Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Multistep and multistage boundary integral methods for the wave equation

(revised version: July 2010)

by

Lehel Banjai

Preprint no.: 58 2009

MULTISTEP AND MULTISTAGE CONVOLUTION QUADRATURE FOR THE WAVE EQUATION: ALGORITHMS AND EXPERIMENTS

LEHEL BANJAI*

Abstract. We describe how time-discretized wave equation in a homogeneous medium can be solved by boundary integral methods. The time discretization can be a multistep, Runge-Kutta, or a more general multistep-multistage method. The resulting convolutional system of boundary integral equations belongs to the family of convolution quadratures of Ch. Lubich.

The aim of this work is two-fold. It describes an efficient, robust, and easily parallelizable method for solving the semi-discretized system. The resulting algorithm has the main advantages of time-stepping methods and of Fourier synthesis: at each time-step a system of linear equations with the same system matrix needs to be solved, yet computations can easily be done in parallel, the computational cost is almost linear in the number of time-steps, and only the Laplace transform of the time-domain fundamental solution is needed. The new aspect of the algorithm is that all this is possible without ever explicitly constructing the weights of the convolution quadrature. This approach also readily allows the use of modern data-sparse techniques to perform computation in space efficiently. We investigate theoretically and numerically to which extent hierarchical matrix ($\mathcal{H}$-matrix) techniques can be used to speed up the space computation.

The second aim of the article is to perform series of large scale 3D experiments with a range of multistep and multistage time discretization methods: backward difference formula of order 2 (BDF2), Trapezoid rule, and the 3-stage Radau IIA methods are investigated in detail. One of the conclusions of the experiments is that the Radau IIA method often performs overwhelmingly better than the linear multistep methods, especially for problems with many reflections, yet, in connection with hyperbolic problems backward difference formulas have so far been predominant in the literature on convolution quadrature.

Key words. wave equation, boundary integral equations, convolution quadrature, multistep methods, Runge-Kutta methods, hierarchical matrices

AMS subject classifications. 35L05, 65M38

1. Introduction.

The use of boundary integral methods in the numerical solution of elliptic equations has a long and successful history; see for example the books [21, 27, 34, 37, 39]. Although time domain boundary integral representations of hyperbolic problems, e.g., the wave equation, have been known for a long time, the scientific community has been slower in accepting the corresponding numerical methods. Main reasons are difficulties with dealing with distributional fundamental solutions, expensive computation, and above all, stability problems for longer time computations. Nevertheless, many good methods do exist and are being used, see for example [15, 19], and the review [11].

In 1994 Ch. Lubich [32] has introduced the so-called convolution quadrature for discretizing time-convolutions arising from the boundary integral representation of the wave equation. This method inherits the unconditional stability properties of the underlying A-stable time discretization method and requires only the Laplace transform of the time-domain fundamental solution, this being a much simpler function to compute with. These favourable properties of the convolution quadrature have sparked a recent theoretical interest [31] and interest in efficiently implementing and experimentally investigating these methods for the wave equation [7, 23, 24, 28], viscoelasticity and poroelasticity [38], and Maxwell equations [41]. So far, however, large scale, long time computations have not been reported on and only the backwards difference formulas (BDF) have been used for the time discretization.

*Max-Planck Institute for Mathematics in the Sciences, Inselstrasse 22, 04103 Leipzig, Germany (banjai@mis.mpg.de).

In this paper we describe an efficient, almost linear time algorithm $\mathcal{O}(N \log^2 N)$, in number of time-steps N, that is easily parallelizable. It has most of the favourable properties of time-stepping methods:

- the solution is obtained in a time-stepping manner,
- at each time-step, a linear system, or a small group of linear systems, with the same system matrix, respectively small group of system matrices, needs to be solved

and of Fourier synthesis:

- the algorithm is easily parallelizable in time,
- only the Laplace transform of the fundamental solution is needed.

The algorithm is obtained by combining in a new way the recursive idea in [25] and the use of scaled Fast Fourier Transform (FFT). The most important aspect which differentiates the new algorithm from the original one in [25] is that the intermediate step of constructing the convolution weights is completely avoided. The advantage of the recursive algorithm compared to the pure decoupling approach taken in [7], which also makes use of scaled FFTs, are the time-stepping aspects listed above.

This algorithm allows easy implementation of A-stable linear multistep, Runge-Kutta, and more general multistep-multistage methods for the time discretization. This made the second important contribution of this work possible: a systematic and extensive numerical experiments. We perform 3D computations with the BDF2, Trapezoid, and Radau IIA methods. The numerical experiments show great stability properties for long time computations. As to the relative merits of the different time discretization methods, the 3-stage Radau IIA method, in our experiments, performed overwhelmingly better than the multistep methods even at moderate accuracies for complicated scattering problems with many reflections. For scattering by unit sphere, i.e. a convex scatterer, the linear multistep methods performed as well as the Runge-Kutta method at moderate accuracies. Note, however, that for the multistep method it was essential to use a direct integral formulation of the problem.

We also discuss the fast computation of spatially discretized operators that need to be inverted. We show that the discretization of these operators can efficiently be represented by an $\mathcal{H}$-matrix [18] and further that the (approximate) LU-decomposition of the matrix can also be efficiently computed in this format. In the recursive algorithm, matrix-vector products with discretizations of high-frequency Helmholtz integral operators need also be computed. These can be performed efficiently using fast multipole methods with diagonal multipole expansions [36]. Some modifications to the standard fast multipole algorithms is however needed, description of which falls out of the scope of the present work.

Questions of convergence and stability of the time and space discretization have been fully answered for linear multistep methods in [32] and for Runge-Kutta methods in [5]; in the latter reference spatial discretization has not been considered, but techniques from [32] are directly applicable. The general results on linear multistep methods in [32] do not cover the case of Trapezoid rule for our application. Therefore, in the appendix, we close this gap in the theory.

The paper is organised as follows. In the first section after this introduction we describe the boundary integral formulation of a time-discretized wave equation. The description holds for a large family of multistep-multistage time discretization methods. In Section 3 we show how A-stable linear multistep and A-stable Runge-Kutta methods fit in this description. Section 4 describes the efficient algorithm for the solution of the resulting discrete convolution system of integral operators. Section 5 briefly discusses the Galerkin discretization in space, gives an analysis of the use of $\mathcal{H}$-matrices for Helmholtz operators with complex frequencies, and mentions the use of fast multipole methods for high frequency problems. In Section 6 details of the numerical experiments are given.

An extended abstract of the work described here has been published in the proceedings of the 2009 ICNAAM conference [3].

2. Time discretization and the boundary integral formulation. Let us consider the problem of finding u which solves the wave equation

$$\begin{aligned} \partial_t^2 u(x,t) - \Delta u(x,t) = 0, \qquad & (x,t) \in \Omega^c \times [0,T], \\ u(x,0) = \partial_t u(x,0) = 0, \qquad & x \in \Omega^c, \\ u(x,t) = g(x,t), \qquad & (x,t) \in \Gamma \times [0,T], \end{aligned} \tag{2.1}$$

where $\Omega \subset \mathbb{R}^3$ is a bounded domain, $\Omega^c = \mathbb{R}^3 \setminus \overline{\Omega}$, $\Gamma = \partial\Omega$, and g is the given boundary data. For smooth, compatible data $g(\cdot,t) \in H^{1/2}(\Gamma)$, a unique solution $u(\cdot,t) \in H^1(\Omega)$ exists.

Let a time discretization at equally spaced points $t_j = j\Delta t$, $j = 0,1,\dots,N$, of the ordinary differential equation (ODE) $y' = \mu y$ be given by:

$$\frac{1}{\Delta t} \sum_{j \le n} \delta_{n-j} Y_j - \mu Y_n = 0, \tag{2.2}$$

Here $\delta_j \in \mathbb{R}^{m\times m}$ are matrices and the vector $Y_j \in \mathbb{R}^m$ is an approximation of the solution at m-stages at time t_j; namely it is an approximation to $y(t_j + c_\ell \Delta t)$, for some $c_1, c_2, \dots, c_m \in \mathbb{R}$. The approximation to $y(t_n)$ is given as a combination of the entries of the Y_js:

$$y_{n+1} = \sum_{j \le n} \gamma_{n-j} Y_j, \qquad \gamma_j \in \mathbb{R}^{1\times m}. \tag{2.3}$$

Linear multistep, Runge-Kutta, and multistage-multistep methods all fit this description.

For later use, let us also define the generating functions of the time discretization:

$$\delta(\zeta) = \sum_{j=0}^{\infty} \delta_j \zeta^j, \qquad \gamma(\zeta) = \sum_{j=0}^{\infty} \gamma_j \zeta^j.$$

We apply above described time discretization to (2.1), where we extend $u(\cdot)$ by zero to negative times, thereby obtaining a semi-discretized system:

$$\begin{aligned} \frac{1}{\Delta t^2} \sum_{j=0}^{n} \delta_{n-j}^{(2)} U_j(x) - \Delta U_n(x) = 0, \qquad & x \in \Omega^c, \\ U_n(x) = G_n(x), \qquad & x \in \Gamma, \end{aligned} \qquad n = 0,1,\dots,N, \tag{2.4}$$

where $(G_n)_\ell$ is $g(\cdot, t_n + c_\ell \Delta t)$ or an approximation of it and the coefficients $\delta_j^{(2)}$ are given by

$$(\delta(\zeta))^2 = \sum_{j=0}^{\infty} \delta_j^{(2)} \zeta^j,$$

reflecting the need to discretize the second derivative. At each time step, the resulting PDE is uniquely solvable if $(G_n)_\ell \in H^{1/2}(\Gamma)$ and the spectrum of $\delta_0 = \delta(0) \in \mathbb{R}^{m\times m}$ is strictly in the right-half complex plane; note for $m = 1$, the condition reads $\delta_0 > 0$.

Making use of generating functions for both U_j and G_j, we can rewrite (2.4) as

$$\begin{aligned} \left(\frac{\delta(\zeta)}{\Delta t}\right)^2 U(\zeta) - \Delta U(\zeta) = 0, \qquad x \in \Omega^c, \\ U(\zeta) = G(\zeta), \qquad x \in \Gamma, \end{aligned} \tag{2.5}$$

where to avoid questions of convergence, we can assume that $G_j \equiv 0$ for large enough j.

We recognise in (2.5) a family of Helmholtz problems. Since the forcing term is zero, we can apply the indirect boundary integral method and write the solution $U(\zeta)$ as a single layer potential:

$$U(\zeta)(x) = (\mathcal{S}\,(\delta(\zeta)/\Delta t)\,\Phi(\zeta))\,(x), \qquad x \in \Omega^c.$$

The single layer potential $\mathcal{S}(s)$ being defined by

$$(\mathcal{S}(s)\varphi)(x) := \int_\Gamma \frac{e^{-s|x-y|}}{4\pi|x-y|}\varphi(y)d\Gamma_y, \qquad x \in \mathbb{R}^3 \setminus \Gamma.$$

Its restriction to the boundary we denote by $\mathcal{V}(s)$:

$$(\mathcal{V}(s)\varphi)(x) := \int_\Gamma \frac{e^{-s|x-y|}}{4\pi|x-y|}\varphi(y)d\Gamma_y, \qquad x \in \Gamma.$$

As a function of $\operatorname{Re} s \geq \sigma_0 > 0$, $\mathcal{V}(s)$ is an analytic function, with, see [2, 31],

$$\|\mathcal{V}(s)\|_{H^{-1/2}(\Gamma)\to H^{1/2}(\Gamma)} \leq C(\sigma_0)\frac{|s|}{\operatorname{Re} s} \tag{2.6}$$

and

$$\|\mathcal{V}(s)^{-1}\|_{H^{1/2}(\Gamma)\to H^{-1/2}(\Gamma)} \leq C(\sigma_0)\frac{|s|^2}{\operatorname{Re} s}. \tag{2.7}$$

Thus, $\mathcal{V}(\delta(\zeta)/\Delta t)$ is well defined if the spectrum of $\delta(\zeta)$ is strictly in the right-half plane. As we will see later, the time discretization methods we use will satisfy this condition for all $|\zeta| < 1$.

Therefore, we need to find the boundary density $\Phi(\zeta)$ such that the boundary condition is satisfied:

$$G(\zeta)(x) = \mathcal{V}\,(\delta(\zeta)/\Delta t)\,\Phi(\zeta)(x), \qquad x \in \Gamma. \tag{2.8}$$

Let $W_j^{\Delta t}(\mathcal{V})$ be the *convolution weights* defined by the generating function

$$\mathcal{V}\,(\delta(\zeta)/\Delta t) = \sum_{j=0}^\infty W_j^{\Delta t}(\mathcal{V})\zeta^j. \tag{2.9}$$

Matching coefficients in (2.8) we obtain a discrete convolution that needs to be satisfied by the unknown densities Φ_j:

$$\sum_{j=0}^n W_{n-j}^{\Delta t}(\mathcal{V})\Phi_j = G_n, \qquad \text{on } \Gamma, \quad n = 0, 1, \ldots, N. \tag{2.10}$$

Since $W_0^{\Delta t}(\mathcal{V}) = \mathcal{V}(\delta(0)/\Delta t)$, the spectrum of $\delta(0)$ is in the right-half plane, and (2.7) holds, $W_0^{\Delta t}(\mathcal{V}) : (H^{-1/2}(\Gamma))^m \to (H^{1/2}(\Gamma))^m$ is invertible and the above semi-discrete system has a unique solution for sufficiently smooth boundary data; $(G_n)_\ell \in H^{1/2}(\Gamma)$ suffices.

Let us note that the above semi-discrete system is the convolution quadrature based on time discretization (2.2) for the time-domain boundary integral equation

$$\int_0^t \int_\Gamma \frac{\delta(t-\tau-|x-y|)}{4\pi|x-y|}\varphi(y,\tau)d\Gamma_y d\tau = g(x,t), \qquad t \in [0,T],\ x \in \Gamma, \tag{2.11}$$

where $\delta(\cdot)$ is the Delta distribution; see [32]. Convolution quadrature is usually presented as a method for discretizing convolutional integral operators, but the above connection with a time discretization of the underlying partial differential equation is well known; see, e.g., Theorem 5.2 in [32].

Finally, it is important to note that we could equally well have applied the direct boundary integral method to obtain an integral formulation of the semi-discretized system. In this case the unknown densities Φ_j would be approximations of the Neumann data $\partial_\nu u(\cdot, t_j)|_\Gamma$ and would have to solve

$$\sum_{j=0}^n W_{n-j}^{\Delta t}(\mathcal{V})\Phi_j = -\frac{1}{2}G_n + \sum_{j=0}^n W_{n-j}^{\Delta t}(\mathcal{D})G_j, \qquad \text{on } \Gamma, \quad n = 0,1,\dots,N, \tag{2.12}$$

where $\mathcal{D}(s)$ is the double layer potential defined by

$$\mathcal{D}(s)\varphi(x) := \int_\Gamma \frac{\partial}{\partial \nu_y}\frac{e^{-s|x-y|}}{4\pi|x-y|}\varphi(y)d\Gamma_y, \qquad x \in \Gamma.$$

3. Examples of time discretizations.

3.1. Linear multistep methods. Multistep time discretization of time domain boundary integral operators of the wave equation have been investigated in detail in [32]. There, under the condition of A-stability of the underlying multistep method, it is shown that under some smoothness and compatibility conditions on the Dirichlet data g optimal stability and convergence properties can be proved. In this article we will make use of two multistep methods, both of second order and both A-stable:

$$\text{Backwards difference formula of order 2 (BDF2): } \delta(\zeta) = \frac{3}{2} - 2\zeta + \frac{1}{2}\zeta^2,$$

$$\text{Trapezoid rule: } \delta(\zeta) = \frac{2(1-\zeta)}{1+\zeta}.$$

Note that A-stability of multistep methods, i.e., $\operatorname{Re}\delta(\zeta) > 0$ for $|\zeta| < 1$, implies that (2.8) is solvable for all $|\zeta| < 1$.

In [32] the stability and second order convergence of BDF2 scheme has been proved. The general theory in [32] can however not be applied to the Trapezoid discretization of the time domain boundary integral operators of the wave equation. For completeness, we give in the appendix a proof of the convergence specific to the Trapezoid rule which can be applied to our case.

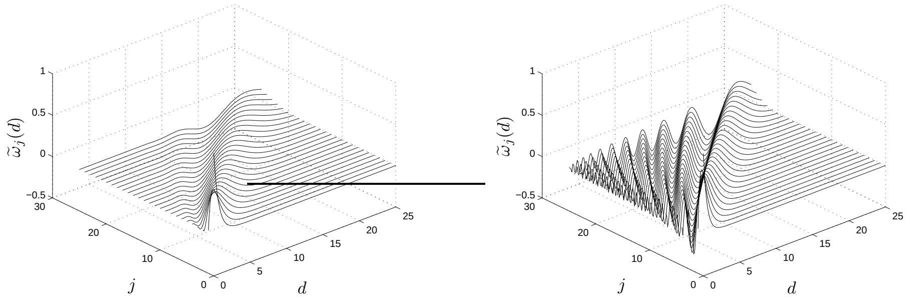

FIG. 3.1. *Kernel functions $\widetilde{\omega}_j(d)$ for the BDF2 (left) and the Trapezoid rule (right).*

Let us look more closely at the boundary integral operators $\omega_j^{\Delta t}(\mathcal{V}) : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma)$ and their dependence on the multistep method. These operators have the form

$$\omega_j^{\Delta t}(\mathcal{V})\varphi = \int_\Gamma \frac{\widetilde{\omega}_j(|x-y|/\Delta t)}{4\pi|x-y|}\varphi(y)d\Gamma_y,$$

where $\widetilde{\omega}_j : \mathbb{R}_{\geq 0} \to \mathbb{R}$ are given by the following generating function

$$e^{-\delta(\zeta)d} = \sum_{j=0}^{\infty} \widetilde{\omega}_j(d)\zeta^j. \tag{3.1}$$

Kernel functions $\widetilde{\omega}_j$ are depicted in Figure 3.1.

Various interesting properties of the multistep methods can be read from Figure 3.1. For both methods $\widetilde{\omega}_j(d)$ goes to zero quickly when $d > j$. This reflects the finite speed of propagation of waves: at time $j\Delta t$ the wave has not yet traveled the distance $d\Delta t = |x-y|$. For the BDF2 method we also see that if j is sufficiently larger than d, $\widetilde{\omega}_j(d)$ is again close to zero: this reflects Huygens' principle present in three dimensions; note that in two dimensions or if some dissipation is present this would not be the case. Interestingly this property is lost with the Trapezoid rule. This can be traced back to the Trapezoid rule, unlike BDF2, having a generating function $\delta(\zeta)$ with an infinite number of terms. It is, however, more illuminating to think of the oscillating tail of $\widetilde{\omega}_j(d)$ as the consequence of the lack of L-stability of the Trapezoid rule. This both suggests that the Trapezoid rule may be more susceptible to stability problems, e.g., spurious oscillations, and that in this case we seem not to be able to make use of the Huygens' principle to reduce the computational costs.

The above comments are only of descriptive nature and we do not attempt to prove them here. For the BDF2 method, the statements have been proved in [23] and estimates are given for regions of discrete time-space where $\widetilde{\omega}_j(d)$ can be approximated by 0 without affecting the accuracy of the overall computation. In [23] it is shown that by doing this sparsification, matrices obtained by Galerkin space discretization are sparse; by sparse, we mean here that gain can be made from using sparse techniques to store and solve the resulting linear systems and not that there are $\mathcal{O}(1)$ entries in each row of the $M \times M$ matrix, for more details see [23]. In [9] the fact that for BDF2 $\widetilde{\omega}_j(d)$ converges exponentially to zero once $j > d$ is used to investigate a convolution quadrature method with reduced costs where all the convolution weights ω_j with $j\Delta t > C\,\mathrm{diam}(\Omega)$, for some constant C, are replaced by 0.

3.2. Runge-Kutta methods. Let us consider a Runge-Kutta method which applied to the initial value problem

$$y' = \mu y, \qquad y(0) = y_0,$$

with a step size $\Delta t > 0$ gives an approximation y_n at time $t_n = n\Delta t$,

$$\begin{aligned} Y_n &= y_n \mathbb{1} + \Delta t \mu A Y_n, \\ y_{n+1} &= y_n + \Delta t \mu b^T Y_n, \end{aligned} \tag{3.2}$$

with $A \in \mathbb{R}^{m\times m}, b \in \mathbb{R}^{m\times 1}$, and $\mathbb{1} = (1,\dots,1)^T$. The method is said to have classical order p if $y_1 - y(t_1) = O(\Delta t^{p+1})$ and stage order q if $Y_{0i} - y(c_i \Delta t) = O(\Delta t^q)$, $i = 1, 2, \dots, m$. A standard reference for Runge-Kutta methods as presented in this section is [26].

We assume that the method is A-stable, i.e., that the stability function

$$R(z) = 1 + zb^T(I - zA)^{-1}\mathbb{1} \tag{3.3}$$

is bounded by

$$|R(z)| \le 1, \quad \text{for } \operatorname{Re} z \le 0 \text{ and } I - zA \text{ is non-singular for all } \operatorname{Re} z \le 0. \tag{3.4}$$

From (3.2) it follows that

$$y_{n+1} = (1 - b^T A^{-1}\mathbb{1})y_n + b^T A^{-1} Y_n = R(\infty)y_n + b^T A^{-1} Y_n = \sum_{j=0}^{n} R(\infty)^{n-j} b^T A^{-1} Y_j,$$

hence $\gamma(\zeta) = \sum_{j=0}^{\infty} R(\infty)^j b^T A^{-1} \zeta^j$. Rearranging the two equations in (3.2) gives the discrete convolutional equation for the Y_n:

$$\frac{1}{\Delta t}(Y_n - Y_{n-1}) - \mu\left(AY_n - (A - \mathbb{1}b^T)Y_{n-1}\right) = 0, \qquad n = 0, \dots, N. \tag{3.5}$$

Proceeding by using generating functions, we see that Runge-Kutta discretization also fits the general description (2.2), with

$$\delta(\zeta) = \left(A + \frac{\zeta}{1-\zeta}\mathbb{1}b^T\right)^{-1}. \tag{3.6}$$

Using the Sherman-Morrison-Woodbury formula we obtain

$$\delta(\zeta) = A^{-1} - \frac{\zeta}{1-\zeta}\frac{A^{-1}\mathbb{1}b^T A^{-1}}{1 + \frac{\zeta}{1-\zeta} b^T A^{-1}\mathbb{1}},$$

and hence $\delta(\zeta)$ is well defined unless $\frac{\zeta}{1-\zeta} b^T A^{-1}\mathbb{1} = -1$. For $|\zeta| < 1$ this can only happen if $b^T A^{-1}\mathbb{1} \in (-\infty, 0) \cup (2, +\infty)$. On the other hand, A-stability implies that $|R(\infty)| \le 1$, and hence $b^T A^{-1}\mathbb{1} = 1 - R(\infty)$ must be contained in $[0, 2]$. Therefore, $\delta(\zeta)$ is well defined for all $|\zeta| < 1$.

We would like to have that $\delta(\zeta) \in \mathbb{R}^{m\times m}$ has all eigenvalues $\mu_j(\zeta)$ satisfying $\operatorname{Re} \mu_j(\zeta) > 0$ for $|\zeta| < 1$. It is not immediately obvious, as in the multistep methods, that A-stability implies this property so we prove this next.

LEMMA 3.1. *Let (3.4) hold, $|\zeta| \neq 1$, and μ be an eigenvalue of $\delta(\zeta)$, but not of A^{-1}. Then $R(\mu) = \zeta^{-1}$.*

Proof. Let $\delta(\zeta)v = \mu v$, $v \neq 0$. Let us first assume that $b^T v \neq 0$. Then, from the definition of $\delta(\zeta)$, see (3.6), we have that

$$\begin{aligned} v &= \mu A v + \mu \frac{\zeta}{1-\zeta} \mathbb{1} b^T v, \\ \frac{1-\zeta}{\zeta} v &= b^T v \mu (I - \mu A)^{-1} \mathbb{1}, \\ \frac{1-\zeta}{\zeta} b^T v &= b^T v (R(\mu) - 1). \end{aligned}$$

Dividing by $b^T v$ and rearranging gives the result.

If $b^T v = 0$ then

$$\delta(\zeta) A v = v = \frac{\delta(\zeta)}{\mu} v.$$

Since $\delta(\zeta)$ is invertible we obtain that μ is an eigenvalue of A^{-1} contradicting the assumption of the lemma. □

COROLLARY 3.2. *If (3.4) holds, then for $|\zeta| < 1, \zeta \in \mathbb{C}$, eigenvalues of $\delta(\zeta)$ have positive real part.*

Proof. The corollary follows directly from the previous lemma and the A-stability (3.4) condition. □

REMARK 3.3. *For a pth-order Runge-Kutta method the approximation property $R(z) = e^z + \mathcal{O}(z^{p+1})$, for $z \to 0$, holds. Though not strict, in practice a useful lower bound for $Re\,\mu$ is given by $\log \frac{1}{|\zeta|}$.*

For convenience, i.e., to be able to easily compute analytic functions of $\delta(\zeta)$, we would also like $\delta(\zeta)$ to be diagonalizable. In the next proposition we discuss the diagonalizability of $\delta(\zeta)$ for the 2-stage Radau IIA method.

PROPOSITION 3.4. *For the 2-stage Radua IIA method, $\delta(\zeta)$ is diagonalizable for all $|\zeta| < 1$ except for $\zeta = 3\sqrt{3} - 5$.*

Proof. The statement can be checked by computing the eigenvalues of $A + \frac{\zeta}{1-\zeta} \mathbb{1} b^T$. If there are 2 distinct eigenvalues, then $\delta(\zeta)$ is diagonalizable. The 2-stage Radau IIA method is defined by

$$A = \begin{pmatrix} 5/12 & -1/12 \\ 3/4 & 1/4 \end{pmatrix}, \qquad b = \begin{pmatrix} 3/4 \\ 1/4 \end{pmatrix}.$$

Linearly independent vectors $(1\ 1)^T$ and $(1\ -3)^T$ are eigenvectors of $\mathbb{1} b^T$ with eigenvalues 1 and 0 respectively. Computing the eigenvalues of $A + \frac{\zeta}{1-\zeta} \mathbb{1} b^T$ is then equivalent to computing the eigenvalues of

$$\begin{pmatrix} 1/2 & 1/2 \\ -1/6 & 1/6 \end{pmatrix} + \frac{\zeta}{1-\zeta} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

An easy computation shows that the above matrix has distinct eigenvalues for all $|\zeta| < 1$ except for $\zeta = 3\sqrt{3} - 5$. A further computation shows that for this value of ζ, $\delta(\zeta)$ happens not to be diagonalizable. □

REMARK 3.5. *A similar, but more tedious calculation, can be done for the 3-stage Radau IIA method. As for the 2-stage case we first perform a simplifying change of*

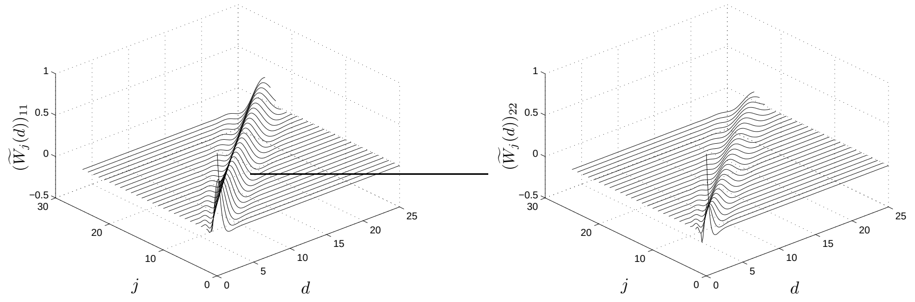

FIG. 3.2. *Kernel functions $(\widetilde{W}_j(d))_{11}$ and $(\widetilde{W}_j(d))_{22}$ for the 2-stage Radau IIA method. The off-diagonal entries have a similar shape.*

basis. Subsequent calculation is easier if here we use the W-transformation as defined in Section IV.5 of [26]. This shows that the eigenvalues of $\delta(\zeta)^{-1}$ are roots of

$$z^3 - z^2\left(\frac{3}{5} + \frac{\zeta}{1-\zeta}\right) + z\left(\frac{3}{20} + \frac{1}{10}\frac{\zeta}{1-\zeta}\right) - \frac{1}{60}\frac{\zeta}{1-\zeta} - \frac{1}{60} = 0.$$

Further calculation, gives that for $\zeta = 0.004598175\cdots + i0.069213506\dots$, $\delta(\zeta)$ and $\delta(\bar{\zeta})$ are not diagonalizable so that $|\zeta| = 0.069366077\dots$ should be avoided. Other values of ζ for which $\delta(\zeta)$ is not diagonalizable are outside of the unit circle. Let us also note here that in [3] it was mistakenly stated that $\delta(\zeta)$ is diagonalizable for all $|\zeta| < 1$; what should have been stated was that for $|\zeta| < 1$ but close enough to 1, $\delta(\zeta)$ is diagonalizable.

In practice, and for a general Runge-Kutta method, it is not essential to perform such calculations since it is is highly unlikely that during the computation a value of ζ for which $\delta(\zeta)$ is not diagonalizable will be required. Still, it is advisable to investigate the condition number of arising matrices $\delta(\zeta)$.

In Figure 3.2 we visualize two entries of the matrix function $\widetilde{W}_j(d)$ for the 2-stage Radau IIA method. Here, as in the multistep case, $\widetilde{W}_j : \mathbb{R}_{\geq 0} \to \mathbb{R}^{m\times m}$ is defined by

$$e^{-\delta(\zeta)d} = \sum_{j=0}^{\infty} \widetilde{W}_j(d)\zeta^j.$$

Properties similar to the BDF2 method can be seen, except that the functions are more concentrated along the diagonal $d \approx j$.

Let us note here that the expression for $\delta(\zeta)$ and $\gamma(\zeta)$ can be simplified if we assume the Runge-Kutta method to be L-stable; Radau IIA methods being an example. For these methods $R(\infty) = 0$, implying

$$\delta(\zeta) = A^{-1} - \zeta A^{-1}\mathbb{1}b^T A^{-1}, \qquad \gamma(\zeta) = b^T A^{-1}.$$

Although we will not use such methods in this paper, let us just remark that a multistep-multistage method is also an option. For example for the two-step Runge-Kutta method of [12] the generating function is given by

$$\delta(\zeta) = \left(\frac{\zeta}{1-\zeta}\mathbb{1}v^T + \frac{\zeta}{1-\zeta}\mathbb{1}w^T + A + \zeta B\right)^{-1}.$$

and

$$\gamma(\zeta) = \left(1 - \zeta + \zeta(v^T + \zeta w^T)(A + B\zeta)^{-1}\mathbb{1}\right)^{-1} (v^T + \zeta w^T)(A + B\zeta)^{-1}.$$

We have implemented the 2-step, 3-stage, A-stable method described in [12] and tested it on the example described in Section 6.1. At useful accuracies it did not perform as well as the 3-stage Radau IIA method, though eventually at very high accuracies, due to its higher stage order, it did outperform the Radau IIA. In view of the results of this experiment, we have decided not to investigate the method further.

4. Efficient solution of lower Triangular Toeplitz systems. To obtain the solution of (2.10) one could solve the lower triangular system by forward substitution:

$$\Phi_n = (W_0^{\Delta t}(\mathcal{V}))^{-1} \left(G_n - \sum_{j=0}^{n-1} W_{n-j}^{\Delta t}(\mathcal{V})\Phi_j \right), \qquad n = 0, 1, \dots, N.$$

The total cost in terms of N would be $\mathcal{O}(N^2)$ and at each time step n, a linear equation $W_0^{\Delta t}\Phi_n = \widetilde{G}_n$ would have to be solved. We wish to reduce the computational cost to $\mathcal{O}(N)$, up to powers of $\log N$, but still keep the favourable property of needing for each time-step to solve a linear equation with the same operator but different right-hand side.

Solving the semi-discrete system (2.10) can be thought of as solving a lower triangular Toeplitz system with the jth lower diagonal of the Toeplitz matrix given by $W_j^{\Delta t}(\mathcal{V})$. Therefore in this section we will consider the fast solution of systems $T_L(\mathbf{a})\mathbf{x} = \mathbf{b}$, where $T_L(\mathbf{a})$ is the lower triangular Toeplitz matrix whose first column is given by the vector $\mathbf{a}$. Similarly we define the upper triangular Toeplitz matrix $T_U(\mathbf{a})$, circulant matrix $C(\mathbf{a})$, $\mathbf{a}$ being the first column in both, and Toeplitz matrix $T(\mathbf{a}_1, \mathbf{a}_2)$ with first column $\mathbf{a}_1$ and first row $\mathbf{a}_2$. Further,

$$a(\zeta) = \sum_{j=0}^{\infty} a_j \zeta^j$$

denotes the generating function, generating the coefficients of the vector $\mathbf{a} \in \mathbb{R}^{J+1}$ for any $J \geq 0$.

First we show how to approximate the lower triangular Toeplitz matrix by a, up to a change of basis, diagonal matrix. This method has already been used in the present context in [7], though presented differently; see also [6].

Lemma 4.1. *Let $0 < \lambda < 1$ and a_j be such that $\sum_{j=0}^{\infty} |\lambda a_j| < \infty$. Define $\Lambda = \operatorname{diag}(1, \lambda, \dots, \lambda^J)$ and $F \in \mathbb{C}^{J+1 \times J+1}$ to be the Fourier matrix given by*

$$(F\mathbf{x})_j = \sum_{\ell=0}^{J} x_\ell \zeta_{J+1}^{-j\ell}, \qquad \text{with } \zeta_{J+1} = e^{\frac{2\pi i}{J+1}}, 0 \leq j \leq J.$$

Then

$$T_L(\mathbf{a}) = \Lambda^{-1} F^{-1} \operatorname{diag}(\mathbf{a}_\lambda) F \Lambda + \lambda^{J+1} T_U(\mathbf{a}') - \lambda^{J+1} \left(\sum_{j=0}^{\infty} \lambda^j a_{j+J+1} \Lambda^{-1} E^j \Lambda \right),$$

where $\mathbf{a} = (a_0, a_1, \dots, a_J)^T$, $\mathbf{a}_\lambda = \left(a(\lambda), a(\lambda\zeta_{J+1}^{-1}), \dots, a(\lambda\zeta_{J+1}^{-J})\right)^T$, $\mathbf{a}' = (0, a_J, a_{J-1}, \dots, a_1)^T$, and $E = C(\mathbf{e_2})$ where $\mathbf{e_2} \in \mathbb{R}^{J+1}$ is the unit vector with $\mathbf{e_2} = (0, 1, 0, \dots, 0)^T$. Note *that* $E^{J+1} = I$ *and*

$$\|\Lambda^{-1}E^j\Lambda\|_\infty = \|\Lambda^{-1}E^j\Lambda\|_1 = \lambda^{-j(\operatorname{mod} J+1)}, \qquad j = 0, 1, \dots, \infty,$$

where $\|\cdot\|_\infty$ *and* $\|\cdot\|_1$ *are the infinity norm (largest row sum) and the one norm (largest column sum).*

Proof. First, from the identity $\sum\limits_{j=0}^{n} a_{n-j}x_j = \lambda^{-n}\sum\limits_{j=0}^{n}\lambda^{n-j}a_{n-j}\lambda^j x_j$ and further easy calculation we see that

$$T_L(\mathbf{a}) = \Lambda^{-1}T_L(\Lambda\mathbf{a})\Lambda = \Lambda^{-1}C(\Lambda\mathbf{a})\Lambda + \lambda^{J+1}T_U(\mathbf{a}').$$

The circulant matrix $C(\Lambda\mathbf{a})$ is diagonalized by the Fourier transform

$$C(\Lambda\mathbf{a}) = F^{-1}\operatorname{diag}(F\Lambda\mathbf{a})F.$$

Finally

$$(F\Lambda\mathbf{a})_\ell = \sum_{j=0}^{J}\lambda^j a_j\zeta_{J+1}^{-\ell j} = a(\lambda\zeta_{J+1}^{-\ell}) - \sum_{j=J+1}^{\infty}\lambda^j a_j\zeta_{J+1}^{-\ell j}.$$

Hence

$$\begin{aligned} C(\Lambda\mathbf{a}) &= F^{-1}\operatorname{diag}(\mathbf{a}_\lambda)F - \sum_{j=J+1}^{\infty}\lambda^j a_j\left(F^{-1}\operatorname{diag}(1,\zeta_{J+1}^{-1},\dots,\zeta_{J+1}^{-J})F\right)^j \\ &= F^{-1}\operatorname{diag}(\mathbf{a}_\lambda)F - \sum_{j=J+1}^{\infty}\lambda^j a_j C(\mathbf{e_2})^j. \end{aligned}$$

Finally we notice that since $E^{J+1} = I$, where $E = C(\mathbf{e_2})$, we have that

$$\sum_{j=J+1}^{\infty}\lambda^j a_j\Lambda^{-1}E^j\Lambda = \lambda^{J+1}\sum_{j=0}^{\infty}\lambda^j a_{j+J+1}\Lambda^{-1}E^j\Lambda.$$

□

Hence, we see that to solve (2.10) for $n = 0, 1, \dots, J$, up to an accuracy $\mathcal{O}(\lambda^{J+1})$, it is sufficient to solve $J+1$ decoupled systems

$$\mathcal{V}\left(\delta(\lambda\zeta_{J+1}^{-\ell})/\Delta t\right)\widehat{\Phi}_\ell = \widehat{G}_\ell, \qquad \ell = 0, 1, \dots, J+1,$$

where

$$\widehat{G}_\ell = \sum_{j=0}^{J}\lambda^j G_j\zeta_{J+1}^{-j\ell}.$$

The overall computational complexity scales as $\mathcal{O}(Jm\log J)$; in the multistage case each operator would first be diagonalized, see Proposition 3.4 and Remark 3.4, but since in practice, m is small, the cost of diagonalization is negligible. Since these problems can be solved in parallel this can be a very effective way to solve the convolutional system (2.10). However, what is not immediately obvious is that some of these problems will be difficult to solve, yet to solve (2.10) by forward substitution it suffices to invert J times the same operator W_0. Next we combine the two ideas.

4.1. A recursive procedure. Let us assume that

$$T_L(\mathbf{a}^{(1)})\mathbf{x}^{(1)} = \mathbf{b}^{(1)} \tag{4.1}$$

with

$$\mathbf{a}^{(1)} = (a_0, a_1, \dots, a_{N_1}) \text{ and } \mathbf{b}^{(1)} = (b_0, b_1, \dots, b_{N_1})$$

has already been solved for $N_1 < N$. Then it remains to solve

$$T_L(\mathbf{a}^{(2)})\mathbf{x}^{(2)} = \mathbf{b}^{(2)} - T(P_{N_1}^{II}\mathbf{a}, P_{N_1}^{I}\mathbf{a})\mathbf{x}^{(1)} \tag{4.2}$$

with

$$\mathbf{a}^{(2)} = (a_0, a_1, \dots, a_{N-N_1}), \qquad \mathbf{b}^{(1)} = (b_{N_1+1}, b_{N_1+2}, \dots, b_N),$$

and

$$P_{N_1}^{I}\mathbf{a} = (a_{N_1+1}, a_{N_1}, \dots, a_1)^T, \qquad P_{N_1}^{II}\mathbf{a} = (a_{N_1+1}, a_{N_1+2}, \dots, a_N)^T.$$

Once the right-hand side in (4.2) is computed, the system is assumed to be solved recursively. The main difficulty now becomes the computation of the matrix-vector product $T(P_{N_1}^{II}\mathbf{a}, P_{N_1}^{I}\mathbf{a})\mathbf{x}^{(1)}$. Toeplitz matrix-vector products can efficiently be computed by using the FFT. This completes the description of the recursive method of solving the triangular system introduced in [25]. As the algorithm stands it seems necessary to know the coefficients a_j explicitly, i.e. in our case the convolution weights $W_j^{\Delta t}$, in order to compute their discrete Fourier transform. We next show how the computation of the discrete Fourier transformation of the coefficients (weights) can be entirely avoided, at the same time removing the need to compute the coefficients explicitly. This will constitute our modification of the algorithm of [25].

We first notice that

$$T(P_{N_1}^{II}\mathbf{a}, P_{N_1}^{I}\mathbf{a}) = \lambda^{-N_1-1}\Lambda_2^{-1}T(P_{N_1}^{II}\Lambda\mathbf{a}, P_{N_1}^{I}\Lambda\mathbf{a})\Lambda_1, \tag{4.3}$$

where $\Lambda_1 = \operatorname{diag}(1, \lambda, \dots, \lambda^{N_1})$, $\Lambda_2 = \operatorname{diag}(1, \lambda, \dots, \lambda^{N-N_1})$, and $\Lambda = \operatorname{diag}(1, \lambda, \dots, \lambda^{N})$. As usual, to compute the matrix-vector product with $T(P_{N_1}^{II}\Lambda\mathbf{a}, P_{N_1}^{I}\Lambda\mathbf{a})$ we extend the Toeplitz matrix to a circulant matrix which can then be diagonalized by discrete Fourier transforms. Next lemma gives the details.

Lemma 4.2. *Let*

$$\mathbf{c}_\lambda := \left(\lambda^{N_1+1}a_{N_1+1}, \dots, \lambda^N a_N, a_0, \dots, \lambda^{N_1}a_{N_1}\right)^T,$$
$$\mathbf{a}_\lambda := \left(a(\lambda), \zeta_{N+1}^{N_1+1}a(\lambda\zeta_{N+1}^{-1}), \dots, \zeta_{N+1}^{N(N_1+1)}a(\lambda\zeta_{N+1}^{-N}\right)^T.$$

Then, under the conditions of Lemma 4.1,

$$C(\mathbf{c}_\lambda) = F^{-1}\operatorname{diag}(\mathbf{a}_\lambda)F - E^{-N_1-1}\lambda^{N+1}\left(\sum_{j=0}^{\infty}\lambda^j a_{N+1+j}E^j\right),$$

where $E = C(\mathbf{e_2})$.

Proof. The proof is similar to the proof of Lemma 4.1. One only needs to take care of the technical difficulty that the entries in the vector $\mathbf{c}_\lambda$ are not in the usual order. The discrete Fourier transform of the permuted vector is given by

$$\begin{aligned}(F\mathbf{c}_\lambda)_\ell &= \sum_{j=0}^{N-N_1-1} \lambda^{j+N_1+1}\omega_{j+N_1+1}\zeta_{N+1}^{-j\ell} + \sum_{j=N-N_1}^{N} \lambda^{j-N+N_1}\omega_{j-N+N_1}\zeta_{N+1}^{-j\ell}\\ &= \zeta_{N+1}^{\ell(N_1+1)} \sum_{j=0}^{N} \lambda^j a_j \zeta_{N+1}^{-\ell j} = \zeta_{N+1}^{\ell(N_1+1)} \left(F\Lambda\mathbf{a}\right)_\ell .\end{aligned}$$

□

Above procedure can be continued recursively, where the lower triangular matrices $T_L(\mathbf{a}^1)$ and $T_L(\mathbf{a}^2)$ are again split in half; see [25]. The procedure is graphically depicted in Figure 4.1. The overall complexity of the recursive solution is $\mathcal{O}(Nm\log^2 N)$. While this is a larger number of operations than is required using only the decoupling procedure of Lemma 4.1, the operations involved are cheaper; see Section 5.2.

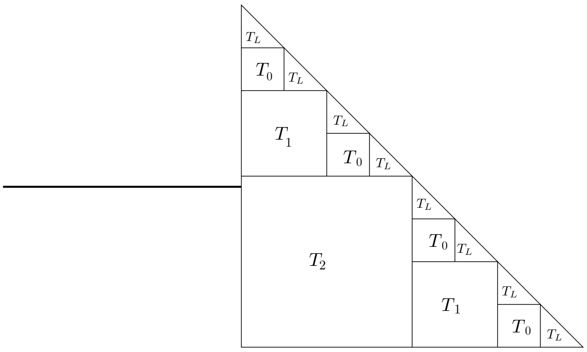

FIG. 4.1. *Depiction of a recursive procedure for solving lower triangular Toeplitz systems. Lower triangular Toeplitz matrices are denoted by T_L and general Toeplitz matrices by T_j.*

4.2. Influence of finite precision arithmetic and the choice of λ. The results in previous lemmas suggest that by taking $\lambda \to 0$ we can obtain any accuracy. This in practice cannot be true as due to finite precision arithmetic the computation of $\Lambda^{-1}F^{-1}DF\Lambda\mathbf{x}$ for a diagonal matrix D, vector $\mathbf{x}$, and $\Lambda = \operatorname{diag}(1, \lambda, \dots, \lambda^J)$, cannot be computed accurately if λ is too small. In fact if eps denotes the machine precision, we cannot hope for better accuracy than $\text{eps}\,\lambda^{-J} + \lambda^{J+1}$. In conclusion if the FFT and the values a_j are computed to an accuracy eps then the optimal choice of λ in both the decoupling procedure Lemma 4.1 and the recursive algorithm is $\lambda = \text{eps}^{\frac{1}{2J}}$. The highest precision obtainable in double precision is around 10^{-16}, hence λ should be chosen larger than $10^{-8/J}$ giving the that the highest achievable accuracy is $\sqrt{\text{eps}} \approx 10^{-8}$.

5. Space discretization. For the space discretization, we use a standard Galerkin boundary element method. Let $X \subset H^{-1/2}(\Gamma)$ be a finite-dimensional space with basis $\{b_1, b_2, \dots, b_M\}$. For the single-stage case the resulting discrete operators are clear: operators $\mathcal{V}(s), \omega_j^{\Delta t}$ and functions φ_j, g_j are replaced by matrices $\mathbf{V}(s)$, $\mathbf{A}_j$ and vectors $\mathbf{x}_j$, $\mathbf{g}_j$ given by

$$(\mathbf{V}(s))_{lk} := (\mathcal{V}(s)b_l, b_k)_{L^2(\Gamma)}, \qquad (\mathbf{A}_j)_{lk} := (\omega_j^{\Delta t} b_l, b_k)_{L^2(\Gamma)},$$

and

$$(\mathbf{x}_j)_l = (\varphi_j, b_l)_{L^2(\Gamma)}, \qquad (\mathbf{g}_j)_l = (g_j, b_l)_{L^2(\Gamma)}.$$

Then the discrete system we need to solve has the form:

$$\begin{pmatrix} \mathbf{A}_0 & 0 & \cdots & & 0 \\ \mathbf{A}_1 & \mathbf{A}_0 & \ddots & & \vdots \\ \vdots & \mathbf{A}_1 & \ddots & \ddots & \\ & & \ddots & \ddots & 0 \\ \mathbf{A}_N & \cdots & & \mathbf{A}_1 & \mathbf{A}_0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \vdots \\ \mathbf{x}_N \end{pmatrix} = \begin{pmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \vdots \\ \mathbf{g}_N \end{pmatrix}. \tag{5.1}$$

A few explanatory words should be said about the multistage case. For an m-stage method, let $\Sigma \in \mathbb{C}^{m\times m}$ be such that $\Sigma U = U \operatorname{diag}(s_1, s_2, \dots, s_m)$ with $\operatorname{Re} s_j > 0$ and U invertible; see Proposition 3.4 and Remark 3.4. Since $\mathcal{V}(s)$ is an analytic function for $\operatorname{Re} s > 0$ we can define

$$\mathcal{V}(\Sigma) = U \operatorname{diag}(\mathcal{V}(s_1), \mathcal{V}(s_2), \dots, \mathcal{V}(s_m))U^{-1}.$$

Space discretization of the above operator results in a $mM \times mM$ block matrix with blocks $\mathbf{V}^{ln}(\Sigma)$, $l, n \in \{1, 2, \dots, m\}$,

$$(\mathbf{V}^{ln}(\Sigma))_{ij} = (\mathcal{V}(\Sigma)_{ln} b_i, b_j) = \left(U \operatorname{diag}\left[(\mathbf{V}(s_1))_{ij}, \dots, (\mathbf{V}(s_m))_{ij}\right] U^{-1}\right)_{ln}.$$

Hence, in tensor notation we can write

$$\mathbf{V}(\Sigma) = (U \otimes I_M) \operatorname{diag}\left[\mathbf{V}(s_1), \mathbf{V}(s_2), \dots, \mathbf{V}(s_m)\right] (U^{-1} \otimes I_M),$$

with $I_M \in \mathbb{R}^{M\times M}$ being the identity matrix.

An important fact to take from the above expression is that to compute $\mathbf{V}(\Sigma)$, only m-integral operators need to be discretized and not m^2. Note, however, that the eigenvectors U do depend on Σ, therefore the total cost of construction of $V(\Sigma)$ is $\mathcal{O}(m^3)$ for the eigenvalue decomposition and $\mathcal{O}(mM^2)$ for the construction of matrices $V(s_j)$. The latter cost can be reduced to $\mathcal{O}(mM \log^a M)$, $a > 0$, with modern data-sparse techniques, but since m will always be a small number ($m \leq 3$ in our experiments), this will still be a much larger computation than the eigenvalue decomposition.

5.1. Fast data sparse techniques. Here we give a very brief introduction to hierarchical matrices as introduced by Wolfgang Hackbusch and co-authors, see the recent book [22] and references therein. In the description we will be guided by the problem of storage and matrix-vector computation of dense matrices $\mathbf{V}(s)$ described in the previous section.

5.1.1. Hierarchical matrices. Recall that for space discretization we use a finite dimensional basis of locally supported functions $\{b_1, \dots, b_M\} \subset H^{-1/2}(\Gamma)$. Further, let the boundary Γ be subdivided into M' disjoint panels π_j, $j = 1, \dots, M'$. Define $\mathcal{J} := \{1, \dots, M\}$.

DEFINITION 5.1. *Given a constant $C_{\text{leaf}} > 0$, a labelled tree $\mathcal{T}_\mathcal{J}$, is said to be a* **cluster tree** *for $\mathcal{J}$ if the following conditions hold:*

- *For each $\tau \in \mathcal{T}_\mathcal{J}$, the label denoted by $\hat{\tau}$ is a subset of $\mathcal{J}$. In particular, the label of the root of the tree is the cluster $\mathcal{J}$ containing all the indices.*

- *If $\tau \in \mathcal{T}_{\mathcal{J}}$ has sons, then the sons form a partition of τ, i.e., $\hat{\tau} = \dot{\bigcup}\{\hat{\tau'} : \tau' \in sons(\tau)\}$.*
- *For each leaf τ, $\#\hat{\tau} \leq C_{\text{leaf}}$.*

Let

$$\Gamma_\tau := \cup_{j\in\hat{\tau}}\{\pi_i \;:\; s.t.\; \pi_i \subset \operatorname{supp} b_j\} \subset \Gamma,$$

be the subset of Γ corresponding to a cluster $\tau \in \mathcal{T}_{\mathcal{J}}$.

Introduce a restriction operator $\chi_\tau : \mathbb{R}^{M\times M}$ for each $\tau \in \mathcal{T}_{\mathcal{J}}$ by

$$(\chi_\tau)_{kj} = \begin{cases} 1, & \text{if } k = j \in \hat{\tau}, \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

We call a pair of clusters (τ, σ) a *block*. The corresponding sub-block of the matrix $\mathbf{V}$ is then $\chi_\tau \mathbf{V} \chi_\sigma$. A hierarchical matrix approximation of $\mathbf{V}$ attempts to replace such blocks with low-rank approximations. Blocks for which we expect this to be possible are called *admissible blocks*. For matrices arising from Galerkin discretization of boundary integral operators the following admissibility property, controlled by a fixed parameter $\eta \in (0,1)$, has proved to be appropriate.

Definition 5.2. *For each $\tau \in \mathcal{T}_{\mathcal{J}}$ let a centre $c_\tau \in \mathbb{R}^3$ and a radius $\rho_\tau > 0$ be given such that $\Gamma_\tau \subseteq D(c_\tau, \rho_\tau) = \{y \in \mathbb{R}^3 \mid \|y - c_\tau\| < \rho_\tau\}$. Then we say that a block $b = (\tau, \sigma) \in \mathcal{T}_{\mathcal{J}} \times \mathcal{T}_{\mathcal{J}}$ is admissible if*

$$\rho_\tau + \rho_\sigma \leq \eta \|c_\tau - c_\sigma\|. \tag{5.3}$$

To easily access such blocks we assume the existence of a block cluster tree $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$.

Definition 5.3. *A tree $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$ is called a* **block cluster tree** *if the node $\mathcal{J} \times \mathcal{J}$ is its root and for each $b = (\tau, \sigma) \in \mathcal{T}_{\mathcal{J}\times\mathcal{J}}$:*

- *b is either admissible*
- *or at least one of the clusters τ or σ are leaves of $\mathcal{T}_{\mathcal{J}}$.*

We are now able to give a definition of $\mathcal{H}$-matrices.

Definition 5.4. *Let $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$ be a block cluster tree and let $k_{max} \in \mathbb{N}$. We define the set of $\mathcal{H}$-matrices with maximal rank k_{max} as*

$$\mathcal{H}\left(\mathcal{T}_{\mathcal{J}\times\mathcal{J}}, k_{max}\right) := \{\mathbf{V} \;:\; \operatorname{rank}(\chi_\tau \mathbf{V} \chi_\sigma) \leq k_{max} \textit{ for all admissible leaves } b = (\tau, \sigma)\}.$$

The following result, proved in [18], gives the cost of storage and matrix-vector multiplication of an $\mathcal{H}$-matrix.

Lemma 5.5. *Let $\mathbf{V} \in \mathcal{H}(\mathcal{T}_{\mathcal{J}\times\mathcal{J}}, k_{max})$ and let p be the depth of $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$. Then*

$$N_{\text{st}} \leq 2C_{\text{sp}}(p+1)\max\{k_{\max}, C_{\text{leaf}}\}M \;\; \textit{and} \;\; N_{\mathcal{H}\cdot v} \leq 2N_{\text{st}},$$

where N_{st} is the storage requirement and $N_{\mathcal{H}\cdot v}$ the complexity of the matrix-vector multiplication.

5.1.2. Dependence of ranks on the complex frequency s.

In this section we investigate the applicability of $\mathcal{H}$-matrices for efficient data-sparse representation of matrices $\mathbf{V}(s)$. In particular we wish to understand the dependence of the rank $k_{\max}$ in Definition 5.4 and Lemma 5.5 on complex parameter s. Recall that

$$(\mathbf{V}(s))_{jk} := \int_\Gamma \int_\Gamma \varkappa_s(x,y) b_j(x) b_k(y) d\Gamma_x d\Gamma_y, \qquad j,k \in \{1,2,\ldots,M\}, \tag{5.4}$$

with the kernel function $\varkappa_s(\cdot,\cdot)$ given by

$$\varkappa_s(x,y) = \frac{e^{-s|x-y|}}{4\pi|x-y|}$$

and $\{b_1, b_2, \ldots, b_M\} \subset H^{-1/2}(\Gamma)$ a standard boundary element basis of functions with local supports.

A kernel function $\varkappa(x,y)$ is said to be *asymptotically smooth* in $X \times Y$, for $X, Y \subset \mathbb{R}^d$, if

$$\begin{aligned}&|\partial_x^\alpha \partial_y^\beta \varkappa(x,y)| \leq c_{\mathrm{as}}(\alpha+\beta)|x-y|^{-|\alpha|-|\beta|-\sigma}\\ &\text{for } x \in X, y \in Y, x \neq y, \alpha, \beta \in \mathbb{N}_0^d, \alpha+\beta \neq 0,\end{aligned} \tag{5.5}$$

where $\sigma \in \mathbb{R}$ and

$$c_{\mathrm{as}}(\nu) = C\nu!|\nu|^r$$

with some constants C and r.

In [22], see in particular [22, Theorem 4.2.13], it is shown that if a kernel function $\varkappa$ is asymptotically smooth and $\mathbf{V}$ is the Galerkin discretization (5.4), then for any accuracy $\varepsilon > 0$ there exists an $\mathcal{H}$-matrix $\mathbf{V}_\varepsilon$ such that $\|\mathbf{V}_\varepsilon - \mathbf{V}\| \leq \varepsilon$ and the rank $k_{\max} = O(\log^d \frac{1}{\varepsilon})$.

Lemma 5.6. *For all $s \in \mathbb{C}$ with $Re\, s \geq 0$ and $|Im\, s|/Re\, s \leq C_0$ for some constant $C_0 > 0$, the kernel function $\varkappa_s(x,y)$ is asymptotically smooth with constants in (5.5) depending on C_0 but not on s.*

Proof. The νth derivative of the function $f(t) = e^{-st}$ is given by

$$f^{(\nu)}(t) = (-s)^\nu e^{-st} = c_1(\nu, st)\nu! t^{-\nu}$$

with $c_1(\nu, st) = \frac{(-st)^\nu}{\nu!} e^{-st}$. Using Stirling's formula we can bound

$$|c_1(\nu, st)| \leq \frac{1}{\sqrt{2\pi\nu}} e^{-\frac{\mathrm{Re}\, s}{|s|}|st|} \left(\frac{|st|e}{\nu}\right)^\nu \leq \frac{1}{\sqrt{2\pi\nu}} \left(\frac{|s|}{\mathrm{Re}\, s}\right)^\nu \leq \frac{1}{\sqrt{2\pi\nu}} (1 + C_0^2)^{\nu/2}.$$

From this estimate and [22, Theorem E.2.1] we can conclude that $F(x,y) := f(|x-y|)$ is asymptotically smooth with implied constants independent of s. Finally, the product of two asymptotically smooth functions is again asymptotically smooth, see [22, Theorem E.3.6], and the proof is done. □

Remark 5.7. *We conclude that if $|Im\, s|/Re\, s \leq const$ then the maximal rank $k_{\max}$ of the $\mathcal{H}$-matrix approximation of $\mathbf{V}(s)$ is bounded by a constant that depends only on the accuracy of the approximation ε. Since the depth of the cluster tree is of size $p = O(\log M)$, it follows that for this range of frequencies the matrices $\mathbf{V}(s)$ can be approximated by $\mathcal{H}$-matrices with computational and storage costs of $O(M \log M)$.*

Furthermore, as shown in [8, 17], an (approximate) LU-decomposition of such matrices can be computed in $O(M \log^2 M)$ time. Though of higher complexity, this computation is much cheaper then the construction of $\mathbf{V}_\varepsilon(s)$, especially as the LU-decomposition is usually computed to a lower accuracy and then used as an effective preconditioner; see Table 5.1.

We have performed an experiment documented in Table 5.1 to illustrate this claim. $\mathcal{H}$-matrix approximations $\mathbf{V}_\varepsilon(s)$ of $\mathbf{V}(s)$ and approximate LU-decompositions of $\mathbf{V}_\varepsilon(s)$ have been computed, with various values of s satisfying $|\mathrm{Im}\, s|/\mathrm{Re}\, s = 1$.

s	Mem/dof (kB)	time (s)	Mem_{LU}/dof	time_{LU}	$\|\mathbf{I}-(\mathbf{LU})^{-1}\mathbf{V}_\varepsilon\|_2$
$1+\mathrm{i}$	8.5	132.62	6.0	12.1	1.0×10^{-4}
$2+2\mathrm{i}$	8.9	138.4	6.2	13.0	5.6×10^{-4}
$4+4\mathrm{i}$	9.3	146.1	6.6	13.8	4.4×10^{-4}
$8+8\mathrm{i}$	9.7	152.9	7.0	15.1	2.3×10^{-4}
$16+16\mathrm{i}$	8.0	124.38	7.1	18.3	9.4×10^{-5}

Table 5.1

We show memory per degree of freedom for increasing $\operatorname{Re}s = \operatorname{Im}s$, *both for the* $\mathcal{H}$*-matrix approximation* $\mathbf{V}_\varepsilon$ *and its approximate LU-decomposition. Computational times and the error in the LU-decomposition are also shown.*

The computation was performed using HLIBpro, a C++ library for computation of $\mathcal{H}$-matrices written by Ronald Kriemann; see [29, 30]. The block-wise accuracy of $\mathbf{V}_\varepsilon(s)$ approximation was set to 10^{-6}, and the accuracy of the LU-decomposition to 10^{-4}. From the results in Table 5.1 we see a slow, logarithmic, increase of costs with increasing $|s|$. The reason for this slight increase is not clear from the theory, but in any case it is clear that increasing s, with s satisfying $\operatorname{Re}s = \operatorname{Im}s$, does not increase the costs significantly.

Next, we investigate the position of frequencies arising in the algorithms of Section 4.

5.2. Frequencies in the complex plane.

We first investigate the frequencies arising in application of Lemma 4.1 to the solution of small triangular Toeplitz systems.

5.2.1. Solution of a small triangular Toeplitz system.

Consider the linear system

$$\sum_{j=0}^{n} \mathbf{A}_{n-j}\mathbf{x}_j = \mathbf{g}_j, \qquad n = 0,1,\dots,J,$$

with $J \in \mathbb{N}$ a constant independent of Δt and $\mathbf{A}_j$ the matrices from the beginning of Section 5. In order to solve this system using Lemma 4.1, we need to solve $J+1$ linear systems of the form

$$\mathbf{V}(s_\ell)\hat{\mathbf{x}}_\ell = \hat{\mathbf{b}}_\ell$$

and $s_\ell = \delta(\lambda\zeta_\ell)/\Delta t$ with $|\zeta_\ell| = 1$ and $\lambda = \text{eps}^{-\frac{1}{2J}}$. In numerical experiments we have used the choice $\lambda = 10^{-6/J}$. For the multistep methods, BDF2 and Trapezoid, this means that

$$\operatorname{Re}\delta(\lambda\zeta)/\Delta t = \delta(\lambda)/\Delta t \geq \frac{1-\lambda}{\Delta t} \approx \frac{6}{J\Delta t}.$$

For Runge-Kutta methods, for each particular method, the lower bound can be obtained from Lemma 3.1 and an investigation of the rational function $R(z)$. In general, $\frac{\log\frac{1}{\lambda}}{\Delta t}$ is a good approximation of the lower bound; note that for λ close to 1, $\log\frac{1}{\lambda} \approx 1-\lambda$.

For a constant $\lambda \in (0,1)$ it holds for all the A-stable linear multistep methods and the Runge-Kutta methods that $\|\delta(\lambda\zeta)\| \leq \mathit{const}$, for $|\zeta| = 1$. Thus, in this case, we conclude that frequencies lie in a sector of complex plane $|\operatorname{Im}s|/\operatorname{Re}s \leq$

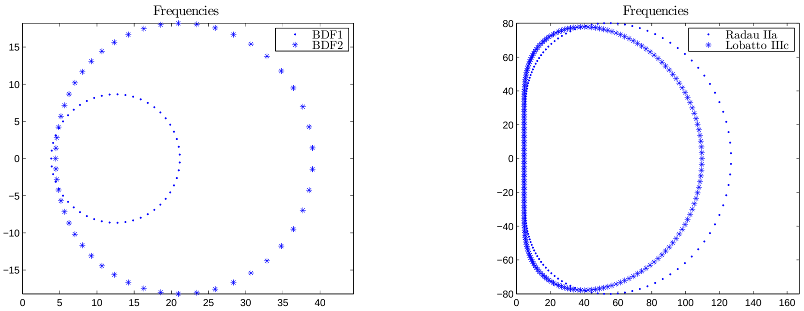

FIG. 5.1. *Frequencies occurring for the various methods with* $\Delta t = 0.08$*,* $N = 50$*,* $\lambda = 0.7$*. The 3-stage Radau IIA and the 4-stage Lobatto IIIC methods are shown.*

const with the constant independent of Δt. Hence, the Galerkin matrices and their LU-decomposition can be cheaply represented in $\mathcal{H}$-matrix format, see Lemma 5.6, Remark 5.7, and Table 5.1.

5.2.2. Computation of the update. For blocks of size $O(N) = O(1/\Delta t)$, e.g., block T_2 in Figure 4.1, it is no longer true that the frequencies satisfy $|\mathrm{Im}\, s|/\mathrm{Re}\, s \leq$ *const*. In fact with the choice $\lambda = \mathrm{eps}^{\frac{1}{2N}} = \mathrm{eps}^{\frac{T}{2\Delta t}}$ we see that this ratio is of $O(\Delta t^{-1})$, this means that the constants in (5.5) depend on Δt^{-1} and that the maximum rank of the $\mathcal{H}$-matrix approximation will increase with powers of Δt^{-1}. The fact that $\mathcal{H}$-matrices perform poorly for highly-oscillatory kernels is a well-known problem [4].

Fortunately, the highly-oscillatory operators occur only in the update of the right-hand side described in Lemma 4.2, which means that these operators need never be inverted, but only a single matrix-vector product needs to be computed. This is an ideal task for the so-called fast multipole methods [35, 36] or $\mathcal{H}^2$-matrices [4]. Here the advantage of the recursive procedure from Section 4 can best be seen, since there are no truly robust preconditioners that would allow a robust solution of systems resulting from the discretization of highly-oscillatory integral operators.

Many fast-multipole like methods for high-frequency Helmholtz integral operators have been developed since the early 1990s [1, 4, 13, 35, 36]. These have dealt only with cases of purely real and purely imaginary wavenumbers and can be adapted to our case of complex frequencies, still, to do this optimally more work is needed.

Since these modifications to the fast multipole method are still to be done, further discussion of fast multipole methods is beyond the scope of this article.

6. Numerical experiments. In this section, to shorten the presentation, we will make use of the operational notation

$$(K(\partial_t)g)(t) := \int_0^t k(t-\tau)g(\tau)d\tau, \qquad \text{for } K(s) = (\mathscr{L}k)(s) = \int_0^\infty k(t)e^{-ts}dt.$$

A motivation for this notation comes from identities such as $\partial_t^{-1} g = \int_0^t g(\tau)d\tau$ and $K_2 K_1(\partial_t)g = K_2(\partial_t)K_1(\partial_t)g$. With this notation the time-domain boundary integral operator (2.11) can be written as

$$(\mathcal{V}(\partial_t)\varphi)(x,t) = \int_0^t \int_\Gamma \frac{\delta(t-\tau-|x-y|)}{4\pi|x-y|}\varphi(y,\tau)d\Gamma_y d\tau, \quad x \in \Gamma,$$

where

$$(\mathcal{V}(s)\psi)(x) := \int_\Gamma \frac{e^{-s|x-y|}}{4\pi|x-y|}\psi(y)d\Gamma_y, \qquad x \in \Gamma.$$

6.1. A simple model problem. Before we consider large scale three dimensional problems, let us first concentrate on a simple model problem. We solve the homogeneous wave equation in the exterior of the unit ball, with zero initial conditions and separable Dirichlet boundary data given by

$$g(x,t) = h(t)Y_\ell^m,$$

where Y_ℓ^m are the spherical harmonics. We first consider using the indirect boundary integral method and represent the solution as a single layer potential:

$$u(x,t) = \mathcal{S}(\partial_t)\varphi, \qquad (\mathcal{S}(s)\psi)(x) := \int_\Gamma \frac{e^{-s|x-y|}}{4\pi|x-y|}\psi(y)d\Gamma_y, \qquad x \in \Omega^c.$$

In order that u satisfies the boundary condition we need to find φ such that

$$g(x,t) = \mathcal{V}(\partial_t)\varphi.$$

For the unit sphere $\Gamma = \mathbb{S}^2$, the Y_ℓ^m are eigenfunctions of the operators $\mathcal{V}(s)$ and $\mathcal{S}(s)$:

$$\begin{aligned} \mathcal{V}(s)Y_n^m &= \lambda_n(s)Y_n^m, \quad \lambda_n(s) = -sh_n(\mathrm{i}s)j_n(\mathrm{i}s),\\ \mathcal{S}(s)Y_n^m(x) &= \lambda_n(s,|x|)Y_n^m, \quad \lambda_n(s,r) = -sh_n(\mathrm{i}rs)j_n(\mathrm{i}s), \end{aligned}$$

where j_n and h_n are spherical Bessel functions. Thus, we see that $\varphi(x,t) = \psi(t)Y_n^m$ where ψ solves

$$\lambda_n(\partial_t)\psi = h(t).$$

We have therefore reduced the problem on the sphere to a space independent convolution equation. As a numerical example we take $g(t,x) = h(t)Y_2^m$ on the interval $[0,10]$, i.e., $T = 10$, with $h(t) = \exp(-0.4t)\sin^5(2t)$. It may be possible to obtain the analytic solution of this problem by first computing the inverse Laplace transform of $1/\lambda_2(s)$, but since it is such a simple problem, we can easily obtain a very accurate numerical solution which can serve as the exact solution in the calculation of errors.

In the left plot of Figure 6.1, we show the convergence of

$$\text{error} = \sqrt{\Delta t \sum_{j=0}^N (\psi(t_j) - \psi_j)^2}$$

against the number of time-steps N in the interval $[0,T]$ for the BDF2, Trapezoid, and 3-stage Radau IIA methods; note that in order to have a fairer comparison of costs the error of the 3-stage Radau IIA method is plotted against $3N$. The quadratic convergence of the multistep methods and the cubic convergence of the Runge-Kutta method can nicely be seen. A remark is needed here: 3-stage Radau IIA method has classical order $p = 5$ and with stage order $q = 3$. In [5] it is proved that the convergence order of the convolution quadrature is in general determined by the stage order and the degree of the polynomial by which the operator is bounded in the

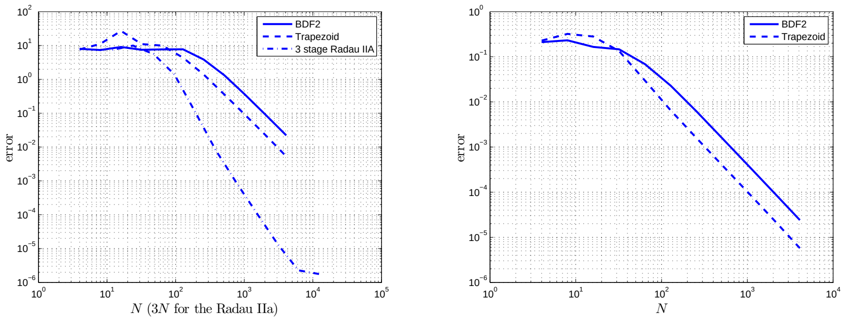

FIG. 6.1. *The convergence of the approximation error, using the indirect method, to the boundary density on the left and to the solution in the exterior on the right.*

Laplace domain. That is, reduction of order compared to the classical convergence order is expected.

Due to its higher order, we expect the Runge-Kutta method to perform better at high accuracy, but from these plots we see that the Runge-Kutta method performs significantly better also for moderate errors. The convergence for the multistep methods starts much later. To investigate this behaviour further, let us consider the error of the solution obtained by the multistep methods at a distance 1.2 from the origin, i.e., the error is now given by

$$\text{error} = \sqrt{\Delta t \sum_{j=0}^{N} (u(1.2\hat{x}, t_j) - u_j(1.2\hat{x}))^2}.$$

The results are now shown in the right-hand plot in Figure 6.1; we see that the convergence starts much earlier and a much better error is achieved.

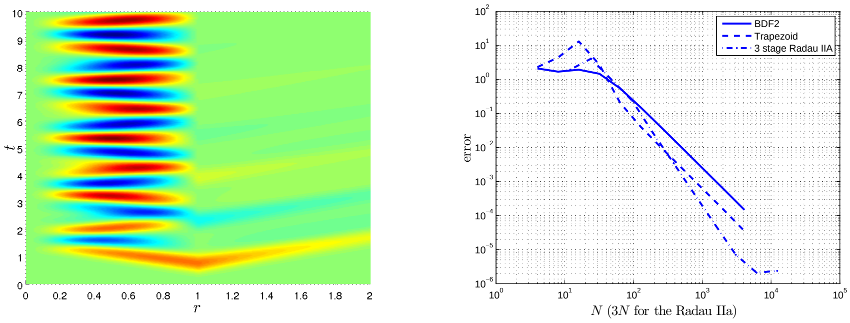

FIG. 6.2. *On the left we plot the solution in the interior ($r < 1$) and the exterior ($r > 1$). We see that the solution in the exterior is much simpler. In the right plot we show the convergence to the boundary density for the direct method which solves only the exterior problem.*

The reason for this significant difference is that in the indirect method the unknown density φ is the difference between the solution of the exterior and the solution of the interior wave equation with the Dirichlet data. Since the interior solution is much more difficult to compute due to the many reflections, see left plot of Figure 6.2, computing an approximation of the unknown density in the indirect method is much more difficult than solving the exterior problem itself. Since, as seen in the introduction, convolution quadrature is equivalent to the multistep discretization of the exterior domain problem, the errors in the computation of φ due to the interior problem after the evaluation of the convolution quadrature of $\mathcal{S}(\partial_t)\varphi^{\Delta t}$ are eliminated.

If the normal derivative of the solution on the boundary of the scatterer is required then it is reasonable to use the direct boundary integral method. In the right-hand plot of Figure 6.2 we show the error convergence. Now all three methods start to converge at around the same discretization refinement and the Runge-Kutta method becomes more effective only at higher accuracies.

This exterior model problem is, however, very simple. In the case of a non-convex domain, oscillations much as in the interior problem can occur and the convergence may be more similar to the convergence of the density for the indirect method. In the last experiment of the next section, this indeed does take place.

6.1.1. Non-smooth data. All the theoretical results for convergence of convolution quadrature require rather stringent smoothness requirements on the data, meaning that the data is overall smooth in the time interval $[0, T]$ and that the first few derivatives of the data are zero at $t = 0$. Here we repeat the previous experiment but with data that breaks the second requirement:

$$h_1(t) = \exp(-0.4t)\sin(t) \quad \text{and} \quad h_2(t) = \exp(-0.4t)\sin^2(t).$$

Note that breaking the requirement at $t = 0$ is equivalent to breaking the requirements at some later time, e.g., consider the above examples, but starting the computation at a negative time.

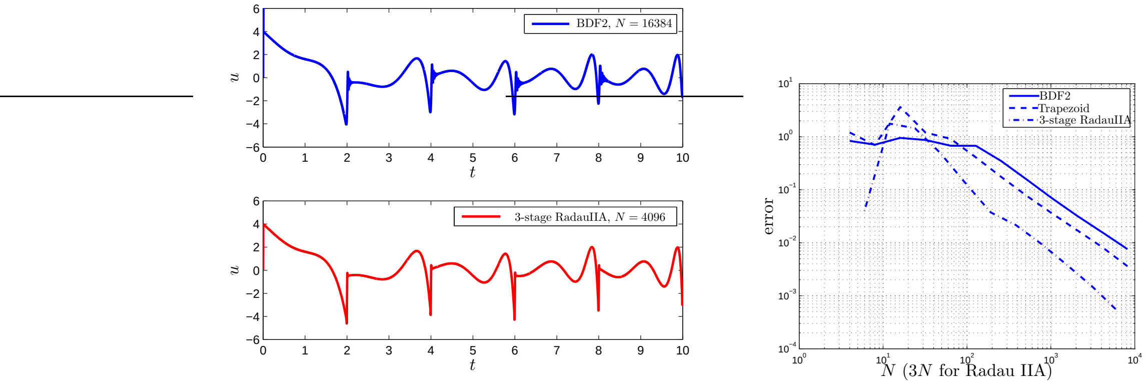

FIG. 6.3. *In the left plot we show the numerical approximation of $K(\partial_t)h_1$ obtained by BDF2 and Radau IIA methods. On the right we show the convergence for BDF2, Trapezoid, and Radau IIA methods for the computation of $K(\partial_t)h_2$.*

For data h_2, for which at most the first derivative is 0 at $t = 0$, all examined time-discretization methods converge, albeit at a slower rate. For h_1 the Trapezoid rule completely fails, whereas BDF2 and 3-stage Radau IIA seem to converge, but at a rate lower than linear; in this case also oscillations occur in the BDF2 solution near the singularities. For the numerical results see Figure 6.3.

6.2. Scattering by three dimensional bounded obstacles. We will consider two examples of Dirichlet boundary data on two different geometries.

A common example of the incident wave is a plane wave modulated by a Gaussian:

$$u^{\text{inc}}(x,t) = -\cos(\omega(t-\alpha.x))e^{-\left(\frac{t-\alpha.x-A}{\sigma}\right)^2}, \tag{6.1}$$

where $|\alpha| = 1$, $\omega, \sigma \in \mathbb{R}$. In the whole of $\mathbb{R}^n$, u^{inc} satisfies the homogeneous wave equation with wave speed $c = 1$. If we assume that the total field is zero at the boundary of the scatterer, then the scattered field $u^{\text{sc}} = u^{\text{tot}} - u^{\text{inc}}$ satisfies the boundary condition

$$u^{\text{sc}}(x,t) = g(x,t) := -u^{\text{inc}}(x,t).$$

Using the indirect method we can represent the scattered field as a single layer potential

$$u := u^{\text{sc}} = \mathcal{S}(\partial_t)\varphi, \text{ with } \varphi \text{ satisfying } \mathcal{V}(\partial_t)\varphi = g.$$

Even though we have chosen the indirect method, the unknown density is in this case a physical quantity: Since u^{inc} satisfies the wave equation in the whole domain, from jump properties of the single layer we can deduce that

$$\varphi = \partial_\nu^+ u^{\text{tot}},$$

where ∂_ν^+ is the exterior normal derivative restricted to Γ.

Another interesting right-hand side, is

$$g(x,t) = \begin{cases} 0, & \text{if } t-\alpha.x-1<0, \\ e^{-\frac{1}{t-\alpha.x-1}}\sin\omega(t-\alpha.x-1), & \text{if } t-\alpha.x-1 \geq 0. \end{cases} \tag{6.2}$$

Due to the limit amplitude principle [14], for star-shaped domains we know that the solution should exponentially quickly converge to a stationary wave; a similar experiment has been performed in [20].

In the case of the incident wave given by (6.1) we will investigate both the boundary density $\varphi = \partial_\nu^+ u^{\text{tot}}$ and the far-field pattern whereas for the second right-hand side (6.2) we will only give the far field pattern.

For $r = |x| \to \infty$ and $\hat{x} = x/r$ in the Laplace domain, with $U = \mathscr{L}u$, we have that

$$U(x,s) = \frac{1}{r}\left\{U_\infty(\hat{x},s) + \mathcal{O}\left(\frac{1}{r}\right)\right\},$$

see for example [10]. The function U_∞ can be represented as a single layer potential

$$U_\infty(x,s) = \frac{1}{4\pi}\int_\Gamma e^{-s(r-\hat{x}.y)}\Phi(y,s)d\Gamma_y,$$

where $\Phi = \mathscr{L}\varphi$. Taking the inverse Laplace transform we obtain

$$u(x,t) \approx \frac{1}{r}u_\infty(\hat{x},t) = \frac{1}{4\pi r}\int_0^t\int_\Gamma \delta(t-\tau-(r-\hat{x}.y))\varphi(y,\tau)d\Gamma_y d\tau, \text{ for } r = |x| \to \infty.$$

For r_0 such that Ω is contained in the ball centred at 0 of radius r_0, and $r_0 \ll r$, we write

$$A(\hat{x},t) := u_\infty(\hat{x}, t+(r-r_0)) = \frac{1}{4\pi}\int_0^t\int_\Gamma \delta(t-\tau-(r_0-\hat{x}.y))\varphi(y,\tau)d\Gamma_y d\tau.$$

The function A is the far field amplitude which is the quantity that will be shown in the numerical experiments.

To perform the experiments in this section, algorithms of Section 4 have been implemented. For the Galerkin discretization of the single layer potentials, a piecewise constant boundary element basis has been used. The computation of the resulting dense matrices and their storage in $\mathcal{H}$-matrix format were done using the HLIBpro library written by Ronald Kriemann; see [29, 30] and the website `www.hlibpro.org`. Further, the functions of HLIBpro for computing approximate LU-factorisations have been used as preconditioners to speed up the iterative solution of linear systems. All the computations were done on a parallel cluster with 34 Dual AMD Opteron 254 with 2800 MHz, 16 GB RAM nodes and 72 Dual AMD Opteron 250 with 2400 MHz, 4 GB RAM nodes at the Max-Planck Institute for Mathematics in the Sciences, Leipzig. All the dense matrices needed for the algorithm of Section 4, including the few preconditioners, were computed in parallel in $\mathcal{H}$-matrix format and then used to obtain the solution using the recursive algorithm.

6.2.1. Scattering by a unit sphere: Far field and the boundary density. We investigate the scattering of the incident wave (6.1) by a unit sphere with the choice of parameters

$$\omega = \frac{1}{2}\pi, \quad \alpha = (1,0,0)^T, \quad A = 4, \quad \sigma = 0.5, \quad T = 10.$$

We compute the solution on a uniform triangulation of the unit sphere by 8192 triangles. We have used the Trapezoid rule and BDF2 with $N = 120$ and the 3-stage Radau IIA method with $N = 40$. As a control computation we have used the 3-stage Radau IIA method with $N = 80$.

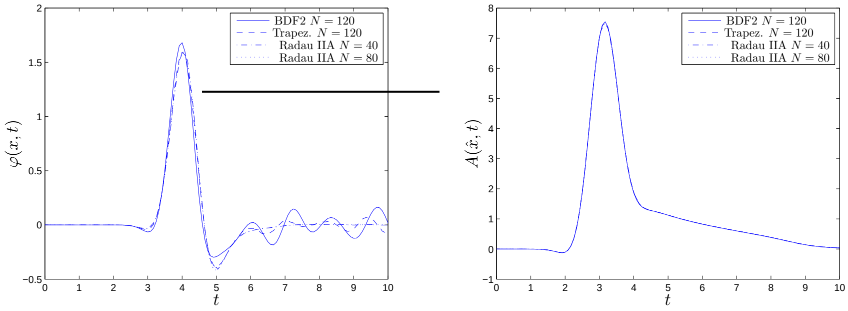

FIG. 6.4. *On the left the boundary density at the triangle centred around $x = (.2,0,.98)^T$ for the three discretization methods. On the right the far field in the direction $\hat{x} = (-1,0,0)^T$ for the same problem.*

In the left-hand side of Figure 6.4 we show the approximations of the boundary density $\varphi(x,t) = \partial_\nu^+ u^{\text{tot}}$ and in the right-hand side we show the approximations of the far field pattern $A(x,t)$ for $x = (-1,0,0)$. We can see a similar effect as in the simplified example of the previous section. All methods perform well for the exterior solution, whereas the Runge-Kutta method performs much better for the computation of the boundary density.

6.2.2. Long time stability. We repeat the previous experiment, but on the time interval $[0,80]$, i.e., $T = 80$. The number of time steps is $N = 960$ for BDF2 and

$N = 320$ for the Radau IIA method. Note that in both cases this choice implies Δt is the same as in the previous experiment.

We do not perform the experiment for the Trapezoid rule as it would require the computation of exceedingly many boundary integral operators. For the BDF2 and the Runge-Kutta method, $\omega_j \approx 0$ for $j > J$ and large enough J; see Figure 3.1 and Figure 3.2. In the computation, we approximate ω_j by 0 for $j > J$ and $J = 120$ for BDF2 and $J = 21$ for Radau IIA. This is easily implemented within the algorithm of Section 4: looking at Figure 4.1, if the block T_2 can be approximated by zero except for the operators contained in the block T_1, we simply replace T_2 by appropriately zero-padded block T_1.

The far-field in direction $(-1, 0, 0)$ is shown in Figure 6.5. We see that no spurious oscillations occur and the difference between the reduced computation on interval $[0, 10]$ and the previous, full, computation is negligible.

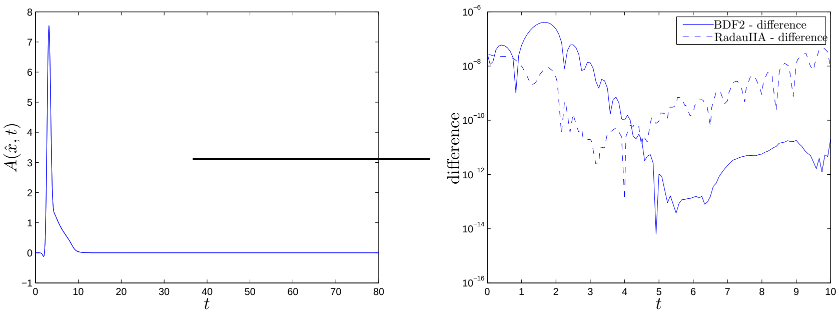

FIG. 6.5. *A long time computation is shown on the left. Both the result of the Runge-Kutta and BDF2 computations are shown, but at this scale no difference can be seen. On the right we show the error due to the approximation of some of the ω_js by zero.*

6.2.3. The limit amplitude principle. We now compute the solution for the right-hand side (6.2). We use the BDF2 and the Radau IIA method with the same Δt and T as in Section 6.2.2. The parameters in the right-hand side (6.2) are chosen as

$$\omega = \pi, \quad \alpha = (1, 0, 0).$$

In Figure 6.6 we plot the far field pattern and see that, as predicted by theory, the solution quickly converges to a standing wave. The standing wave oscillates at the frequency π which is the interior resonance frequency and which would usually pose a problem for a single layer potential representation in the frequency domain. Yet, here we do not see any adverse effect. In some time-domain boundary integral discretization methods, interior resonances can pose problems and need to be dealt with by the use of combined field integral equations where also a double layer potential would have to be computed; see [40].

6.2.4. A more complicated, non-convex domain. We also consider scattering by a more complicated non-convex scatter. The domain and the mesh were constructed by the easy-to-use meshing tool Gmsh of Geuzaine and Remacle [16]. The geometry was defined using Gmsh with the following code:

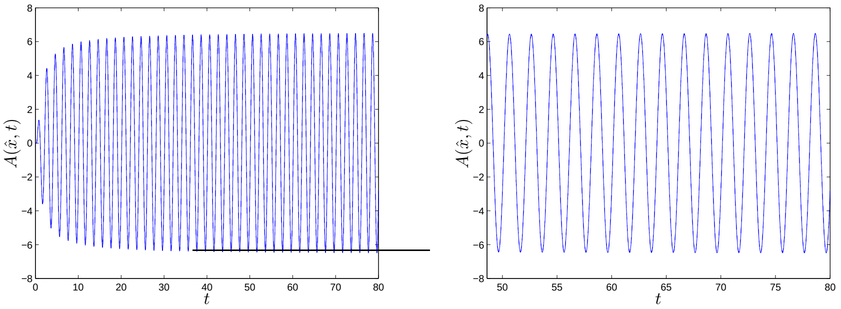

FIG. 6.6. *The far field is shown on the whole time interval* $[0, 80]$ *on the left and on the right a zoom to a later time is shown to demonstrate that the solution has become a time harmonic wave. Again, both Runge-Kutta and BDF2 computations are shown, but at this scale no difference can be seen.*

```
refine = 0.05;
Point(1) = {-0.5,-1/2,0,refine}; Point(2) = {-0.5,-3/4,0,refine};
Point(3) = {-0.5,-1,0,refine}; Point(4) = {0.0,-1,0,refine};
Point(5) = {0.0,0,0,refine}; Point(6) = {0.0,-0.5,0,refine};
Point(7) = {1.0,0.0,0,refine}; Point(8) = {0.5,0.0,0,refine};
Point(9) = {-.75,-.75,0,refine}; Circle(1) = {1,2,9};
Circle(2) = {9,2,3}; Circle(3) = {8,5,6}; Circle(4) = {4,5,7};
Line(5) = {3,4}; Line(6) = {6,1};
Extrude {{1,0,0}, {0,0,0}, Pi} {
  Line{3,6,1,2,5,4};}
Extrude {{1,0,0}, {0,0,0}, Pi/2} {
  Line{26,22,18,14,10,7};}
Extrude {{1,0,0}, {0,0,0}, Pi/2} {
  Line{36,32,40,44,48,29};}
```

The triangulation of the surface resulted in a mesh with 14440 triangles. The domain can be seen in Figure 6.7. It was constructed so that waves can be trapped inside its cavity.

The incident wave is the modulated Gaussian wave (6.1) with the parameters

$$\omega = \frac{1}{2}\pi, \quad \alpha = \sqrt{\frac{4}{5}}\left(1, \frac{1}{2}, 0\right)^T, \quad A = 4, \quad T = 10.$$

We have computed the boundary density, i.e., the normal derivative of the total field, using the 3-stage Radau IIA method with $N = 60$ and $N = 80$ and with the BDF2 method with $N = 180$ and $N = 240$. The density is shown in Figure 6.7 at various times, as computed with the Radau IIA method, $N = 80$. We can see that the wave stays for a long time inside the cavity.

We have also computed the total field at the point $x = (0,0,0)^T$ which is at the centre of the cavity. The results of the four computations are shown in Figure 6.8. We can see a similar result as in the computations of the interior solution: as soon as many reflections occur the BDF2 method requires exceedingly small time-steps to converge, whereas convergence occurs much earlier with the Radau IIA method.

7. Conclusion. We have described an efficient, robust, and parallelizable method for the numerical solution of time-domain boundary integral equations of acoustic scattering. The method allows for a multitude of different time-discretizations to be used. In this article we have investigated two multistep methods: BDF2 and Trapezoid rule, and a Runge-Kutta method: the 3-stage Radau IIA method.

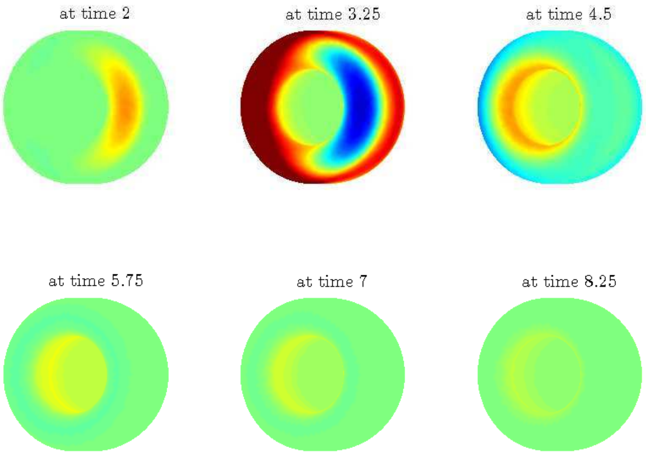

FIG. 6.7. *The normal derivative of the total field on the boundary of a non-convex scatterer. Due to the non-convexity the wave is trapped for a long time after the incident wave has passed the object. The view is chosen in the direction of the incident wave.*

In most of the experiments Radau IIA performed overwhelmingly better than the multistep methods, especially for more complicated problems. The Trapezoid rule, though somewhat more accurate than the BDF2, has a number of disadvantages if longer time simulations are needed: complicated, oscillatory kernel functions, see Figure 3.1, that prevent the use of Huygens' principle and an extreme distribution of frequencies in the complex plane. The latter is due to Trapezoid rule having the optimal stability region: exactly the left-half complex plane.

These experiments motivate the search for optimal A-stable methods for acoustic scattering applications. There is a large literature on relative merits of different time-discretization methods for wave propagation problems. However, as hyperbolic problems are not particularly stiff, mostly explicit schemes have been investigated [42], whereas we require optimal implicit schemes. Furthermore, since the triangular Toeplitz system is anyway dense, unlike in PDE based methods, the boundary integral approach does not immediately become more expensive if an k-step method is used with a larger k; high order, low stage number, multistep-multistage methods may be a class in which to seek optimal time discretization methods.

The implementation of the algorithms presented in this paper is not yet optimal, in particular diagonal multipole expansions have not been used to speed up the computation of high frequency operators. Nevertheless, we were able to perform complicated, large scale computations. Here, the crucial thing was the ability to efficiently use the capabilities of a large parallel cluster. In the future, we will report the computational times and storage requirements in detail for an optimal implementation, the bases of which has been given in this article.

Acknowledgments. The author gratefully acknowledges fruitful discussions with Christian Lubich, Peter Monk, Daniel Peterseim, Stefan Sauter, and Imbo Sim. In

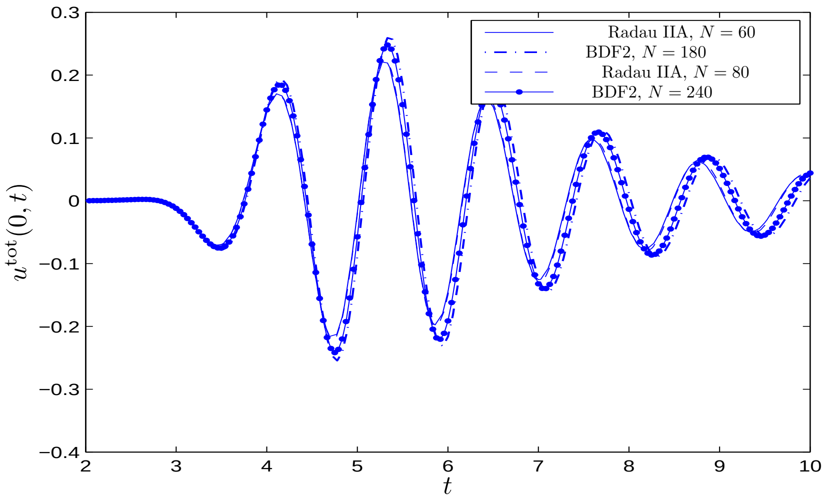

FIG. 6.8. *The total field at* $x = (0,0,0)^T$*; which is a point inside the cavity of the scatterer. Solutions using 3-stage Radau IIA* $N = 60$*,* $N = 80$ *and BDF2 with* $N = 180$ *and* $N = 240$*, are shown.*

particular I thank Ronald Kriemann for always being ready to help with the use of HLIBpro and the parallel cluster.

Appendix A. Proof of convergence of Trapezoid convolution quadrature.

Let us consider the convolution

$$u(t) = (K(\partial_t)g)(t),$$

with $K(s)$ analytic for $\operatorname{Re} s > 0$ and bounded as $|K(s)| \leq C(\sigma_0)|s|^\mu$ for $\operatorname{Re} s \geq \sigma_0$ and some constant $\mu \in \mathbb{R}$. Let the convolution quadrature approximation be given by

$$u_n := (K(\partial_t^{\Delta t})g)_n = \sum_{j=0}^{n} \omega_{n-j}^{\Delta t}(K) g_j$$

with

$$K\left(\frac{\delta(\zeta)}{\Delta t}\right) = \sum_{j=0}^{\infty} \omega_j^{\Delta t}(K)\zeta^j \tag{A.1}$$

and $\delta(\zeta)$ the generating function of the Trapezoid rule

$$\delta(\zeta) = \frac{2(1-\zeta)}{1+\zeta}.$$

In [32] it is shown that $K(\partial_t^{\Delta t})g$ converges to u at the optimal quadratic rate if $\mu < 0$. In our case, $K(s) = \mathcal{V}^{-1}(s)$, and according to (2.7), $\mu = 2$, therefore we cannot use the results of [32] directly.

We show next how to extend the results of [32] for Trapezoid discretization to the case $\mu > 0$.

Lemma A.1. *Let $K(s)$ be analytic for $Re\, s > 0$ and bounded $|K(s)| \leq C(\sigma_0)$ for $Re\, s \geq \sigma_0 > 0$. Then for and $\sigma > \sigma_0$ there exist $\bar{t} > 0$ such that for the Trapezoid based convolution quadrature*

$$\sup_{|\zeta| \leq e^{-\Delta t \sigma}} \left| K\left(\frac{\delta(\zeta)}{\Delta t}\right) \right| \leq C(\sigma_0), \quad \textit{for all } 0 < \Delta t < \bar{t}.$$

Proof. The proof follows easily from the estimate

$$\mathrm{Re}\, \frac{\delta(\zeta)}{\Delta t} = \frac{2(1-|\zeta|^2)}{\Delta t(1+2\mathrm{Re}\,\zeta + |\zeta|^2)} \geq \frac{2(1-|\zeta|^2)}{\Delta t(3+|\zeta|^2)} = \sigma + O(\Delta t) > 0,$$

for $|\zeta| = e^{-\sigma \Delta t}$. □

Theorem A.2. *Let $K(s)$ be analytic for $Re\, s > 0$ and bounded as $|K(s)| \leq C(\sigma_0)|s|^\mu$, $\mu > 0$, for $Re\, s \geq \sigma_0 > 0$. Then, if g is r times continuously differentiable with $g(0) = g'(0) = \ldots, g^{(r-1)}(0) = 0$ and $r > 2\lceil \mu \rceil + 3$, it holds for Trapezoid time discretization that*

$$\left(\Delta t \sum_{j=0}^{N} |K(\partial_t^{\Delta t}) g(t_j) - K(\partial_t) g(t_j)|^2 \right)^{1/2} = O(\Delta t^2).$$

Proof. First of all let us split the error as

$$e_1 + e_2 = \left[K_k(\partial_t^{\Delta t}) g^{(k)} - K_k(\partial_t) g^{(k)} \right] + \left[K_k(\partial_t^{\Delta t})(g^{(k)} - (\partial_t^{\Delta t})^k g) \right],$$

where $K_k(s) = K(s) \cdot s^{-k}$ and $k = \lceil \mu \rceil$. Theory from [32] can be applied to the first term in the error. Recalling (A.1), the second term can be estimated using Parseval's formula and Lemma A.1 as follows

$$\begin{aligned}\left(\Delta t \sum_{j=0}^{N} |e_2(t_j)|^2 \right)^{1/2} &\leq e^{\sigma T} \sup_{|\zeta| \leq e^{-\Delta t \sigma}} \left| K_k\left(\frac{\delta(\zeta)}{\Delta t}\right) \right| \left(\Delta t \sum_{j=0}^{N} |g^{(k)}(t_j) - (\partial_t^{\Delta t})^k g(t_j)|^2 \right)^{1/2} \\ &\leq \mathit{const} \left(\Delta t \sum_{j=0}^{N} |g^{(k)}(t_j) - (\partial_t^{\Delta t})^k g(t_j)|^2 \right)^{1/2}.\end{aligned}$$

Therefore if we can show that $g^{(k)} - (\partial_t^{\Delta t})^k g = O(\Delta t^2)$ the proof is done.

In order to prove this, we write the error as in [32, Theorem 3.1] with $s = \sigma + \mathrm{i}\omega$ and some constant $0 < c < \pi$,

$$\begin{aligned}|g^{(k)}(t) - (\partial_t^{\Delta t})^k g(t)| &\leq \frac{1}{|2\pi|} \int_{\sigma + \mathrm{i}\mathbb{R}} \left| s^k - \left(\frac{\delta(e^{-s\Delta t})}{\Delta t} \right)^k \right| |\mathscr{L} g(s)|\, |ds| \\ &\lesssim \int_{|s\Delta t| \leq c} \left| s^k - \left(\frac{\delta(e^{-s\Delta t})}{\Delta t} \right)^k \right| |\mathscr{L} g(s)|\, |ds| \qquad \text{(A.2)} \\ &+ \int_{|s| \geq c/\Delta t} \left| s^k \mathscr{L} g(s) \right| |ds| + \int_{|s| \geq c/\Delta t} \left| \left(\frac{\delta(e^{-s\Delta t})}{s\Delta t} \right)^k \right| \left| s^k \mathscr{L} g(s) \right| |ds|;\end{aligned}$$

the inequality sign $\lesssim$ denotes that a multiplicative constant has been left out. The first term in the sum (A.2) can be estimated by using the approximation property $\delta(e^{-z}) = z + O(z^{p+1})$for $|s\Delta t| < c < \pi$, as

$$\begin{aligned}\int_{|s\Delta t|\leq c} \left|1 - \left(\frac{\delta(e^{-s\Delta t})}{s\Delta t}\right)^k\right| \left|s^k \mathscr{L}g(s)\right| |ds| &\leq C\Delta t^2 \int_{|s\Delta t|\leq c} \left|s^{k+2}\mathscr{L}g(s)\right| |ds| \\ &= C\Delta t^2 \int_{|s\Delta t|\leq c} \left|\mathscr{L}g^{(k+2)}(s)\right| |ds|,\end{aligned}$$

smoothness assumptions on g insuring that the integral is bounded.

For the final integral in (A.2) we notice that $\left|\left(\frac{\delta(e^{-s\Delta t})}{s\Delta t}\right)\right| \leq C|\Delta t|^{-1} \leq \frac{C}{c}|s|$ for $|s| \geq c/\Delta t$. Hence the final integral is bounded by

$$C\Delta t^2 \int_{|s|\geq c/\Delta t} \left|s^{2k+2}\mathscr{L}g(s)\right| |ds| = O(\Delta t^2).$$

Similarly, the smoothness condition on g gives sufficient decay in the second integral in (A.2). □

REFERENCES

[1] S. Amini and A. Profit. Multi-level fast multipole solution of the scattering problem. *Engineering Analysis with Boundary Elements*, 27(5):547–564, 2003.

[2] A. Bamberger and T. Ha-Duong. Formulation variationelle espace-temps pour le calcul par potentiel retardé d'une onde acoustique. *Math. Meth. Appl. Sci.*, 8:405–435 and 598–608, 1986.

[3] L. Banjai. Multistep and multistage boundary integral methods for the wave equation. *AIP Conference Proceedings*, 1168(1):302–305, 2009.

[4] L. Banjai and W. Hackbusch. Hierarchical matrix techniques for low and high frequency Helmholtz equation. *IMA J. Numer. Anal.*, 28(1):46–79, 2008.

[5] L. Banjai and Ch. Lubich. An error analysis of Runge-Kutta convolution quadrature. *submitted*, 2010.

[6] L. Banjai and D. Peterseim. Parallel multistep methods for linear evolution problems. Technical Report 26, MPI for Mathematics in the Sciences, Leipzig, 2009.

[7] L. Banjai and S. Sauter. Rapid solution of the wave equation in unbounded domains. *SIAM Journal on Numerical Analysis*, 47(1):227–249, 2008.

[8] M. Bebendorf. Hierarchical LU decomposition-based preconditioners for BEM. *Computing*, 74(3):225–247, 2005.

[9] D. Chappell. Convolution quadrature Galerkin boundary element method for the wave equation with reduced quadrature weight computation. *To appear in IMA J. Numer. Anal.*

[10] D. Colton and R. Kress. *Inverse acoustic and electromagnetic scattering theory.* Springer-Verlag, Berlin, second edition, 1998.

[11] M. Costabel. Developments in boundary element methods for time-dependent problems. In *Problems and methods in mathematical physics (Chemnitz, 1993)*, volume 134 of *Teubner-Texte Math.*, pages 17–32. Teubner, Stuttgart, 1994.

[12] R. D'Ambrosio, G. Izzo, and Z. Jackiewicz. Highly stable general linear methods for differential systems. *AIP Conference Proceedings*, 1168(1):21–24, 2009.

[13] E. Darve and P. Havé. Efficient fast multipole method for low-frequency scattering. *J. Comput. Phys.*, 197(1):341–363, 2004.

[14] D. Eidus. The principle of limit amplitude. *Russ. Math. Surv.*, 24:97–167, 1969.

[15] A. Ergin, B. Shanker, and E. Michielssen. Fast analysis of transient acoustic wave scattering from rigid bodies using the multilevel plane wave time domain algorithm. *J. Acoust. Soc. Am.*, 117(3):1168–1178, 2000.

[16] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *Internat. J. Numer. Methods Engrg.*, 79(11):1309–1331, 2009.

[17] L. Grasedyck. Adaptive recompression of $\mathcal{H}$-matrices for BEM. *Computing*, 74(3):205–223, 2005.

[18] L. Grasedyck and W. Hackbusch. Construction and arithmetics of $\mathcal{H}$-matrices. *Computing*, 70(4):295–334, 2003.

[19] T. Ha-Duong. On retarded potential boundary integral equations and their discretization. In M. Ainsworth, P. Davies, D. Duncan, P. Martin, and B. Rynne, editors, *Computational Methods in Wave Propagation*, volume 31, pages 301–336, Heidelberg, 2003. Springer.

[20] T. Ha-Duong, B. Ludwig, and I. Terrasse. A Galerkin BEM for transient acoustic scattering by an absorbing obstacle. *Int. J. Numer. Meth. Engng*, 57:1845–1882, 2003.

[21] W. Hackbusch. *Integral equations*. Birkhäuser Verlag, Basel, 1995.

[22] W. Hackbusch. *Hierarchische Matrizen*. Springer Berlin Heidelberg, 2009. Algorithmen und Analysis.

[23] W. Hackbusch, W. Kress, and S. Sauter. Sparse convolution quadrature for time domain boundary integral formulations of the wave equation by cutoff and panel-clustering. In M. Schanz and O. Steinbach, editors, *Boundary Element Analysis: Mathematical Aspects and Applications*, volume 29, pages 113–134. Springer Lecture Notes in Applied and Computational Mechanics, 2006.

[24] W. Hackbusch, W. Kress, and S. A. Sauter. Sparse convolution quadrature for time domain boundary integral formulations of the wave equation. *IMA J. Numer. Anal.*, 29(1):158–179, 2009.

[25] E. Hairer, Ch. Lubich, and M. Schlichte. Fast numerical solution of nonlinear Volterra convolution equations. *SIAM J. Sci. Stat. Comput.*, 6(3):532–541, 1985.

[26] E. Hairer and G. Wanner. *Solving ordinary differential equations. II*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 1996. Stiff and differential-algebraic problems.

[27] G. C. Hsiao and W. L. Wendland. *Boundary integral equations*, volume 164 of *Applied Mathematical Sciences*. Springer-Verlag, Berlin, 2008.

[28] W. Kress and S. Sauter. Numerical treatment of retarded boundary integral equations by sparse panel clustering. *IMA J. Numer. Anal.*, 28(1):162–185, 2008.

[29] R. Kriemann. HLIBpro C language interface. Technical Report 10/2008, MPI for Mathematics in the Sciences, Leipzig, 2008.

[30] R. Kriemann. HLIBpro user manual. Technical Report 9/2008, MPI for Mathematics in the Sciences, Leipzig, 2008.

[31] A. R. Laliena and F.-J. Sayas. Theoretical aspects of the application of convolution quadrature to scattering of acoustic waves. *Numer. Math.*, 112(4):637–678, 2009.

[32] Ch. Lubich. On the multistep time discretization of linear initial-boundary value problems and their boundary integral equations. *Numer. Math.*, 67:365–389, 1994.

[33] Ch. Lubich and A. Ostermann. Runge-Kutta methods for parabolic equations and convolution quadrature. *Math. Comp.*, 60(201):105–131, 1993.

[34] W. McLean. *Strongly Elliptic Systems and Boundary Integral Equations*. Cambridge, Univ. Press, 2000.

[35] V. Rokhlin. Rapid solution of integral equations of scattering theory in two dimensions. *J. Comput. Phys.*, 86(2):414–439, 1990.

[36] V. Rokhlin. Diagonal forms of translation operators for the Helmholtz equation in three dimensions. *Appl. Comput. Harmon. Anal.*, 1(1):82–93, 1993.

[37] S. Sauter and C. Schwab. *Randelementmethoden*. Teubner, Leipzig, 2004.

[38] M. Schanz. *Wave Propagation in Viscoelastic and Poroelastic Continua. A Boundary Element Approach*. Lecture Notes in Applied and Computational Mechanics 2. Springer, 2001.

[39] A. H. Schatz, V. Thomée, and W. L. Wendland. *Mathematical theory of finite and boundary element methods*, volume 15 of *DMV Seminar*. Birkhäuser Verlag, Basel, 1990.

[40] B. Shanker, A. A. Ergin, K. Aygün, and E. Michielssen. Analysis of transient electromagnetic scattering from closed surfaces using a combined field integral equation. *IEEE Trans. Antennas and Propagation*, 48(7):1064–1074, 2000.

[41] X. Wang, R. A. Wildman, D. S. Weile, and P. Monk. A finite difference delay modeling approach to the discretization of the time domain integral equations of electromagnetics. *IEEE Trans. Antennas and Propagation*, 56(8, part 1):2442–2452, 2008.

[42] D. W. Zingg. Comparison of high-accuracy finite-difference methods for linear wave propagation. *SIAM J. Sci. Comput.*, 22(2):476–502 (electronic), 2000.