \theta \mapsto \gamma_\ell(\theta) = \mu_\ell\,(1 - \sin(\alpha + i\theta)) + \sigma \tag{13}$$

with an ℓ-dependent parameter $\mu_\ell > 0$. The angle α satisfies $0 < \alpha < \frac{\pi}{2} - \varphi$ with φ of (2), and σ is the shift in (2). The weights and quadrature points in (12) are given by

$$w_k^{(\ell)} = \frac{i\tau}{2\pi}\, \gamma_\ell'(\theta_k)\ , \quad \lambda_k^{(\ell)} = \gamma_\ell(\theta_k) \qquad \text{with} \qquad \theta_k = k\tau\ ,$$

where τ is a step length parameter that can be chosen independent of ℓ.

The following bound of the necessary number of quadrature points is a consequence of the error analysis in Section 6.

Theorem 1 *In* (12)*, a quadrature error bounded in norm by* ε *for* $nh \in I_\ell$ *is obtained with*

$$K = O(\log \tfrac{1}{\varepsilon})\,.$$

This holds for $n \geq c\,\log(1/\varepsilon)$*, with some constant* $c > 0$*. The required number* K *is independent of* ℓ *and of* n *and* $h \leq h_0$ *with* $nh \leq T$*. For* $\sigma \leq 0$*,* K *is also independent of the length* T *of the time interval.* K *depends on the angle* φ*, the bound* M *and the shift* σ *in* (2)*, but is otherwise independent of* A*.*

The approximation is, however, poor for the first few n; cf. also [13].

Concerning the choice of parameters we remark that the above asymptotic bound for K is obtained with $1/\tau$ proportional to $\log(1/\varepsilon)$ and with the parameter μ_ℓ for the contour Γ_ℓ chosen such that $\mu_\ell B^\ell h = c_1 \log(1/\varepsilon)$ with c_1 independent of ℓ and h, e.g., with $c_1 = 1/4$. Since perturbations in the terms of (12) can be magnified with $r(h\kappa_\ell)^n \approx e^{\kappa_\ell nh}$ with $\kappa_\ell = \mu_\ell(1 - \sin\alpha) + \sigma$, the factor c_1 should not be chosen too large. We refer to [9] for an optimized strategy to choose the parameters.

4 The fast algorithm

We start from the discrete variation-of-constants formula (9) for the Runge-Kutta approximation u_N with a fixed N. For the expression $r(hA)^N u_0$ we use the discretization of the Cauchy integral like in the previous section and in fact similarly to the approach of [14] for computing $\exp(-tA)u_0$.

The novel algorithm is concerned with the treatment of the inhomogeneity. For a fixed step number N and a given base B we split the sum in (9) into L sums, where L is the smallest integer such that $N \leq B^L$:

$$u_N = u_N^{(0)} + \cdots + u_N^{(L)}$$

with $u_N^{(0)} = hq(-hA)g_{N-1}$ and

$$u_N^{(\ell)} = h \sum_{(N-1-j)h \in I_\ell} r(-hA)^{N-1-j}\, q(-hA)\, g_j$$

for $\ell \geq 1$. On inserting the integral representation (10) we obtain, with $n_\ell = N - B^\ell$ for $0 \leq \ell \leq L-1$ and $n_L = 0$,

$$u_N^{(\ell)} = h \sum_{j=n_\ell}^{n_{\ell-1}-1} \frac{1}{2\pi \mathrm{i}} \int_{\Gamma_\ell} (\lambda + A)^{-1}\, r(h\lambda)^{N-1-j}\, q(h\lambda)\, g_j\, d\lambda.$$

The integral is discretized with the quadrature formula of Section 3: we approximate $u_N^{(\ell)}$ by $U_N^{(\ell)}$ given as

$$\begin{aligned} U_N^{(\ell)} &= h \sum_{j=n_\ell}^{n_{\ell-1}-1} \sum_{k=-K}^{K} w_k^{(\ell)} \, (\lambda_k^{(\ell)} + A)^{-1} \, r(h\lambda_k^{(\ell)})^{N-1-j} \, q(h\lambda_k^{(\ell)}) \, g_j \\ &= \sum_{k=-K}^{K} w_k^{(\ell)} \, r(h\lambda_k^{(\ell)})^{N-n_{\ell-1}} \, (\lambda_k^{(\ell)} + A)^{-1} \, y_k^{(\ell)}, \end{aligned}$$

where

$$y_k^{(\ell)} = h \sum_{j=n_\ell}^{n_{\ell-1}-1} r(h\lambda_k^{(\ell)})^{n_{\ell-1}-1-j} \, q(h\lambda_k^{(\ell)}) \, g_j.$$

Comparing this formula with (9), we see that $y_k^{(\ell)}$ is the Runge-Kutta approximation to the solution at time $t = n_{\ell-1}h$ of the linear initial-value problem

$$y'(t) = \lambda_k^{(\ell)} y(t) + g(t), \qquad y(n_\ell h) = 0, \tag{14}$$

and hence $y_k^{(\ell)}$ is computed by Runge-Kutta time-stepping on (14), using (5) with the scalar $h\lambda_k^{(\ell)}$ in place of the operator $-hA$. With the solutions $x_k^{(\ell)}$ of the linear systems of equations

$$(\lambda_k^{(\ell)} + A) \, x_k^{(\ell)} = y_k^{(\ell)}, \tag{15}$$

we obtain $U_N^{(\ell)}$ as the linear combination

$$U_N^{(\ell)} = \sum_{k=-K}^{K} c_k^{(\ell)} \, x_k^{(\ell)} \qquad \text{with} \quad c_k^{(\ell)} = w_k^{(\ell)} \, r(h\lambda_k^{(\ell)})^{B^{\ell-1}}. \tag{16}$$

There are only $(K+1)L$ linear systems (15) to be solved, for $k = 0, \ldots, K$ and $\ell \le L$. (Since the quadrature points lie symmetric with respect to the real axis, only the sum of the real parts of half the terms in (16) needs to be computed when approximating solutions with real components.) We recall $L - 1 \le \log_B N$ and $K = O(\log(1/\varepsilon))$, where ε is the accuracy requirement in the discretization of the contour integrals. Note that the only approximation made in the computation of $U_N^{(\ell)}$, is the discretization of the contour integrals.

Because of the poor approximation of the contour integral (10) for small n, we evaluate $U_N^{(0)} + U_N^{(1)}$ by B direct Runge-Kutta steps up to time $t = Nh$ for the initial value problem

$$v'(t) + Av(t) = g(t), \qquad v((N-B)h) = 0. \tag{17}$$

This requires the solution of another mB linear systems with matrices of the form $(\lambda + A)$. For small values of B or stringent accuracy requirements, we take B^2 direct Runge-Kutta steps to compute $u_N^{(0)} + u_N^{(1)} + u_N^{(2)}$. (Asymptotically, we need to take $O(\log(1/\varepsilon))$ direct steps according to Theorem 1.)

Finally we sum up the $U_N^{(\ell)}$ to obtain

$$U_N = U_N^{(0)} + \cdots + U_N^{(L)} \tag{18}$$

as the approximation to u_N. The fast algorithm thus consists of doing the steps (14)–(18) in the given order.

Remark 1 The algorithm extends to differential equations with a positive definite mass matrix M,

$$Mu'(t) + Au(t) = g(t), \qquad u(0) = u_0, \tag{19}$$

which is transformed to a system $\widetilde{u}'(t) + \widetilde{A}\widetilde{u}(t) = \widetilde{g}(t)$ for $\widetilde{u}(t) = M^{1/2}u(t)$ with $\widetilde{A} = M^{-1/2}AM^{-1/2}$ and $\widetilde{g}(t) = M^{-1/2}g(t)$. Applying *formally* the above algorithm to the transformed system and then transforming back yields again (16), where now $x_k^{(\ell)}$ is the solution of the linear system

$$(\lambda_k^{(\ell)} M + A)\, x_k^{(\ell)} = y_k^{(\ell)}, \tag{20}$$

and $y_k^{(\ell)}$ is the Runge-Kutta approximation at $t = n_{\ell-1}h$ of the initial value problem (14) with the untransformed inhomogeneity $g(t)$.

Remark 2 We have formulated the algorithm for a constant time step size h, but this is not essential. The algorithm is readily extended to accommodate variable step sizes, with the same step size sequence for all k in (14), chosen adaptively according to the behaviour of the inhomogeneity $g(t)$. Adaptivity in space can be used in solving the linear systems (15), choosing the spatial mesh according to the behaviour of the right-hand sides $y_k^{(\ell)}$ and the operator A. Note that in a hierarchical basis representation, adding a mesh point just corresponds to adding a scalar differential equation in (14). The details of such an adaptive algorithm are beyond the scope of this paper.

5 Numerical experiment

We consider an initial-boundary value problem of the heat equation in two space dimensions for $u = u(x,t)$,

$$\begin{cases} \partial_t u(x,t) &= \Delta u(x,t), & x \in \Omega,\ 0 \le t \le T, \\ u(x,0) &= 0, & x \in \Omega, \\ \partial_\nu u(x,t) &= 0, & x \in \Gamma_{int},\ 0 \le t \le T, \\ \partial_\nu u(x,t) &= \beta(x,t) - \rho(u(x,t) - u_{out}), & x \in \Gamma_{out},\ 0 \le t \le T, \end{cases}$$

on a wire-fence like structure (rectangle of size 10.65×12.64 with hexagonal holes, each hole with radius 0.8), see Figure 1. Here Γ_{int} denotes the boundary of the holes, and Γ_{out} is the boundary of the rectangle. In the example we set the heat flux $\beta = 5\sin^2(t)$ on the upper and left boundary of the rectangle and $\beta = 0$ on the lower and right boundary, and the convective heat flux to $\rho(u - u_{out})$, with the ambient temperature $u_{out} = 0$ and the coefficient of surface heat transfer $\rho = 0.5$, cf. the introduction in [7]. Space is discretized using linear finite elements on a triangular mesh, with 27346 vertices and 50368 triangles. Triangulation is done using the tool Triangle [16].

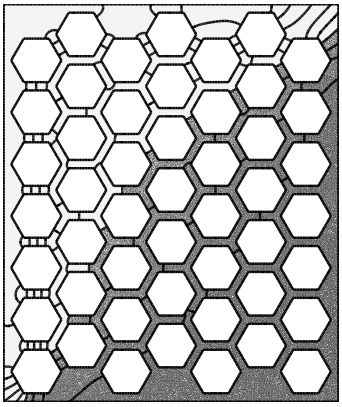

Fig. 1 Domain for the heat equation, with isolines of the temperature distribution at $t = 20$.

The finite element equations are of the form (19), where M is the standard mass matrix containing the L^2 inner products of the nodal basis functions φ_i. The stiffness matrix is the sum $A = A_0 + \rho M_b$ with

$$A_0\big|_{ij} = \int_\Omega \nabla\varphi_i \nabla\varphi_j \, dx \ , \quad M_b\big|_{ij} = \int_{\Gamma_{out}} \varphi_i \varphi_j \, d\sigma \ .$$

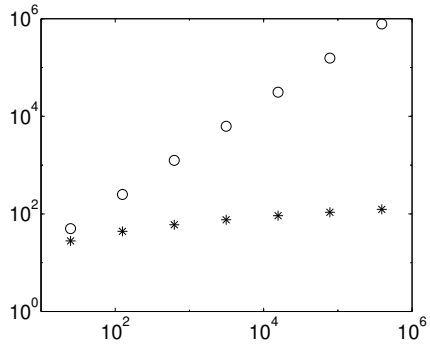

Fig. 2 Number of solves of linear systems versus step number: direct time-stepping (∘) and fast algorithm (∗).

The inhomogeneity $g(t)$ is given by

$$g_i(t) = \int_{\Gamma_{out}} (\beta(x,t) + \rho u_{out})\varphi_i \, d\sigma(x).$$

The algorithm takes into account that $g(t)$ has nonzero entries only along the outer boundary Γ_{out}, so that effectively $g(t)$ is a vector whose dimension is the number of degrees of freedom on the outer boundary – in this example 776. The differential equations (14) need to be integrated only for this reduced dimension, since they have no coupling between the components.

We have used the 2- and 3-stage Radau IIA methods (of orders 3 and 5, respectively) for time discretization in our numerical experiments.

In the fast algorithm we set $B = 5$ and $K = 15$ and, from the experience of [9,13], we choose the angle in the hyperbola as $\alpha = \pi/4$, the parameter $\mu_\ell = 3/(hB^\ell)$ and the parameter $\tau = 5/K$. This choice of parameters leads to a deviation of the order 10^{-6} from the Runge-Kutta approximation at time $t = 20$.

The two-dimensional example above is still small enough that a direct solution of the linear systems using sparse solvers is reasonable. A direct implementation of the m-stage Radau IIA method (cf. [6]) requires only m sparse LU factorizations, computed at the beginning of the integration, followed by mN substitutions. On the other hand, for the algorithm presented here we need to solve $(K+1)(L-1)$ linear systems with matrices $\lambda M + A$ for as many different values of λ, and the mB linear systems for the B direct steps. Especially with a diagonal, lumped mass matrix $M = DD^T$, this work can be reduced by a similarity transform taking $D^{-1}AD^{-T}$ to tridiagonal

(or Hessenberg) form T, but exploiting sparsity here becomes an issue; see [4,12]. The resulting linear systems with $\lambda I + T$ are then inexpensive to solve. Even without using such a transform, the fast algorithm eventually overtakes the standard algorithm for sufficiently large step numbers N, in the present example for $N \approx 1000$. Much earlier and larger relative gains arise when iterative solvers are used for the linear systems in both algorithms, as is clear from the linear systems count in Figure 2.

6 Error analysis

Our analysis relies on the good behaviour of the trapezoidal rule for certain holomorphic integrands [8,17,18]. Following the ideas in [8], we consider the continuation of the parametrization (13) to the conformal mapping

$$\gamma(w) = \mu\,(1 - \sin(\alpha + iw)). \tag{21}$$

(For ease of presentation we set $\sigma = 0$ in (2).) This conformal mapping transforms each horizontal straight line

$$\operatorname{Im} w = y, \qquad -d \le y \le d,$$

with $0 < \alpha - d < \alpha + d < \frac{\pi}{2}$, into the left branch of the hyperbola

$$\lambda \in \mathbb{C}: \ \left(\frac{\operatorname{Re}\lambda - \mu}{\mu \sin(\alpha - y)}\right)^2 - \left(\frac{\operatorname{Im}\lambda}{\mu\cos(\alpha - y)}\right)^2 = 1,$$

i.e., the left branch of the hyperbola with center at $(\mu, 0)$, foci at $(0,0)$, $(2\mu, 0)$ and with asymptotes forming angles $\pm[\pi/2 - (\alpha - y)]$ with the real axis. Therefore, γ transforms the horizontal strip

$$D_d = \{w \in \mathbb{C} : |\operatorname{Im} w| \le d\}$$

into the region $\Omega = \gamma(D_d)$ limited by the left branches corresponding to $y = \pm d$. To indicate the dependence on the parameter μ of (21), we write $\Omega = \Omega_\mu$. We note that $\lambda \in \Omega_\mu$ if and only if $h\lambda \in \Omega_{h\mu}$ for any $h > 0$, so that

$$h\Omega_\mu = \Omega_{h\mu}.$$

Because of (2), henceforth we will assume that $\alpha > 0$ and $d > 0$ satisfy $0 < \alpha - d < \alpha + d < \frac{\pi}{2} - \varphi$. Under these conditions, all the hyperbolas we are considering lie outside the spectrum of $-A$.

After parametrizing (10) via γ, we get

$$r(-hA)^n\, q(-hA) = \int_{-\infty}^{+\infty} G_{h,n}(x)\, dx,$$

where $G_{h,n}(w)$ is given, for $w \in D_d$, by

$$G_{h,n}(w) = \frac{1}{2\pi i}\Big(\gamma(w) + A\Big)^{-1} r(h\gamma(w))^n \, q(h\gamma(w)) \, \gamma'(w). \tag{22}$$

For an integrable mapping $G : \mathbb{R} \to X$, $K \geq 1$ and $\tau > 0$, set

$$E_{\tau,K}(G) = \int_{-\infty}^{+\infty} G(x)\, dx - \tau \sum_{k=-K}^{K} G(k\tau), \tag{23}$$

i.e., $E_{\tau,K}(G)$ stands for the quadrature error of the truncated trapezoidal rule for the integral of G. Our goal is precisely to estimate $E_{\tau,K}(G_{h,n})$. To this end we first consider the behaviour of $G_{h,n}$ on D_d. We need the following lemma whose elementary proof is omitted.

Lemma 1 *Let $r(z)$ be a rational function with $r(0) = 1$, $r'(0) = 1$ which satisfies the L-stability condition (8). Then, there exist $\rho > 0$ and $b > 0$ such that*

$$|r(z)| \leq \frac{e^{2\delta}}{1 + b\,|z|}, \qquad \textit{for} \;\; z \in \Omega_\delta \;\; \textit{with} \;\; 0 < \delta \leq \rho. \tag{24}$$

Now, from the sectorial condition (2) on A and Lemma 1 with $\delta = h\mu \leq \rho$, we obtain

$$\|G_{h,n}(x + iy)\| \leq C_0 \, \frac{e^{2\mu h n}}{(1 + bh\mu(\cosh x - \sin(\alpha - y)))^n} \tag{25}$$

for $x \in \mathbb{R}$ and $|y| \leq d$ (recall that $0 < \alpha - d < \alpha + d < \frac{\pi}{2} - \varphi$), where C_0 is the constant given by

$$C_0 = \frac{M}{2\pi}\sqrt{\frac{1 + \sin(\alpha + d)}{1 - \sin(\alpha + d)}} \;\; \max_{z \in \Omega_\rho} \|q(z)\|.$$

Finally, the above bound (25), the elementary inequality

$$1 + c - s \geq (1 - s)(1 + c), \qquad c, s > 0,$$

and the bound 1 for the sine yield, for $|y| \leq d$ and $t = nh$,

$$\|G_{h,n}(x + iy)\| \leq \frac{C_0\, e^{2\mu t}}{(1 - b\mu t/n)^n} \Big(1 + \frac{b\mu t}{n} \cosh x\Big)^{-n}. \tag{26}$$

Next, to estimate $E_{\tau,K}(G_{h,n})$, we are going to use an approach similar to the one in [8,17,18]. We denote by $S(D_d, X)$ the class formed by all the continuous mappings $G : D_d \to X$ (for a complex

Banach space X, here a space of matrices) holomorphic on the interior of the strip D_d, and satisfying the following two conditions:

$$\int_{-d}^{d} \|G(x+iy)\|\, dy \to 0, \quad \text{as } |x| \to +\infty, \tag{27}$$

$$N(G, D_d) := \int_{-\infty}^{+\infty} \{\|G(x+id)\| + \|G(x-id)\|\}\, dx < +\infty. \tag{28}$$

Given $G \in S(D_d, X)$, it turns out, assuming that G has a fast decay at ∞, that $E_{\tau,K}(G)$ becomes very small as $K \to +\infty$ if τ is properly tuned (see [8,17,18] for various situations). In Theorem 2 we assume that G exhibits the kind of decay of $G_{h,n}$ in (26) and this theorem will directly provide the estimate for $E_{\tau,K}(G_{h,n})$ we are looking for.

Theorem 2 *Assume that $G \in S(D_d, X)$ for some $d > 0$, and that there exist $C, a > 0$ and $n \geq 1$ such that*

$$\|G(x)\| \leq C\Big(1 + \frac{a}{n}\cosh x\Big)^{-n}, \qquad x \in \mathbb{R}. \tag{29}$$

Then, for $\tau > 0$, $K \geq 1$, there holds

$$\|E_{\tau,K}(G)\| \leq \frac{N(G, D_d)}{e^{2\pi d/\tau} - 1} + C\left(\phi(a)\, e^{-a\cosh(K\tau)/2} + \Big(1 + \frac{a}{n}\cosh K\tau\Big)^{-(n-1)}\right),$$

with $\phi(a) = 2 + |\log(1 - e^{-a/2})|$.

Notice that ϕ is decreasing, $\phi(a) \to 2$ as $a \to +\infty$ and $\phi(a) \sim |\log a|$ as $a \to 0^+$.

Proof Set

$$E_{\tau,\infty}(G) = \int_{-\infty}^{+\infty} G(x)\, dx - \tau \sum_{k=-\infty}^{\infty} G(k\tau), \qquad \tau > 0.$$

For fixed $K \geq 1$, it is clear that

$$\|E_{\tau,K}(G)\| \leq \|E_{\tau,\infty}(G)\| + \tau \sum_{|k| \geq K+1} \|G(k\tau)\|.$$

On the one hand, by Theorem 4.1 in [17] (see also [18]), we have

$$\|E_{\tau,\infty}(G)\| \leq \frac{N(G, D_d)}{e^{2\pi d/\tau} - 1}.$$

On the other hand,

$$\tau \sum_{|k|\geq K+1} \|G(k\tau)\| \leq 2C\tau \sum_{k=K+1}^{+\infty} \Big(1+\frac{a}{n}\cosh k\tau\Big)^{-n}$$
$$\leq 2C \int_{K\tau}^{+\infty} \Big(1+\frac{a}{n}\cosh x\Big)^{-n} dx.$$

The proof of the theorem is now completed by applying the following lemma. □

Lemma 2 *For $R \geq 0$, $a > 0$ and $n \geq 1$ there holds*

$$\int_R^{+\infty} \Big(1+\frac{a}{n}\cosh x\Big)^{-n} dx \leq \phi(a)\, e^{-a\cosh R/2} + \Big(1+\frac{a}{n}\cosh R\Big)^{-(n-1)}.$$

Proof The change of variables $u = \cosh x$ shows that

$$\int_R^{+\infty} \Big(1+\frac{a}{n}\cosh x\Big)^{-n} dx = \int_{\cosh R}^{+\infty} \Big(1+\frac{a}{n}u\Big)^{-n} \frac{du}{\sqrt{u^2-1}}\,.$$

Set $\beta = \max\{\cosh R, n/a\}$. Then, from the estimates in [8] and the elementary inequality

$$(1+y/n)^{-n} \leq e^{-y/2}, \qquad \text{for } \ 0 \leq y \leq n \tag{30}$$

it turns out that

$$\int_{\cosh R}^{\beta} \Big(1+\frac{a}{n}u\Big)^{-n} \frac{du}{\sqrt{u^2-1}} \leq \int_{\cosh R}^{\beta} e^{-au/2} \frac{du}{\sqrt{u^2-1}}$$
$$\leq \int_R^{+\infty} e^{-a\cosh x/2}\, dx$$
$$\leq \phi(a)\, e^{-a\cosh R/2}\,.$$

Moreover,

$$\int_\beta^{+\infty} \Big(1+\frac{a}{n}u\Big)^{-n} \frac{du}{\sqrt{u^2-1}}$$
$$\leq \Big(1+\frac{a}{n}\cosh R\Big)^{-(n-1)} \int_\beta^{+\infty} \Big(1+\frac{a}{n}u\Big)^{-1} \frac{du}{\sqrt{u^2-1}}\,.$$

Now, since $\beta \geq \max\{1, n/a\}$, the result follows from the observation that for both $n/a \geq 1$ and $n/a \leq 1$ we have

$$\int_{\max\{1,n/a\}}^{+\infty} \Big(1+\frac{a}{n}u\Big)^{-1} \frac{du}{\sqrt{u^2-1}} \leq \int_1^{+\infty} (1+v)^{-1} \frac{dv}{\sqrt{v^2-1}} = 1. \qquad \square$$

We apply Theorem 2 to $G_{h,n}$. First of all, notice that by (26) it is clear that $G_{h,n}$ satisties (27). Moreover, by Lemma 2, we have

$$N(G_{h,n}, D_d) \le \frac{4C_0\, e^{2\mu t}}{(1-b\mu t/n)^n} \times\left(\phi(b\mu t)\, e^{-b\mu t/2} + \left(1+\frac{b\mu t}{n}\right)^{-(n-1)}\right), \tag{31}$$

and conclude that $G_{h,n} \in S(D_d, X)$. Then, in view of (26) and (31), Theorem 2 yields directly

$$\|E_{\tau,K}(G_{h,n})\| \le \frac{4C_0\, e^{2\mu t}}{(1-b\mu t/n)^n}\left(\frac{\phi(b\mu t)\, e^{-b\mu t/2} + (1+b\mu t/n)^{-(n-1)}}{e^{2\pi d/\tau}-1} + \phi(b\mu t)\, e^{-b\mu t\cosh(K\tau)/2} + \left(1+\frac{b\mu t}{n}\cosh(K\tau)\right)^{-(n-1)}\right).$$

A simplified version of this estimate is obtained by using the elementary inequalities (30) and

$$\begin{aligned} (1-y/n)^{-n} &\le e^{2y}, && \text{for } 0 \le y \le n/2,\\ \phi(y) &\le 3, && \text{for } y \ge 1. \end{aligned}$$

Setting

$$C = 20\, C_0,\ \ a_0 = 2+\tfrac{3}{2}b,\ \ a_1 = 2+2b,\ \ a_2 = \tfrac{1}{2}b,$$

with b of Lemma 1 as before, we can summarize the final result in the following theorem.

Theorem 3 *The quadrature error* (23) *for* $G_{h,n}$ *of* (22) *with* (21) *satisfies, for* $t = nh$ *and if* $n/2 \ge b\mu t \ge 1$,

$$\|E_{\tau,K}(G_{h,n})\| \le C\left(\frac{e^{a_0\mu t}}{e^{2\pi d/\tau}-1} + e^{(a_1 - a_2\cosh(K\tau))\mu t} + e^{a_1\mu t}\left(1+\frac{b\mu t}{n}\cosh(K\tau)\right)^{-(n-1)}\right).$$

The first term in the error bound becomes $O(\varepsilon)$ if τ is chosen so small that $a_0\mu t - 2\pi d/\tau \le \log\varepsilon$, which requires an asymptotic proportionality

$$\frac{1}{\tau} \sim \log\frac{1}{\varepsilon} + \mu t.$$

For μ chosen such that

$$\frac{c_1}{B}\log\frac{1}{\varepsilon} \le \mu t \le c_1 \log\frac{1}{\varepsilon}$$

with an arbitrary positive constant c_1 and with $B > 1$, we obtain that the second term is $O(\varepsilon)$ if $a_1 - a_2\cosh(K\tau) \le -B/c_1$, i.e., with

$$\cosh(K\tau) = c_2$$

for a sufficiently large constant c_2. With the above choice of τ, this yields

$$K \sim \log\frac{1}{\varepsilon}.$$

The third term then becomes smaller than ε for

$$n \ge c\,\log\frac{1}{\varepsilon}$$

with a sufficiently large constant c. Taken together, these estimates prove Theorem 1.

Acknowledgements. The research of the first and third authors has been supported by DGI-MCYT under project MTM2004-07194 co-financed by FEDER funds. The research of the second author has been supported by DFG, SFB 382. The reserach of the fourth author has been supported by the DFG Research Center MATHEON "Mathematics for key technologies" in Berlin.

References

1. A. Ashyralyev and P. Sobolevskii, Well-Posedness of Parabolic Difference Equations. Birkhäuser, Basel, 1994.
2. N. Y. Bakaev, V. Thomée, and L. Wahlbin, Maximum-norm estimates for resolvents of elliptic finite element operators. Math. Comp. 72 (2002), 1597–1610.
3. P. Brenner, M. Crouzeix, V. Thomée, Single step methods for inhomogeneous linear differential equations in Banach space. RAIRO Modél. Math. Anal. Numér. 16 (1982), 5–26.
4. I.A. Cavers, A hybrid tridiagonalization algorithm for symmetric sparse matrices. SIAM J. Matrix Anal. Appl. 15 (1994), 1363–1380.
5. I.P. Gavrilyuk, V. Makarov, Exponentially convergent algorithms for the operator exponential with applications to inhomogeneous problems in Banach spaces. Preprint, 2004.
6. E. Hairer, G. Wanner, Solving Ordinary Differential Equations. II. Stiff and Differential-Algebraic Problems. Second edition. Springer, Berlin, 1996.
7. R. W. Lewis, K. Morgan, H.R. Thomas, K.N. Seetharamu, The Finite Element Method in Heat Transfer Analysis. John Wiley & Sons Ltd, Chichester, 1996.

8. M. López-Fernández, C. Palencia, On the numerical inversion of the Laplace transform of certain holomorphic mappings. Appl. Numer. Math. 51 (2004), 289-303.
9. M. López-Fernández, C. Palencia, A. Schädle, On the numerical inversion of the Laplace transform of certain holomorphic mappings, Addendum. (In preparation).
10. C. Lubich, A. Ostermann, Runge-Kutta methods for parabolic equations and convolution quadrature. Math. Comput. 60 (1993), 105–131.
11. C. Lubich, A. Schädle, Fast convolution for nonreflecting boundary conditions. SIAM J. Sci. Comp. 24 (2002), 161–182.
12. J.L. Nikolajsen, An improved Laguerre eigensolver for unsymmetric matrices. SIAM J. Sci. Comp. 22 (2000), 822–834.
13. A. Schädle, M. López-Fernández, C. Lubich, Fast and oblivious convolution quadrature. Preprint, 2005.
14. D. Sheen, I. H. Sloan, V. Thomée, A parallel method for time-discretization of parabolic problems based on contour integral representation and quadrature. Math. Comp. 69 (2000), 177–195.
15. D. Sheen, I. H. Sloan, V. Thomée, A parallel method for time discretization of parabolic equations based on Laplace transformation and quadrature. IMA J. Numer. Anal. 23 (2003), 269–299.
16. J. R. Shewchuk, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, in Applied Computational Geometry: Towards Geometric Engineering, Eds. M. C. Lin and D. Manocha, Lecture Notes in Computer Science 1148, Springer, 1996, 203–222.
17. F. Stenger, Approximations via Whittaker's cardinal function. J. Approx. Theory 17 (1976), 222–240.
18. F. Stenger, Numerical methods based on Whittaker cardinal, or sinc functions. SIAM Review 23 (1981), 165–224.
19. A. Talbot, The accurate numerical inversion of Laplace transforms. J. Inst. Math. Appl. 23 (1979), 97–120.