Comput. Methods Appl. Mech. Engrg. 245–246 (2012) 90–101

Contents lists available at SciVerse ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Runge–Kutta convolution quadrature for the Boundary Element Method

Lehel Banjai [a], Matthias Messner [b], Martin Schanz [b,*]

[a] Max-Planck Institute for Mathematics in the Sciences, Inselstr. 22–26, 04103 Leipzig, Germany
[b] Institute of Applied Mechanics, Graz University of Technology, Technikerstr. 4, 8010 Graz, Austria

ARTICLE INFO

Article history:
Received 21 November 2011
Received in revised form 21 May 2012
Accepted 9 July 2012
Available online 20 July 2012

Keywords:
Convolution Quadrature Method
Runge–Kutta method
Boundary Element Method
Time domain
Wave propagation

ABSTRACT

Time domain boundary element formulations can be established either directly in time domain or via Laplace or Fourier domain. Somewhere in between are the convolution quadrature based boundary element formulations which utilize the Laplace domain fundamental solution but establish a time stepping procedure. Up to now in applications mostly backward differential formulas of second order are used as the underlying multistep method. However, in recent mathematical literature also Runge–Kutta methods have been applied. Here, the use of Runge–Kutta methods is explained in detail and some numerical studies are given. In these studies the backward difference based procedures are compared to Runge–Kutta methods for a non-smooth problem. An ℓ^2 norm of the error is used as the basis of comparison, the convergence of which is investigated theoretically as well. The results confirm that the usage of the new techniques is preferable with regard to less numerical oscillations in the solution and better representation of wave fronts.

1. Introduction

The Boundary Element Method (BEM) in time domain is especially important for treating wave propagation problems in semi-infinite or infinite domains. In this application the main advantage of this method becomes obvious, i.e., its ability to model the radiation condition correctly. Certainly this is not the only advantage of a time domain BEM but very often the main motivation as, e.g., in earthquake engineering or scattering problems. The mathematical background of time-dependent boundary integral equations is summarized by Costabel [1].

The proposed methodologies to treat time dependent problems with the BEM can be split in two main groups: direct computation in time domain (e.g., [2,3]) or inverse transformation combined with a computation in Laplace domain (e.g., [4]). Not only due to the dependency of numerical inverse transformations on some sophisticated parameter, but also due to physical reasons it is more natural to work in the real time domain and observe the phenomenon as it evolves. But, as all time-stepping procedures, such a formulation requires an adequate choice of the time step size. An improperly chosen time step size leads to instabilities or numerical damping [5–7]. An improved and stable version of the underlying integral equation has been published by Bamberger and Ha-Duong [8] and Aimi and Diligenti [9]. Both rely on an energy principle and require two temporal integrations.

Beside these improved approaches there exists the possibility to solve the convolution integral in the boundary integral equation with the so-called Convolution Quadrature Method (CQM) proposed by Lubich[10,11]. Applications to hyperbolic and parabolic integral equations can be found in [12,13]. The CQM utilizes the Laplace domain fundamental solution and results not only in a more stable time stepping procedure but also damping effects in case of visco- or poroelasticity can be taken into account (see [14–16]). The motivation to use the CQM in these engineering applications is that only the Laplace domain fundamental solutions are required. This fact is also used for BE formulations in cracked anisotropic elastic [17] or piezoelectric materials [18]. Another aspect is the better stability behavior compared with the above mentioned formulation. For acoustics this may be found in [19] and for elastodynamics in [20]. Recently work has begun in investigating CQM for electromagnetism [21]. In the framework of fast BE formulations the CQM is used in a Panel-clustering formulation for the Helmholtz equation by Hackbusch et al. [22] and in a Multipole formulation by Saitoh et al. [23]. Recently, some newer mathematical aspects of the CQM have been published by Lubich [24]. Further, interest in high order Runge–Kutta based CQM has lately increased due to its good performance in applications, see [25] for numerical experiments in acoustics and [26–28] for convergence results. In this paper, the Runge–Kutta based CQM is described in an engineering way and emphasis is put on the numerical experiences. The mathematical background can be found in [25,29].

In the following, matrices and vectors are denoted by sans serif characters, e.g., A, and tensors by bold faced letters. The Laplace

* Corresponding author. Tel.: +43 316 8737600; fax: +43 316 8737641.
E-mail address: m.schanz@tugraz.at (M. Schanz).

0045-7825/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cma.2012.07.007

transform of a function $f(t)$ is denoted by $\hat{f}(s)$ with the complex Laplace parameter $s \in \mathbb{C}$ s.t. $\Re s > 0$.

2. Convolution quadrature with Runge–Kutta methods

The Convolution Quadrature Method (CQM) approximates a convolution integral by a finite sum

$$y(t) = f * g(t) = \int_0^t f(t-\tau) g(\tau) \mathrm{d}\tau \rightsquigarrow y(n\Delta t) \approx y_n = \sum_{k=0}^{n} \omega_{n-k}^{\Delta t}\left(\hat{f}\right) g_k \tag{1}$$

with the integration weights $\omega_{n-k}^{\Delta t}\left(\hat{f}\right)$ determined by the Laplace transform of the function $f(s)$ and the underlying multistep method. The time step size is denoted by Δt, which is assumed to be a uniform decomposition of the total time T. An index $()_n$ denotes here and in the following the function at the discrete time step $n\Delta t$. The derivation of the formula (1) will be recalled in its essential steps. Details are given at those points in the derivation which are different for Runge–Kutta methods. The other details may be found in [20]. A more abstract derivation avoiding the inverse Laplace integral can be found for the Runge–Kutta methods in [25].

However, before the CQM is given some notation and the characteristic function of Runge–Kutta methods have to be provided.

2.1. Runge–Kutta methods

A comprehensive presentation of Runge–Kutta methods may be found in the book by Hairer and Wanner [30]. Here, only the necessary aspects for the CQM will be given.

Let a Runge–Kutta method of (classical) order p and stage order q be given by its Butcher tableau $\begin{array}{c|c} \mathsf{c} & \mathsf{A} \\ \hline & \mathsf{b}^T \end{array}$ with $\mathsf{A} \in \mathbb{R}^{m\times m}, \mathsf{b}, \mathsf{c} \in \mathbb{R}^m$ and m is the number of stages. A Runge–Kutta method is said to be A-stable if the stability function

$$R(z) = 1 + z\mathsf{b}^T(\mathsf{I} - z\mathsf{A})^{-1}\mathbb{1}, \qquad \mathbb{1} := (1,1,\ldots,1)^T \tag{2}$$

is bounded as

$$|R(z)| \leqslant 1, \ \text{for } \Re z \leqslant 0 \ \text{and } \mathsf{I} - z\mathsf{A} \ \text{is non-singular for all } \Re z \leqslant 0. \tag{3}$$

The experience with the multistep based CQM in the application on BEM shows that the assumption of A-stability is necessary (see, e.g., [20]). In order to be able to make use of the convergence results proved in [27], the following assumptions will be made on the Runge–Kutta method.

Assumption 2.1

1. The Runge–Kutta method is A-stable with (classical) order $p \geqslant 1$ and stage order $q \leqslant p$.
2. The stability function satisfies $|R(\mathrm{i}y)| < 1$ for all real $y \neq 0$.
3. $R(\infty) = 0$.
4. The Runge–Kutta coefficient matrix A is invertible.

To simplify expressions assume further that $\mathsf{b}^T\mathsf{A}^{-1} = (0,0,\ldots,1)$ holds, i.e., that the method is stiffly accurate [30] or also called L-stable. This in turn implies that $c_m = 1$. Radau IIA and Lobatto IIIC are examples of Runge–Kutta methods satisfying all of the above conditions. In a Runge–Kutta method computations are done not only at the equally spaced points $t_n = n\Delta t$ but also at the stages $t_n + c_\ell \Delta t, \ell = 1,2,\ldots,m$. Note that $c_m = 1$ implies $t_n + c_m\Delta t = t_{n+1}$.

The description of a Runge–Kutta method with one formula is not straightforward due to the implicit definition of the stages. Hence, in the following the Runge–Kutta method is given for the specific differential equation

$$x'(t) = sx(t) + g(t) \quad \text{with} \quad x(t=0) = 0, \tag{4}$$

which shows up in the derivation of the CQM. An m-stage Runge–Kutta method for this equation is

$$x_{n+1} = x_n + \Delta t\mathsf{b}^T(s\mathsf{X}_n + \mathsf{g}_n), \tag{5a}$$

$$\mathsf{X}_n = x_n\mathbb{1} + \Delta t\mathsf{A}(s\mathsf{X}_n + \mathsf{g}_n). \tag{5b}$$

In (5), the right hand side $g(t_n + c_i\Delta t)$ at the stages c_i is collected in the vector g_n. Further, X_n denotes the vector of approximations at the stages and time step n.

For the derivation of the CQM it is necessary to find the characteristic function of the method. In case of multistep methods it is the quotient of the characteristic polynomials. For Runge–Kutta methods a similar expression can be given. For this, (5) is reformulated as difference of stages

$$\frac{1}{\Delta t}(\mathsf{X}_{n+1} - \mathsf{X}_n) = \mathsf{A}\left(s\mathsf{X}_{n+1} + \mathsf{g}_{n+1}\right) - \left(\mathsf{A} - \mathbb{1}\mathsf{b}^T\right)(s\mathsf{X}_n + \mathsf{g}_n) \tag{6}$$

by inserting the solution $x_{n+1} - x_n$ of (5a) into (5b). Next, a formal z-transform is performed yielding the infinite series

$$\begin{aligned} \frac{z^{-1}-1}{\Delta t}\sum_{n=0}^{\infty}\mathsf{X}_n z^n &= \left((z^{-1}-1)\mathsf{A} + \mathbb{1}\mathsf{b}^T\right)\left[s\sum_{n=0}^{\infty}\mathsf{X}_n z^n + \sum_{n=0}^{\infty}\mathsf{g}_n z^n\right] \\ \frac{\Delta(z)}{\Delta t}\sum_{n=0}^{\infty}\mathsf{X}_n z^n &= s\sum_{n=0}^{\infty}\mathsf{X}_n z^n + \sum_{n=0}^{\infty}\mathsf{g}_n z^n, \\ \left(\frac{\Delta(z)}{\Delta t} - s\mathsf{I}\right)\sum_{n=0}^{\infty}\mathsf{X}_n z^n &= \sum_{n=0}^{\infty}\mathsf{g}_n z^n, \end{aligned} \tag{7}$$

with $z \in \mathbb{C}$ and the assumption that X_n and g_n for $n \leqslant 0$, i.e., $t \leqslant 0$ is zero. The characteristic function is defined as

$$\Delta(z) = \left(\mathsf{A} + \frac{z}{1-z}\mathbb{1}\mathsf{b}^T\right)^{-1}. \tag{8}$$

Under the assumption of A- and L-stability, as mentioned above, the characteristic function can be simplified to

$$\Delta(z) = \mathsf{A}^{-1} - z\mathsf{A}^{-1}\mathbb{1}\mathsf{b}^T\mathsf{A}^{-1}. \tag{9}$$

The solution of the differential Eq. (4) at t_{n+1} is given by the solutions at the stages X_n which can be calculated with

$$x_{n+1} = \mathsf{b}^T\mathsf{A}^{-1}\mathsf{X}_n. \tag{10}$$

This expression can be found by inserting (5b) in (5a).

2.2. Runge–Kutta based convolution quadrature

The explicit formula for computing the integration weights in (1) is derived in the same manner as in the original work of [10]. The function $f(t)$ in the convolution integral is replaced by the inverse Laplace transform of its Laplace transformed function $\hat{f}(s)$, i.e.,

$$y(t) = \frac{1}{2\pi\mathrm{i}}\int_{c-\mathrm{i}\infty}^{c+\mathrm{i}\infty}\hat{f}(s)\int_0^t e^{s(t-\tau)}g(\tau)\mathrm{d}\tau\mathrm{d}s. \tag{11}$$

The integral inside the complex integral is a solution of the differential Eq. (4) and, hence, can be approximated by the Runge–Kutta solution (10) after the discretisation of the total time T in N equidistant time steps Δt. To insert the discrete solution (10) in (11) as well a formal z-transform is applied. This yields

$$\begin{aligned} \sum_{n=0}^{\infty}y_{n+1}z^n &= \frac{1}{2\pi\mathrm{i}}\int_{c-\mathrm{i}\infty}^{c+\mathrm{i}\infty}\hat{f}(s)\mathsf{b}^T\mathsf{A}^{-1}\left(\frac{\Delta(z)}{\Delta t} - s\mathsf{I}\right)^{-1}\mathrm{d}s\sum_{n=0}^{\infty}\mathsf{g}_n z^n \\ &= \mathsf{b}^T\mathsf{A}^{-1}\hat{f}\left(\frac{\Delta(z)}{\Delta t}\right)\sum_{n=0}^{\infty}\mathsf{g}_n z^n. \end{aligned} \tag{12}$$

In the last step, the residue theorem has been applied. Note, different to the CQM for a multistep method the argument of the function f is a matrix and, hence, the expression itself as well. The next step is the power series expansion

$$\hat{f}\left(\frac{\Delta(z)}{\Delta t}\right)=\sum_{n=0}^{\infty}\mathsf{W}_n^{\Delta t}\left(\hat{f}\right)z^n \tag{13}$$

with the computation of the coefficients

$$\begin{aligned}\mathsf{W}_n^{\Delta t}\left(\hat{f}\right)&=\frac{1}{2\pi \mathrm{i}}\oint_{|z|=\mathscr{R}}\hat{f}\left(\frac{\Delta(z)}{\Delta t}\right)z^{-n-1}\mathrm{d}z=\frac{\mathscr{R}^{-n}}{2\pi}\int_0^{2\pi}\hat{f}\left(\frac{\Delta(\mathscr{R}e^{\mathrm{i}\varphi})}{\Delta t}\right)e^{-\mathrm{i}n\varphi}\mathrm{d}\varphi\\&\approx\frac{\mathscr{R}^{-n}}{L}\sum_{\ell=0}^{L-1}\hat{f}\left(\frac{\Delta(\mathscr{R}\zeta^{-\ell})}{\Delta t}\right)\zeta^{n\ell}\ \text{ with }\ \zeta=e^{\mathrm{i}\frac{2\pi}{L}},\ 0<\mathscr{R}<1.\end{aligned} \tag{14}$$

The remaining steps are to insert the series (13) into (12), using Cauchy's formula for the product of two infinite series, and comparing the coefficients of the final series to obtain

$$y_{n+1}=\mathsf{b}^T\mathsf{A}^{-1}\sum_{k=0}^{n}\mathsf{W}_{n-k}^{\Delta t}\left(\hat{f}\right)\mathsf{g}_k. \tag{15}$$

This is formally the same result as for multistep methods but the essential difference is that the integration weights are now matrices with the size of the stages. Further, the results are given for time step $(n+1)\Delta t$ and not for $n\Delta t$. This results from (10) and is, obviously, the consequence from collecting the results at all stages in the next time step. Last, it should be recalled that $\mathsf{b}^T\mathsf{A}^{-1}=(0,0,\ldots,1)$ has been assumed.

3. A simple analytical study

In this section, the convolution

$$\int_0^t\delta(t-\tau-1)g(\tau)\mathrm{d}\tau=g(t-1) \tag{16}$$

is considered. Here, $\delta(t)$ is the Dirac delta distribution.

3.1. Backward Euler discretization

Firstly, the above convolution (16) is discretised with the simplest convolution quadrature based on the backward Euler method. Note that backward Euler is also often called the backward differentiation formula of order 1 (BDF 1), but is also equivalent to the 1-stage Radau IIA Runge–Kutta method. Its characteristic function is given by

$$\Delta(\zeta)=1-\zeta$$

and since the Laplace transform of $\delta(t-1)$ is e^{-s}, the convolution weights for (16) are given by the expansion

$$e^{-(1-\zeta)/\Delta t}=\sum_{j=0}^{\infty}w_j^{\Delta t}\zeta^j,\quad\text{with}\quad w_j^{\Delta t}=\frac{1}{j!}e^{-\frac{1}{\Delta t}}\Delta t^{-j}. \tag{17}$$

To further simplify matters let $g(t)=H(t)$, i.e., the Heaviside function defined by

$$H(t)=\begin{cases}1 & \text{if}\quad t\geqslant 0\\ 0 & \text{otherwise.}\end{cases} \tag{18}$$

The convolution quadrature approximation of (16) is then given by

$$y_{n+1}^{\Delta t}=\sum_{j=0}^{n}w_j^{\Delta t}H(t_{n+1}-t_j)=\sum_{j=0}^{n}w_j^{\Delta t}=e^{-\frac{1}{\Delta t}}\sum_{j=0}^{n}\frac{1}{j!}\Delta t^{-j}. \tag{19}$$

The standard theory of convolution quadrature cannot be applied to this case, the main reason being that $H(t)$ is not a smooth function of $t\in\mathbb{R}$. However, the simple setting allows to investigate the properties of the discrete solution directly. The results are summarized in the following.

Proposition 3.1. *Let $\Delta t>0$. Then, with the above definitions, it can be proved that:*

(a) With $\Delta t>0$ fixed,

$$\lim_{n\to\infty}y_n^{\Delta t}=1.$$

(b) $w_j^{\Delta t}>0$ for all $j\geqslant 0$ and, hence, with $\Delta t>0$ fixed, $y_n^{\Delta t}$ is a strictly increasing sequence with $0<y_n^{\Delta t}<1$.

(c) For any fixed $t\in[0,1)\cup(1,\infty)$,

$$\lim_{\Delta t\to 0}|y_{t/\Delta t+1}^{\Delta t}-H(t-1)|=0,$$

where it is implicitly assumed that Δt is always chosen so that $t/\Delta t\in\mathbb{N}$.

Proof. Part (a) follows directly from (19) and part (b) from (17) and (a).

To prove (c), let first $t=1+\varepsilon$ for some $\varepsilon>0$ and let $n=(1+\varepsilon)/\Delta t$. Then

$$\begin{aligned}1-y_{n+1}^{\Delta t}&=e^{-\frac{1}{\Delta t}}\left(e^{\frac{1}{\Delta t}}-\sum_{j=0}^{n}\frac{1}{j!}\frac{1}{\Delta t^j}\right)=e^{-\frac{1}{\Delta t}}\sum_{j=n+1}^{\infty}\frac{1}{j!}\frac{1}{\Delta t^j}\\&\leqslant e^{-\frac{1}{\Delta t}}\sum_{j=n+1}^{\infty}\frac{1}{\sqrt{2\pi j}}\left(\frac{e}{j\Delta t}\right)^j\leqslant\frac{1}{\sqrt{2\pi n}}\sum_{j=n+1}^{\infty}\left(\frac{1}{j\Delta t}e^{1-\frac{1}{j\Delta t}}\right)^j,\end{aligned}$$

where use is made of Stirling's approximation. It is easy to check that $f(x)=x^{-1}e^{1-x^{-1}}<1$ is a decreasing function for $x>1$ and, hence,

$$\sum_{j=n+1}^{\infty}\left(\frac{1}{j\Delta t}e^{1-\frac{1}{j\Delta t}}\right)^j\leqslant\sum_{j=n+1}^{\infty}f(n\Delta t)^j=\frac{f(n\Delta t)^{n+1}}{1-f(n\Delta t)}=\frac{f(1+\varepsilon)^{n+1}}{1-f(1+\varepsilon)}\to 0,$$

as $n\to\infty$. From this it follows that

$$|1-y_{n+1}^{\Delta t}|=1-y_{n+1}^{\Delta t}\to 0$$

as $\Delta t\to 0$. With that, the result for $t>1$ is proved.

Finally let $t=n\Delta t<1$. Then

$$\begin{aligned}y_{n+1}^{\Delta t}&=e^{-n/t}\sum_{j=0}^{n}\frac{1}{j!}\left(\frac{n}{t}\right)^j\leqslant e^{-n/t}+e^{-n/t}\sum_{j=1}^{n}\left(\frac{ne}{jt}\right)^j\\&=e^{-n/t}+\sum_{j=1}^{n}\left(\frac{n}{jt}e^{1-\frac{n}{jt}}\right)^j=e^{-n/t}+\sum_{j=1}^{n}\left[\left(\frac{n}{jt}e^{1-\frac{n}{jt}}\right)^{jt/n}\right]^{n/t}.\end{aligned}$$

Similarly as above, notice that $g(x)=(xe^{1-x})^{1/x}=e^{x^{-1}(1+\log x)-1}<1$ and $g'(x)<0$ holds for $x>1$ and, hence, $g(n/(jt))\leqslant g(1/t)$ for $j=1,\ldots,n$ and

$$y_{n+1}^{\Delta t}\leqslant e^{-n/t}+n\left(\frac{e^{1-1/t}}{t}\right)^n\to 0$$

as $n\to\infty$, i.e., as $\Delta t\to 0$. □

The principal reason for such nice properties of the discrete solution in the above example is the fact that in this case all the weights are positive. For instance, the consequence of this is that there is no over or undershoot, the discrete solution remains in the interval $[0,1]$. However, as soon as more complicated methods are considered this positivity of weights is lost. Such examples are studied in Section 4, whereas in Fig. 1 only a numerical comparison between the convolution quadrature based on backward Euler approximation and the BDF 2 based results is shown. For the latter method the overshoot is obvious and does not seem to disappear with decreasing Δt.

3.2. An $L^2(\mathbb{R})$ convergence result

In a weaker norm such as $L^2(\mathbb{R})$, it is possible to prove convergence for a larger family of Runge–Kutta convolution quadratures. To do this, let $g(t) \in L^2(\mathbb{R})$ with $g(t) \equiv 0$ for $t < 0$ and let

$$|(\mathscr{L}g)(s)| \leqslant C|s|^{-\mu}, \quad \forall \Re s > 0,$$

with $\mu > 1/2$.

The following definitions will be used

$$y(t) = \int_0^t \delta(t-\tau-1)g(\tau)\mathrm{d}\tau = g(t-1)$$

and

$$y^{\Delta t}(t) = \mathsf{b}^T\mathsf{A}^{-1}\sum_{j=0}^{\infty}\mathsf{W}_j^{\Delta t} g(t - t_j + (\mathsf{c}-1)\Delta t),$$

where $\mathsf{W}_j^{\Delta t} \in \mathbb{R}^{m\times m}$ are the weights of a Runge-Kutta method with generating function $\Delta(z)$

$$e^{-\Delta(z)/\Delta t} = \sum_{j=0}^{\infty}\mathsf{W}_j^{\Delta t} z^j$$

and $g(t + \mathsf{c}\Delta t) \in \mathbb{R}^m$ denotes the vector

$$g(t+\mathsf{c}\Delta t) = (g(t+c_1\Delta t), g(t+c_2\Delta t), \ldots, g(t+c_m\Delta t))^T \in \mathbb{R}^m,$$

in particular

$$g(t_n + \mathsf{c}\Delta t) = \mathsf{g}_n,$$

with g_n as in (5). Note that in grid points the definition matches the CQ approximation of $\int_0^t \delta(t-\tau-1)g(\tau)\mathrm{d}\tau$ as defined in the previous sections

$$y_{n+1} = \mathsf{b}^T\mathsf{A}^{-1}\sum_{k=0}^{n}\mathsf{W}_{n-k}^{\Delta t}\mathsf{g}_k = \mathsf{b}^T\mathsf{A}^{-1}\sum_{j=0}^{\infty}\mathsf{W}_j^{\Delta t}\mathsf{g}_{n-j} = y^{\Delta t}(t_{n+1}).$$

Theorem 3.2. *Let $\Delta(\zeta)$ be the generating function of an A-stable RK method of order p, that satisfies the further assumptions listed in Assumption 2.1. Then, under the above assumptions on g, it holds that*

$$\|y^{\Delta t} - y\|_{L^2([0,T])} = \mathcal{O}(\Delta t^{\beta})$$

with

$$\beta = \min\left\{\frac{(2\mu-1)p}{2(p+1)}, p\right\}.$$

Proof. Clearly $y^{\Delta t}$ and y are $L^2([0,T])$ functions and their Laplace transforms are given respectively by

$$\begin{aligned}\left(\mathscr{L}y^{\Delta t}\right)(s) &= \mathsf{b}^T\mathsf{A}^{-1}\left(\sum_{j=0}^{\infty}\mathsf{W}_j^{\Delta t}e^{-st_j}\right)e^{s\Delta t(\mathsf{c}-1)}(\mathscr{L}g)(s)\\ &= \mathsf{b}^T\mathsf{A}^{-1}e^{-\Delta(e^{-s\Delta t})/\Delta t}e^{s\Delta t(\mathsf{c}-1)}(\mathscr{L}g)(s)\end{aligned}$$

(see also (12)) and

$$(\mathscr{L}y)(s) = e^{-s}(\mathscr{L}g)(s).$$

Let $s = \sigma + \mathrm{i}\omega$ with $\sigma > 0$ constant and $\omega \in \mathbb{R}$. Then by Parseval's identity

$$\begin{aligned}\|y - y^{\Delta t}\|^2_{L^2[0,T]} &\leqslant Ce^{2\sigma T}\int_{-\infty}^{\infty}\frac{1}{|s|^{2\mu}}\left|e^{-s} - \mathsf{b}^T\mathsf{A}^{-1}e^{-\Delta(e^{-s\Delta t})/\Delta t}e^{s\Delta t(\mathsf{c}-1)}\right|^2\mathrm{d}\omega\\ &= Ce^{2\sigma T}\left(\int_{|\omega|\leqslant(\Delta t)^{-p/(p+1)}}\cdot\mathrm{d}\omega + \int_{|\omega|>(\Delta t)^{-p/(p+1)}}\cdot\mathrm{d}\omega\right)\\ &= Ce^{2\sigma T}(I_1 + I_2).\end{aligned}$$

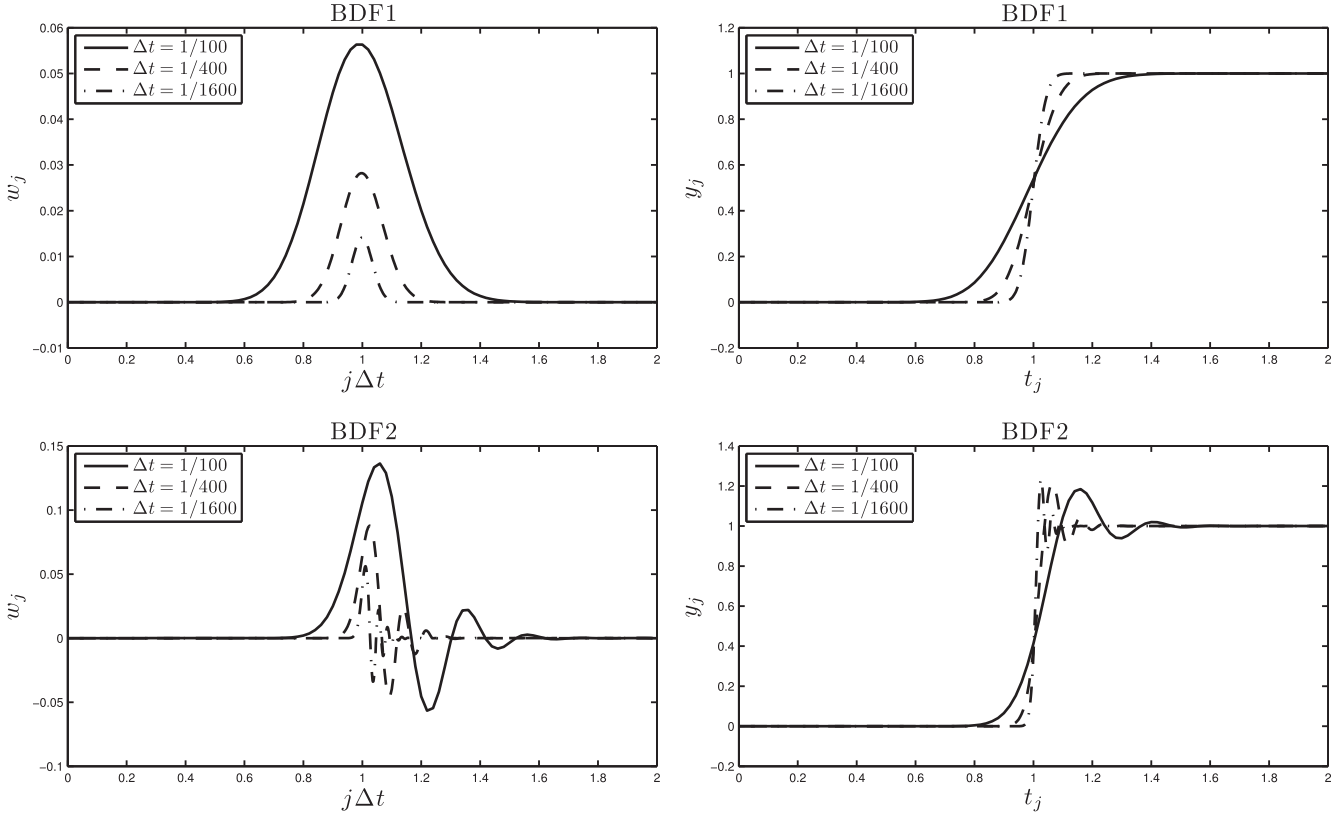

Fig. 1. In the left column, the weights of the BDF 1 (backward Euler) and the BDF 2 based convolution quadrature weights for the function $\hat{f}(s) = e^{-s}$ are shown. In the right column, the respective convolution quadrature approximations of (16) are displayed.

Due to the assumptions made on the RK method (in particular the A-stability) the bound

$$\|e^{-\Delta(e^{-s\Delta t})/\Delta t}\| \leqslant const$$

holds in any matrix norm $\|\cdot\|$. Therefore integral I_2 can be estimated as

$$I_2 \leqslant const \int_{|\omega|>(\Delta t)^{-p/(p+1)}} \frac{1}{|s|^{2\mu}} \mathrm{d}\omega = \mathcal{O}(\Delta t^{(2\mu-1)p/(p+1)}).$$

To estimate integral I_1 the approximation property

$$\mathrm{b}^T \mathrm{A}^{-1} e^{-\Delta(e^{-s\Delta t})/\Delta t} e^{s\Delta t(\mathrm{c}-1)} = e^{-s} + s^{p+1}\mathcal{O}(\Delta t^p)$$

proved in [27, Lemma 4] can be used to show that

$$\begin{aligned} I_1 &\leqslant const \Delta t^{2p} \int_{|\omega|<(\Delta t)^{-p/(p+1)}} |s|^{2p+2-2\mu} \mathrm{d}\omega \\ &= \mathcal{O}(\Delta t^{(2\mu-1)p/(p+1)}) + \mathcal{O}(\Delta t^{2p}). \end{aligned}$$

With this the proof is complete. □

Remark 3.3. The above proof can easily be modified to obtain the same result for linear multistep methods. Discrete ℓ^2 error estimates for linear multistep based CQ have been given in [13]. These, however, require more smoothness of g than the numerical examples in this paper possess.

Let $g(t) = H(t)$, then clearly g satisfies the assumptions of the theorem with $\mu = 1$, therefore the expected convergence order is $p/(2p+2)$. Numerical experiments confirming this result are illustrated in Fig. 2 with one difference: instead of the $L^2(\mathbb{R})$ error, the discrete ℓ^2 error is computed

$$e^{\Delta t} = \left(\Delta t \sum_{j=0}^{N} (y(t_n) - y_n^{\Delta t})^2 \right)^{1/2}, \quad t_N = T. \tag{20}$$

As can be seen from Fig. 2, the numerically computed rates closely match the rates predicted by the above theorem. Only for the 3-stage method there is a slight discrepancy. Here the expected rate is $5/12 \approx 0.42$ and the numerically computed is 0.45. The rate is however decreasing with Δt so that this does not contradict the theoretical result.

4. Numerical study for a model convolution integral

To keep the study as simple as possible, only two functions are convoluted, which have focus on the application in boundary element formulations for wave propagation problems. Hence, in principle a wave in time is simulated with the two functions

$$\begin{aligned} &f(t) = \delta(t-a), \quad g(t) = H(t) - H(t-b) \\ &\Rightarrow \int_0^t f(t-\tau) g(\tau) \mathrm{d}\tau = H(t-a) - H(t-a-b). \end{aligned} \tag{21}$$

In the following, the behavior of the CQM with respect to the chosen Runge–Kutta method compared to the multistep method Backward Differential Formula of order two (BDF 2) is numerically studied. The used Runge–Kutta methods are the 2-stage and 3-stage Radau IIA and Lobatto IIIC methods. The respective Butcher tableaus can be found in Table 1.

Before discussing the results a remark must be added on how to compute the integration weights in (15). The Laplace transform of function $f(t) = \delta(t-a)$ in (21) is an exponential function. Hence, the expression

$$e^{-a\frac{\Delta(\mathscr{R}\zeta^{-\ell})}{\Delta t}} = \mathrm{Y}\,\mathrm{diag}(e^{-\frac{a}{\Delta t}\lambda_1}, \ldots, e^{-\frac{a}{\Delta t}\lambda_m})\mathrm{Y}^{-1} \tag{22}$$

has to be computed. The relation (22) is true if there exists an invertible matrix Y and a diagonal matrix $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$ such that $\Delta(\mathscr{R}\zeta^{-\ell}) = \mathrm{Y}\Lambda\mathrm{Y}^{-1}$ holds. In [25], it has been shown that there is only a single value of $\mathscr{R}\zeta^{-\ell}$, respectively two such values, for which $\Delta(\mathscr{R}\zeta^{-\ell})$ is not diagonalizable in the case of the 2-stage Radau IIA method and, respectively, the 3-stage Radau IIA method. These particular values are very unlikely to be hit during a computation, still the condition number of the basis of eigenvectors Y should, as a precaution, be examined. In this unlikely case, a slight change of $\mathscr{R}$ will solve the problem.

First, the above mentioned Runge–Kutta methods are compared, where $\mathscr{R}^N = \sqrt{10^{-5}}$ is used (see, [10]). The total time is set to $T = t_N = 4.5$ s and the parameters are chosen with $a = 0.5$ and $b = 3$. The total amount of time steps has been chosen to be $N = 512$ for the BDF 2, which results in a time step size of $\Delta t = 0.0088$ s. As for an m-stage Runge–Kutta method the functions have to be evaluated not only at the time steps but also at the stages, for a fair comparison (results' quality compared to effort) in case of the 2-stage method $N/2 = 256$ time steps and for the 3-stage method $N/3 = 170$ time steps are selected. This results obviously in larger time step sizes. Still, also with this choice the numerical effort for the Runge–Kutta methods are slightly higher due to the matrix evaluations.

In Fig. 3, the results for the test functions (21) are displayed for the BDF 2, the Radau IIA, and Lobatto IIIC method. Both Runge–Kutta methods are displayed for their 2-stage version because the principal behavior is the same for the 3-stage version. It is obvious

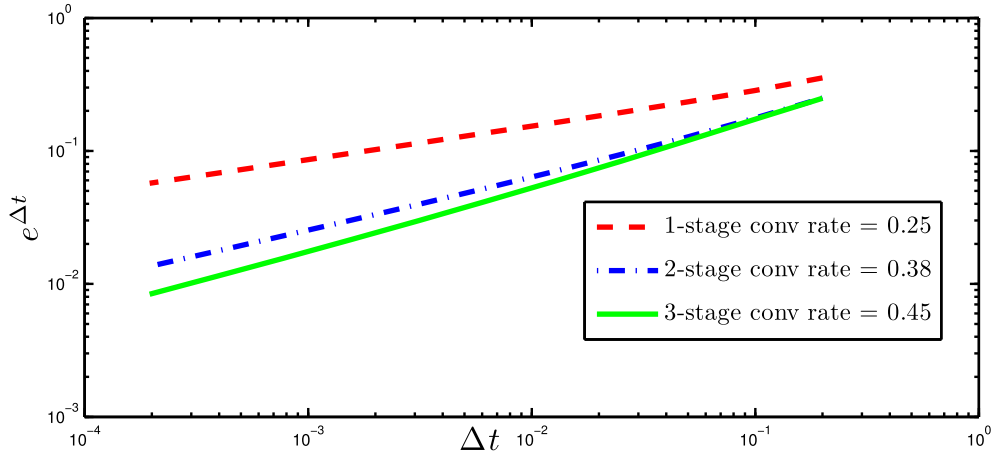

Fig. 2. The convergence of the ℓ^2 error and the numerically computed rates for m-stage Radau IIA based CQ of non-smooth data; see (16). Note that the order of the m-stage method is $p = 2m - 1$.

Table 1
Butcher tableaus of the used Runge–Kutta methods.

$$\begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array}$$

(a) 2-stage Radau IIA

$$\begin{array}{c|cc} 0 & \frac{1}{2} & -\frac{1}{2} \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

(b) 2-stage Lobatto IIIC

$$\begin{array}{c|ccc} \frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ 1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array}$$

(c) 3-stage Radau IIA

$$\begin{array}{c|ccc} 0 & \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{6} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

(d) 3-stage Lobatto IIIC

that all methods have oscillations around the jump. They differ only in the influence of these oscillations and when they appear. In Fig. 3b, a zoom at the jump is displayed for the same setting. It shows that the oscillations are the smallest in amplitude and area of influence for Radau IIA. This method has some effects before and after the jump. The BDF 2 has only some influence after the jump, however with a large amplitude. Comparable in size of these disturbances is Lobatto IIIC but in this case the disturbances are concentrated in front of the jump. The 3-stage variants of both Runge–Kutta methods have smaller disturbed areas and slightly smaller amplitudes. This is shown in Fig. 4 again with a zoom at the jump. The conclusion out of these studies is that Radau IIA seems to be the preferable method for functions with jumps.

The reason for this result can be found if the complex frequencies used for determining the integration weights are explored. For the Runge–Kutta methods these weights are defined in (14). Essentially, the eigenvalues of the matrix function $\frac{\Delta(\mathscr{R}\zeta^{-1})}{\Delta t}$ determine the used complex frequencies s_ℓ. In case of the BDF 2 this matrix function degenerates to a scalar function. These complex values are plotted in Fig. 5. Obviously, Radau IIA include the highest frequencies and, hence, is better suited to represent such a transient function as a jump. The 2-stage Lobatto IIIC has the smallest frequencies and this results in the large oscillations. The exception is the BDF 2. It also has large frequencies but compared to Radau IIA (3-stage) the relation $\frac{\Re s_\ell}{\Im s_\ell}$ is larger. It may be concluded that the 3-stage Radau IIA is the best choice. However, having in mind the application in BEM this might be not the case. In a BE formulation the fundamental solution consists of exponential functions like g in (21) and must be integrated, i.e., an oscillating function has to be integrated. Further, thinking on fast methodologies higher frequencies may cause problems. Fortunately, the real part of the complex frequency acts like a damping factor. Consequently, it is interesting to know how the complex frequencies are distributed in the area with small real part. A zoom close to $\Re s = 0$ is displayed in Fig. 5b. It is observed that the distance to a zero real part is equal for all methods. This distance is governed by two factors. If the value of $\mathscr{R}$ tends to its limit 1 the graph comes closer to the imaginary axis. This happens if either N is increased or ϵ tends to 1. The second influencing factor is the time step size Δt. Decreasing of Δt increases overall the frequencies which is somehow clear if the CQM is seen as an inverse transformation. Whereas changing $\mathscr{R}$ within some limits ($10^{-40} < \epsilon < 1$, for $\epsilon = 10^{-40}$ all calculations broke down) does not influence the final result at all, changing the time step size must have some influence. In Fig. 6, the solution of the test example is displayed for different time step sizes Δt (2-stage Radau IIA). The results confirm the expectation that smaller time steps resolve the jumps better, however, they influence the amplitude of the oscillations. It should be remarked that this effect is much less pronounced for the Runge–Kutta methods compared to BDF 2. This is directly correlated to the higher imaginary parts of the frequencies used (see (14) for the scaling of the argument of $\hat{f}$ with Δt). Finally, it should be remarked that an increase of N pushes $\mathscr{R} \to 1$ and has no influence on the results as long as realistic values are chosen.

5. Boundary element formulation with Runge–Kutta methods

Next the application of the Runge–Kutta based CQM in a collocation boundary element formulation is presented. Certainly, the same can be done for a Galerkin formulation. To keep the presentation as simple as possible the scalar wave equation is used as model problem. Obviously, the principle can be transferred to other hyperbolic problems, e.g., for elastodynamics it can be found in [29].

5.1. Governing equations

Describing with $\mathbf{x}$ and t the position in the three-dimensional Euclidean space $\mathbb{R}^3$ and the time point from the interval $(0, \infty)$ the scalar wave equation for the pressure field $p(\mathbf{x}, t)$ is

$$
\begin{aligned}
c^2\nabla^2 p(\mathbf{x},t) - \frac{\partial^2 p}{\partial t^2}(\mathbf{x},t) &= 0 \ (\mathbf{x},t) \in \Omega\times(0,\infty),\\
p(\mathbf{y},t) &= g_D(\mathbf{y},t) \ (\mathbf{y},t) \in \Gamma_D\times(0,\infty),\\
q(\mathbf{y},t) &= g_N(\mathbf{y},t) \ (\mathbf{y},t) \in \Gamma_N\times(0,\infty),\\
p(\mathbf{x},0) &= \frac{\partial p}{\partial t}(\mathbf{x},0) = 0 \ (\mathbf{x},t) \in \Omega\times(0).
\end{aligned}
\tag{23}
$$

The wave velocity is defined by

$$
c = \sqrt{\frac{K}{\varrho}}, \tag{24}
$$

with the compressibility K and the density ϱ of the inviscid fluid. The co-normal derivative defines the normal flux

$$
q(\mathbf{y},t) = (\mathscr{T}p)(\mathbf{y},t) = \lim_{\Omega\ni\mathbf{x}\to\mathbf{y}\in\Gamma}[\nabla p(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{y})], \tag{25}
$$

with the outward normal $\mathbf{n}$. The spatial domain Ω has the boundary Γ which is subdivided into two disjoint sets Γ_D and Γ_N at which boundary conditions are prescribed. The Dirichlet boundary condition is given with g_D and the Neumann boundary condition with g_N. The boundary conditions have to hold for all times. In the last statement of (23), the condition of a quiescent past is given which implies homogeneous initial conditions.

For the wave Eq. (23), a representation formula can be derived (see, e.g., [31]) and the trace to the boundary yields the boundary integral equation. Using operator notation, this boundary integral equation reads

$$
(\mathscr{V}q)(\mathbf{x},t) = \mathscr{C}(\mathbf{x})p(\mathbf{x},t) + (\mathscr{K}p)(\mathbf{x},t) \quad (\mathbf{x},t)\in\Gamma\times(0,\infty). \tag{26}
$$

The introduced operators are the single layer operator $\mathscr{V}$, the integral-free term $\mathscr{C}$, and the double layer operator $\mathscr{K}$ which are defined as

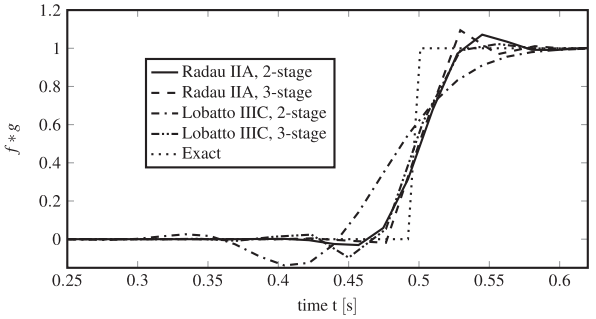

Fig. 4. Approximations for different stages of the Runge–Kutta methods.

$$
(\mathscr{V}q)(\mathbf{x},t) = \int_0^t\int_\Gamma P(\mathbf{x}-\mathbf{y},t-\tau)q(\mathbf{y},\tau)\mathrm{d}\Gamma_{\mathbf{y}}\mathrm{d}\tau, \tag{27a}
$$

$$
\mathscr{C}(\mathbf{x}) = \mathscr{I} + \lim_{\varepsilon\to0}\int_{\partial B_\varepsilon(\mathbf{x})\cap\Omega}(\mathscr{T}_{\mathbf{y}}P_{\text{static}})(\mathbf{x}-\mathbf{y})\mathrm{d}\Gamma_{\mathbf{y}}, \tag{27b}
$$

$$
(\mathscr{K}p)(\mathbf{x},t) = \lim_{\varepsilon\to0}\int_0^t\int_{\Gamma\setminus B_\varepsilon(\mathbf{x})}(\mathscr{T}_{\mathbf{y}}P)(\mathbf{x}-\mathbf{y},t-\tau)p(\mathbf{y},\tau)\mathrm{d}\Gamma_{\mathbf{y}}\mathrm{d}\tau. \tag{27c}
$$

The surface measure $\mathrm{d}\Gamma_{\mathbf{y}}$ carries its subscript in order to emphasize that the integration variable is $\mathbf{y}$. Similarly, $\mathscr{T}_{\mathbf{y}}$ indicates that the normal derivative involved in the computation of the surface flux is taken with respect to the variable $\mathbf{y}$. The function $P(\mathbf{x}-\mathbf{y},t-\tau)$ denotes the fundamental solution for the wave Eq. (23). In the expressions (27), $B_\varepsilon(\mathbf{x})$ denotes a ball of radius ε centered at $\mathbf{x}$ and $\partial B_\varepsilon(\mathbf{x})$ is its surface. In (27b), the integral free term is only determined by the static counterpart of the operator, i.e., $P_{\text{static}} = \frac{1}{4\pi r}$ with $r = |\mathbf{x}-\mathbf{y}|$. It corresponds to the solid angle at the boundary point with the value $\frac{1}{2}$ for smooth boundaries. Note that the single layer operator (27a) involves a weakly singular integral over Γ. Further, it should be remarked that the operator notation in (27a) and (27c) includes the convolution operator in time.

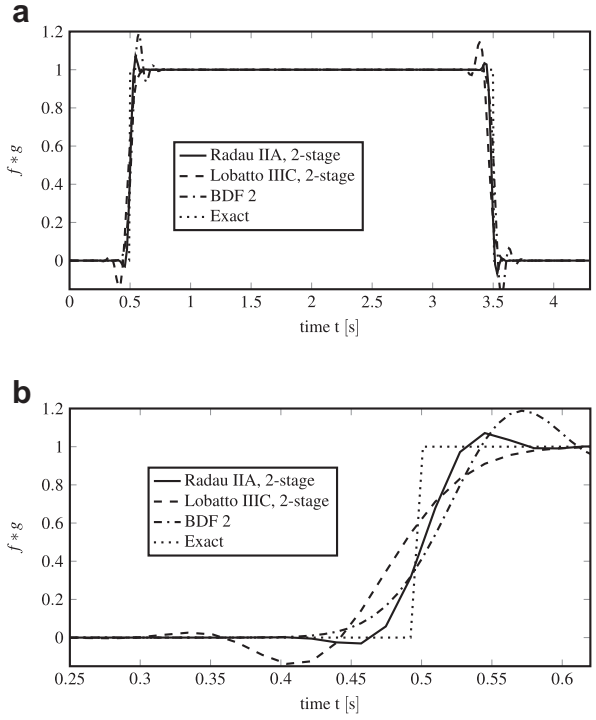

Fig. 3. Convolution $f*g$ from (21) for different Runge–Kutta methods and the multistep method BDF 2. (a) Complete time range, (b) Zoom at the jump.

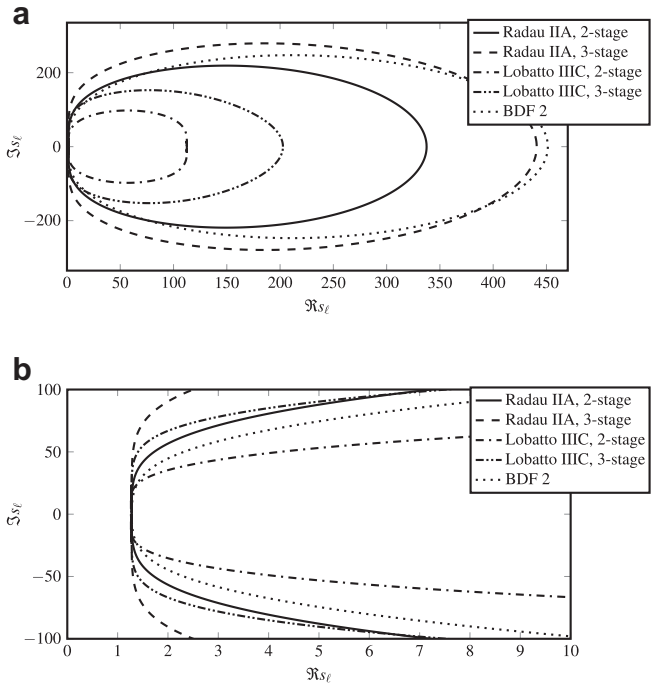

Fig. 5. Real part versus the imaginary part of the used complex frequencies s_ℓ for the data in the above study. (a) Overall picture, (b) Zoom for small real parts.

5.2. Semi-discrete equations

Let the boundary Γ of the considered domain be represented in the computation by an approximation Γ_h which is the union of geometrical elements

$$\Gamma_h = \bigcup_{e=1}^{E} \tau_e. \tag{28}$$

τ_e denote boundary elements, e.g., surface triangles as in this work, and their total number is E. Now, the boundary data p and q are approximated with continuous shape functions φ_i or discontinuous shape functions ψ_j, which are defined with respect to the geometry partitioning (28), and time dependent coefficients $p_i(t)$ and $q_j(t)$. This yields

$$p(\mathbf{y},t) = \sum_{i=1}^{M_e} p_i(t)\varphi_i(\mathbf{y}) \quad \text{and} \quad q(\mathbf{y},t) = \sum_{j=1}^{N_e} q_j(t)\psi_j(\mathbf{y}). \tag{29}$$

Inserting these spatial shape functions in the boundary integral equations (26) and applying a collocation method results in the semi-discrete equation system

$$\mathsf{V} * \mathsf{q} = \mathsf{C}\mathsf{p} + \mathsf{K} * \mathsf{p}. \tag{30}$$

In the Eq. (30), the time is still continuous and the convolution has to be performed. Further, the notation of matrices/vectors with sans serif letters denotes that in these matrices the data at all nodes and all degrees of freedom are collected.

5.3. Application of CQM

Next, the temporal discretization by the CQM has to be introduced. The CQM is used for the time discretisation of the semi-discrete equation system (30), i.e., for Runge–Kutta methods (15) is used or for multistep methods the respective counterpart (1). This results in the time stepping procedure

$$\mathsf{b}^T\mathsf{A}^{-1}\sum_{k=0}^{n}\mathsf{W}_{n-k}^{\Delta t}\left(\hat{\mathsf{V}}\right)\mathsf{q}(k\Delta t) = \mathsf{C}\tilde{\mathsf{p}}((n+1)\Delta t) + \mathsf{b}^T\mathsf{A}^{-1}\sum_{k=0}^{n}\mathsf{W}_{n-k}^{\Delta t}\left(\hat{\mathsf{K}}\right)\mathsf{p}(k\Delta t), \tag{31}$$

where the vectors q (size mN_e) and p (size mM_e) now contain all the data at each node and at each stage of the Runge–Kutta method. Eq. (31) is formulated at the final stage of each time step and, consequently, in the vector $\tilde{\mathsf{p}}$ (size M_e) only the results at each node are collected.

Using (10), the first term on the right hand side in (31) can be written as

$$\mathsf{C}\tilde{\mathsf{p}}((n+1)\Delta t) = \mathsf{C}\mathsf{b}^T\mathsf{A}^{-1}\mathsf{p}(n\Delta t) = \mathsf{b}^T\mathsf{A}^{-1}\tilde{\mathsf{C}}\mathsf{p}(n\Delta t), \tag{32}$$

i.e., by a proper arrangement of C into $\tilde{\mathsf{C}}$ this term can as well be written at the stages. Taking this representation into account and separating the actual time step from the time history, the representation of the time stepping algorithm at the stages of the Runge–Kutta method is given with

$$\mathsf{W}_0^{\Delta t}\left(\hat{\mathsf{V}}\right)\mathsf{q}(n\Delta t) = \tilde{\mathsf{C}}\mathsf{p}(n\Delta t) + \mathsf{W}_0^{\Delta t}\left(\hat{\mathsf{K}}\right)\mathsf{p}(n\Delta t) + \sum_{k=0}^{n-1}\left[\mathsf{W}_{n-k}^{\Delta t}\left(\hat{\mathsf{K}}\right)\mathsf{p}(k\Delta t) - \mathsf{W}_{n-k}^{\Delta t}\left(\hat{\mathsf{V}}\right)\mathsf{q}(k\Delta t)\right]. \tag{33}$$

Second, in the actual time step the boundary data are sorted in unknown and given boundary data, where the latter are approximated by the shape functions. The collocation is performed on the Dirichlet boundary Γ_D at the center of the element (for constant shape functions) and on the Neumann boundary Γ_N at the nodes. This yields the quadratic block system

$$\begin{bmatrix} \mathsf{W}_0^{\Delta t}\left(\hat{\mathsf{V}}_{DD}\right) & -\mathsf{W}_0^{\Delta t}\left(\hat{\mathsf{K}}_{DN}\right) \\ \mathsf{W}_0^{\Delta t}\left(\hat{\mathsf{V}}_{ND}\right) & -\left(\tilde{\mathsf{C}} + \mathsf{W}_0^{\Delta t}\left(\hat{\mathsf{K}}_{NN}\right)\right) \end{bmatrix} \begin{bmatrix} \mathsf{q}_D \\ \mathsf{p}_N \end{bmatrix}(n\Delta t) = \begin{bmatrix} \mathsf{f}_D \\ \mathsf{f}_N \end{bmatrix}(n\Delta t), \tag{34}$$

where the vectors f_D and f_N contain the product of the given boundary data with the weights of the first time step and the complete history. Certainly, instead of computing the above formula the reformulated version of the CQM as presented in [32] can be used. This is discussed in more detail and from a mathematical point of view in [29].

A remark must be made on computing the matrix valued integration weights $\mathsf{W}_n^{\Delta t}\left(\hat{\mathsf{V}}\right)$. For acoustics the fundamental solution in Laplace domain is $\hat{P}(r,s) = \frac{1}{4\pi r}e^{-\frac{rs}{c}}$ with the distance $r = |\mathbf{x}-\mathbf{y}|$. Hence, the integration weight for the collocation point $\mathbf{x}_i$ is

$$\mathsf{W}_n^{\Delta t}\left(\hat{\mathsf{V}}[i,j]\right) = \frac{\mathscr{R}^{-n}}{L}\sum_{\ell=0}^{L-1}\zeta^{n\ell}\int_{\mathrm{supp}(\psi_j)}\frac{1}{4\pi r}e^{-\frac{\Delta t}{c}\left(\mathscr{R}\zeta^{-\ell}\right)}\psi_j(\mathbf{y})\,\mathrm{d}\Gamma_{\mathbf{y}}. \tag{35}$$

To compute the exponential function of a matrix the same decomposition as discussed in (22) is used.

The remaining part is the numerical realization of the above given procedure. All regular integrals are performed with Gaussian quadrature formulas. The weakly singular integrals are solved with the formulas by Erichsen and Sauter [33]. Finally, the block equation system (34) is solved by inserting the first equation into the second to obtain the Schur complement. Solving this system gives the pressure data and subsequently the data for the flux are computed. This procedure can be performed by a nested iterative solution with GMRES or with direct solvers (see, e.g. [37]).

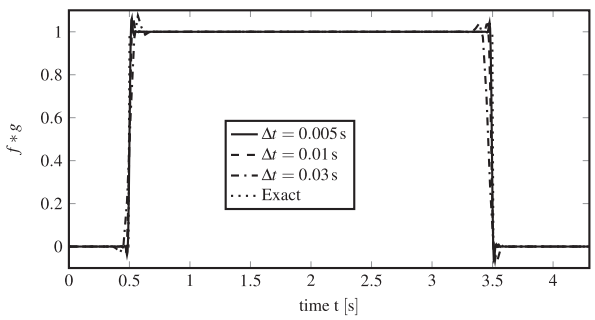

Fig. 6. Convolution $f * g$ from (21) for different time step sizes (2-stage Radau IIA).

6. Numerical studies

The above sketched solution procedure is tested with different Runge–Kutta and multistep methods using a 3-d benchmark example with known analytical 1-d solution for comparison. All computations were performed by using the HyENA C++ library for the numerical solution of partial differential equations using the Boundary Element Method [34]. For the Fourier like transformations the FFTW routines [35] are taken. To speed up the calculation the fast methodology based on the Adaptive Cross Approximation (ACA) as presented in [36] for a symmetric Galerkin formulation is used. In contrast to this publication, here as discussed above, a collocation approach is used.

A 3-d column of size $\ell_1 = 3.0$ m and $\ell_2 = \ell_3 = 1.0$ m, as depicted in Fig. 7, is considered. It has zero pressure on one end and on the other end the normal flux $q = -1H(t)$ N/m^2 is prescribed. The material parameters of air ($c = 346$ m/s) are taken. The column shown in Fig. 7 is discretised with 12032 triangular boundary elements of mesh size $h = 0.05$ m on 5529 nodes. The pressure and flux are approximated by piecewise constant and continuous linear polynomials, respectively. In order to compare different time discretizations the dimensionless value

$$\beta = \frac{c\Delta t}{mh} \tag{36}$$

is introduced. The parameter m denotes as in the previous sections the number of stages, i.e., for the Runge–Kutta methods β represents a time step size related to the stages and not to the time steps. This is introduced to have a fair comparison with the multistep method with regard to the numerical effort (see the discussion of this aspect in Section 4).

6.1. Behavior of Runge–Kutta CQM

The first comparison is performed for the BDF 2, 2-stage Radau IIA, and 3-stage Radau IIA. The flux results at that end where the pressure is set to zero are displayed in Fig. 8 versus time. The chosen time step size is a nearly optimal choice for all three methods. Obviously, all methods produce good results, however, the BDF 2 has large overshoots at the jumps, i.e., at the wave fronts. These oscillations are also visible for the Runge–Kutta methods but with much smaller values and in a narrower region. This is in accordance with the experiences of Section 4. Further, it can be stated that the 3-stage Radau IIA method produces the best results. This corresponds obviously to the higher frequencies which are used during the calculation (see Fig. 5).

It can be as well observed that the solution of the 3-stage method is not a straight line but it slightly oscillates. This indicates that the time step size is for this method close to the instability limit, i.e., this method does not allow very small time steps. A study concerning the time step sizes is presented in Fig. 9 for both Radau IIA methods. Overall, it can be observed that the time step size has not too much influence. Certainly, a smaller time step resolves the jumps better than a larger one. Contrary to the experiences with the BDF 2, the overshoots at the jumps show no clear dependence on the time step size. However, as already noted above, the 3-stage Radau IIA can not go much beyond $\beta = 0.3$, whereas the 2-stage Radau IIA can still compute with a $\beta = 0.2$. At this point it must be recalled that the used β (36) is related to the stage size and not to the time step size which is by a factor of m larger.

To have a deeper insight in the influence of the time step size a closer look on the last third of the plots in Fig. 9 is presented in Fig. 10. There, additionally, the results for a BDF 2 solution are plotted to compare. The zooms confirm the observations from above. The higher the order of the Runge–Kutta method, the better are the results, i.e., the area of the oscillations becomes narrower. On the other hand the method becomes more sensitive on the lower limit for the time step size. An overall observation is that the Runge–Kutta based solutions are not such affected by a too coarse time step size as the BDF 2 based solutions. In principle the solutions suggest that for Runge–Kutta based CQM the sensitivity on the time step size is smaller than for multistep based CQM. Certainly, the time step must be small enough to resolve the physical effect in time.

Summarizing, the quality of the BEM results is improved by the Runge–Kutta method. However, it is clear that one Runge–Kutta time step is more expensive than a BDF 2 step. Hence, the question arise whether the numerical costs are as well better or not.

6.2. Computational cost

The comparison of numerical costs is a difficult task because it is not obvious what has to be measured. Beside, in the authors opinion a BEM formulation without fast methods is not suitable for real world problems. Consequently, a study on numerical costs must take a fast method into account, though an additional approximation is introduced. In the following, first the influence of this approximation is shown and the efficiency of the fast algorithm is studied, i.e., the performance of the ACA is presented. Further, the convergence behavior compared to the specific analysis in Section 3 is presented. Last, the numerical costs are compared.

In the proposed BEM formulation ACA is used to speed up the calculation. This algebraic technique allows to compute only the necessary matrix entries to achieve a pre-selected accuracy ϵ_{ACA}. As shown in [36], an $\epsilon_{ACA} = 10^{-3}$ results in a deviation of the results for larger times. The same holds if Runge–Kutta methods are used as displayed in Fig. 11. In the same paper, the results suggest that

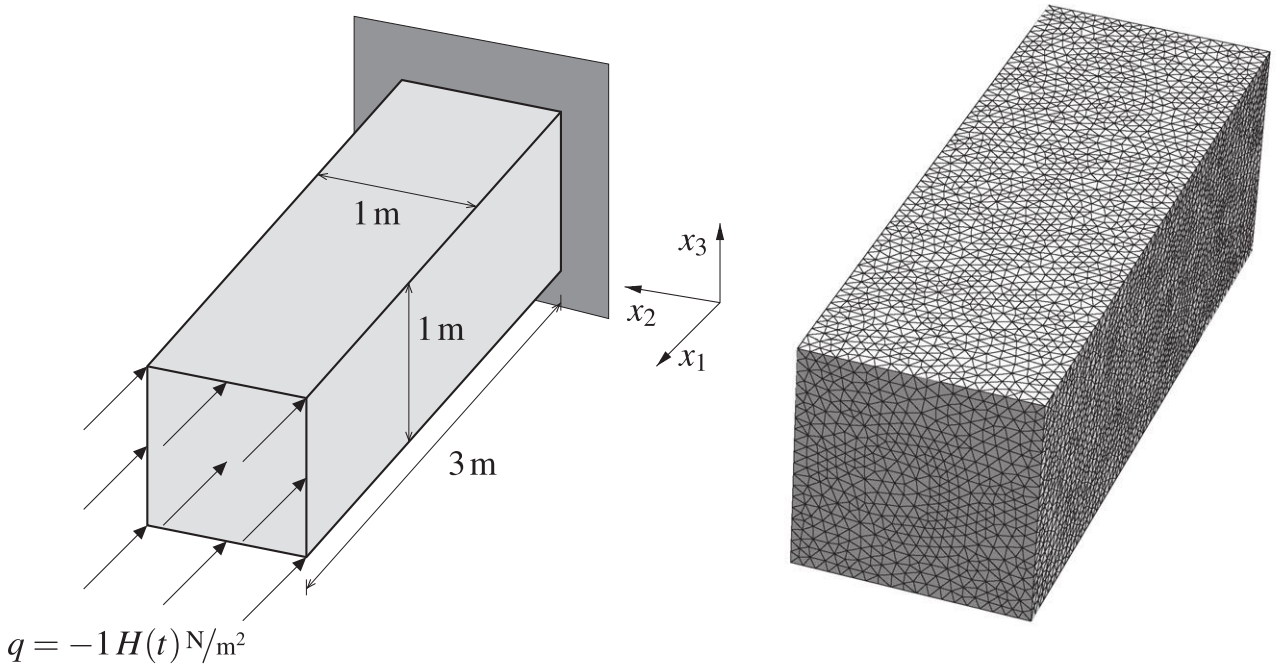

Fig. 7. System, boundary conditions, and mesh.

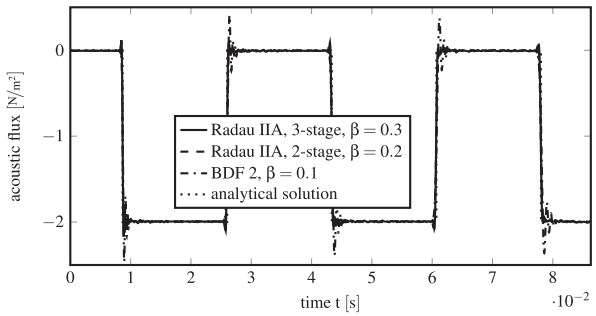

Fig. 8. Flux at the free end versus time for the 2- and 3-stage Radau IIA method compared to the BDF 2 solution.

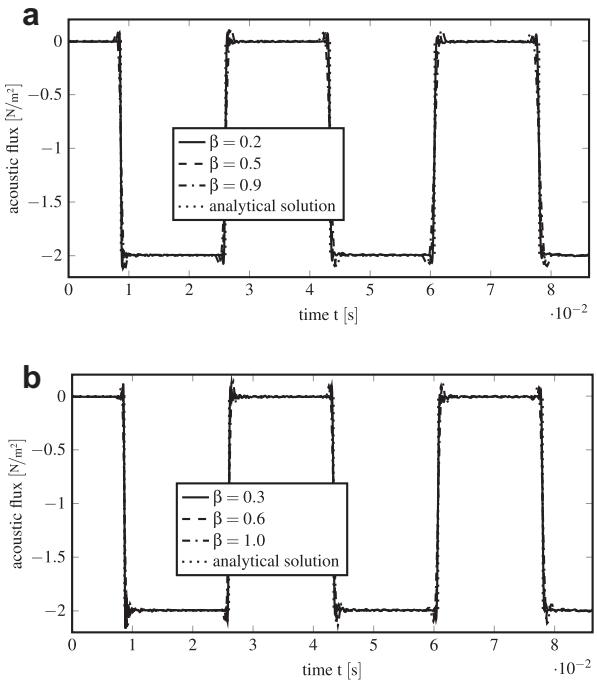

Fig. 9. Flux at the free end versus time: influence of the time step size. (a) 2-stage Radau IIA, (b) 3-stage Radau IIA.

$\epsilon_{ACA} = 10^{-5}$ is sufficient for good long-time behavior. This holds as well for the Runge–Kutta based formulation. In Fig. 12, the long-time behavior is studied for the 3-stage Radau IIA using $\beta = 1.1$. Even for this long observation time the result follows closely the analytical solution. Based on this study all other results have been computed with $\epsilon_{ACA} = 10^{-5}$.

An essential criterion for the numerical costs is the compression rate, i.e., the size relation of the used $\mathscr{H}$-matrix to the dense matrix without using ACA. In the CQM based BEM, ACA is applied in each frequency step. Hence, the compression rate is different at each complex frequency s_ℓ. For the largest matrix block $\hat{\mathsf{K}}_{\mathsf{NN}}$ (the double layer operator for the Neumann boundary) the compression rate is plotted in Fig. 13 for the Radau IIA methods compared to the BDF 2. On the horizontal axis half of the used complex frequencies s_ℓ have been plotted starting from small real parts to larger real parts. These are the only ones to be calculated because the other half are the complex conjugate (see Fig. 5). Obviously, the compression rate is large in the beginning and then decreases to a nearly constant value. In view of Fig. 5 such a behavior has to be expected. The bad compression rates correspond to small real parts but large imaginary parts. As seen on the frequency distribution in Fig. 5b

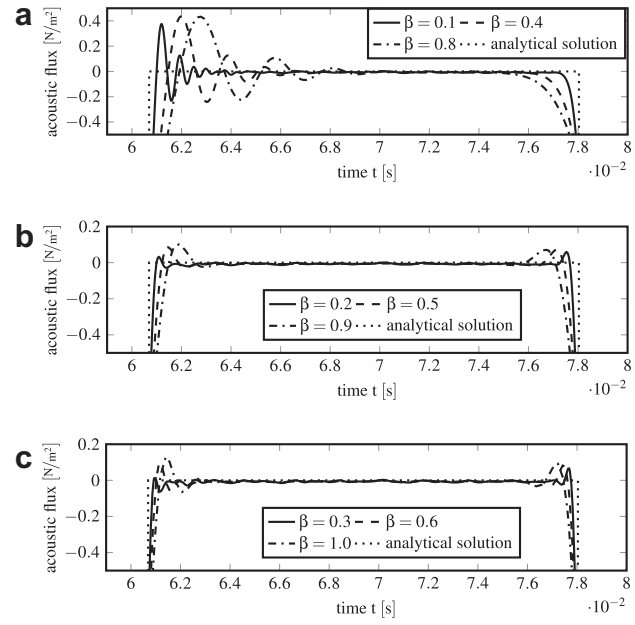

Fig. 10. Flux at the free end versus time: zoom on the last third of Fig. 9. (a) BDF 2, (b) 2-stage Radau IIA, (c) 3-stage Radau IIA.

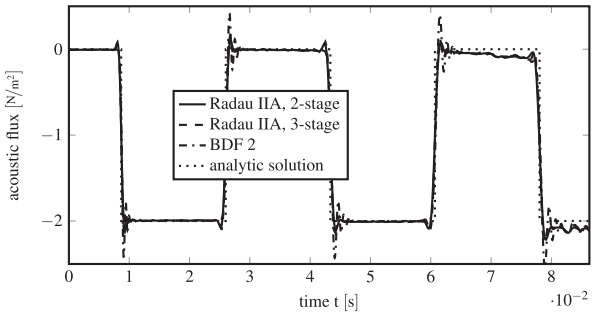

Fig. 11. Flux at the free end versus time: reduced accuracy ($\epsilon_{ACA} = 10^{-3}$) of the approximation (ACA) and solver.

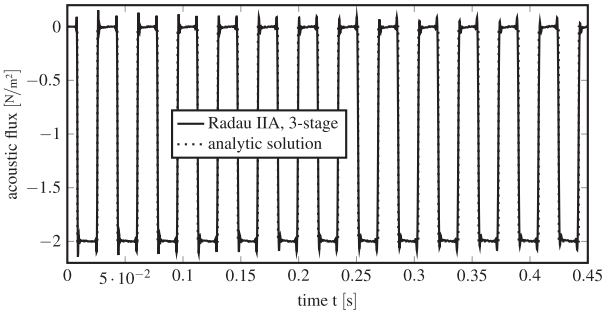

Fig. 12. Flux at the free end versus time: long time behavior of the 3-stage Runge–Kutta method.

the Radau IIA methods has a larger ratio $\Im s_\ell / \Re s_\ell$ and, hence, a worse compression compared to the BDF 2.

Certainly, if a larger time step size would have been used for Fig. 13 the compression rates would have been better. This effect is studied in Fig. 14. In this figure, the overall compression is plotted versus the time step size for the Radau IIA methods and the BDF 2. Overall compression means the summed compression rates over all frequencies. It is in principle a measure of the computing

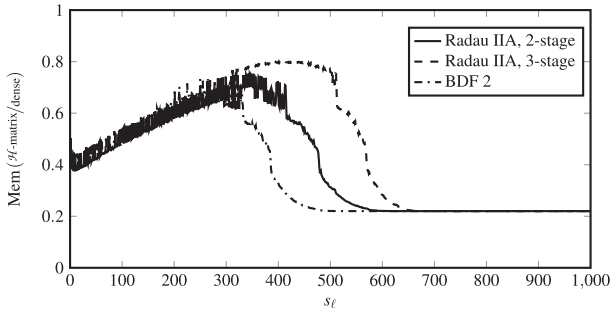

Fig. 13. Compression of K_N versus complex frequencies s_ℓ for $\beta = 0.3$.

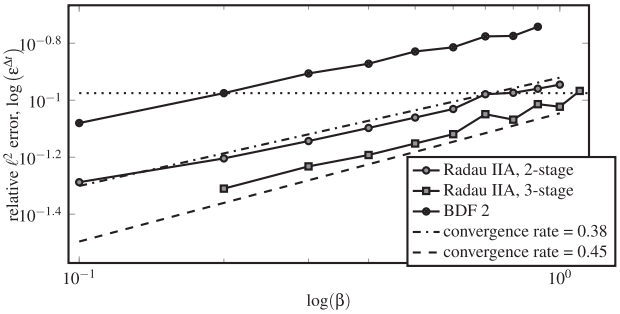

Fig. 15. Log–log-plot of the point wise relative ℓ^2 error.

time because the used memory corresponds in ACA directly to the computed matrix entries. Hence, the storage is proportional to the computing time. The strongest effect can be observed for the 3-stage Radau IIA method, whereas nearly no effect can be seen for the BDF 2. As discussed above the used frequencies cause this behavior. In combination with the experience on the sensitivity of the different methods on β the Radau IIA method seems to be promising. In principle the higher order of this method becomes visible here.

Collecting all studied effects the question arise which method is the best? The above results show different effects and overall the Radau IIA (3-stage) seems to be the method of choice. However, it is clear that this method is the most expensive one if the time step size and precision is fixed in a comparison. But, it is clear as well that a fair comparison should check the relation numerical effort to quality of the results. Unfortunately, a measure for the quality of the results is difficult. In Section 3, it has been proven that for the test convolution (16) the L^2 error converges. As the acoustic problem under consideration has a similar form, i.e., the fundamental solution in time domain is also a Dirac distribution with a retarded argument, the L^2 error may converge as well. The discrete ℓ^2 error (20) is plotted in Fig. 15 versus β in a logarithmic scale. Additionally, as dashed lines the theoretical convergence rate for the Runge–Kutta methods of Section 3 are presented. Two observations can be made: First, the error converges and, second, the theoretical values are nearly obtained. Further, the BDF 2 seems to have the worst convergence rate and the Radau IIA (3-stage) the best. This fits to the presented results from above.

To answer the question which method is the most cost efficient, all three methods are compared in their numerical costs for the same quality. The quality can now be measured with the above plotted ℓ^2 error. The horizontal dotted line in Fig. 15 indicates an error $\epsilon^{\Delta t} \approx 10^{-1}$. In Table 2, the costs for the different methods are compared for computations with this error. Additionally, the β-values which are required to achieve the error and the necessary number of time steps N_β are given. As discussed above, the number of frequencies to be computed is half of the time step number. That is why in the fourth row $N_\beta/2$ is displayed. The last row compares the numerical costs of the different methods, where the BDF 2 calculation is used as basis. In this measure the matrix entries necessary for the ACA are counted and summed up over all $N_\beta/2$ time (frequency) steps. This determines in principle the necessary storage but as well the speed because the CPU time is proportional to the amount of necessary matrix entries to be computed. The preferable feature of these numbers is their independence of the used CPU. Evidently, the Radau IIA in its 3-stage version is the most efficient technique.

Table 2
Cost for a point wise relative ℓ^2 error $\epsilon^{\Delta t} \approx 10^{-1}$ (see the dotted line in Fig. 15 for the required β)

Time stepping scheme	Required β	$\epsilon^{\Delta t}$	$N_\beta/2$	Speed up
BDF 2	0.2	$1.06 \cdot 10^{-1}$	1500	1
Radau IIA, 2-stage	0.7	$1.05 \cdot 10^{-1}$	432	3.19
Radau IIA, 3-stage	1.1	$1.08 \cdot 10^{-1}$	303	4.30

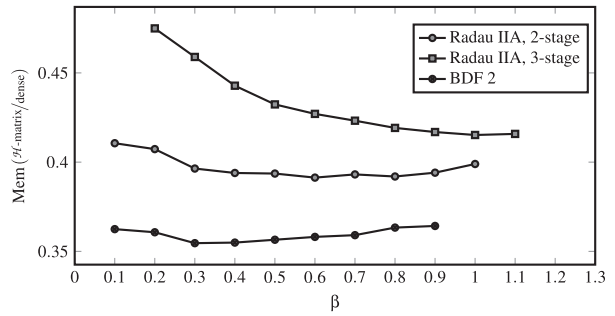

Fig. 14. Overall compression of K_N versus time step size.

7. Conclusions

The application of Runge–Kutta methods in the Convolution Quadrature Method has been discussed. The principal behavior is studied with two specific functions representing a wave front. The proposed methodology is applied on a collocation BEM.

The results of both, the test example and the BEM, show that the Runge–Kutta methods improve the behavior at the jumps at wave fronts. The oscillations around these fronts become smaller and the area of influence decreases as well. There is still a lower stability limit which increases slightly for higher order Runge–Kutta methods. The sensitivity on the time step size is slightly improved compared to the BDF 2. Finally, it can be concluded that the usage of Runge–Kutta methods pays off but the BDF 2 results are still good. Nevertheless, there are possible examples where the use of Runge–Kutta methods is necessary.

References

[1] M. Costabel, Time-dependent problems with the boundary integral equation method, in: E. Stein, R. de Borst, T.J.R. Hughes (Eds.), Encyclopedia of Computational Mechanics, Fundamentals, vol. 1, John Wiley & Sons, New York, Chichester, Weinheim, 2005.
[2] W.J. Mansur, A time-stepping technique to solve wave propagation problems using the boundary element method, Phd thesis, University of Southampton, 1983.
[3] J. Domínguez, Boundary Elements in Dynamics, Computational Mechanics Publication, Southampton, 1993.
[4] T.A. Cruse, F.J. Rizzo, A direct formulation and numerical solution of the general transient elastodynamic problem, I, Aust. J. Math. Anal. Appl. 22 (1968) 244–259.
[5] A. Peirce, E. Siebrits, Stability analysis and design of time-stepping schemes for general elastodynamic boundary element models, Int. J. Numer. Methods. Engrg. 40 (1997) 319–342.

[6] B. Birgisson, E. Siebrits, A.P. Peirce, Elastodynamic direct boundary element methods with enhanced numerical stability properties, Int. J. Numer. Methods. Engrg. 46 (1999) 871–888.
[7] A. Frangi, G. Novati, On the numerical stability of time-domain elastodynamic analyses by BEM, Comput. Methods Appl. Mech. Engrg. 173 (1999) 403–417.
[8] A. Bamberger, T. Ha-Duong, Formulation variationelle espace-temps pour le calcul par potentiel retardé d'une onde acoustique, Math. Methods Appl. Sci. 8 (1986) 405–435 (598–608).
[9] A. Aimi, M. Diligenti, A new space-time energetic formulation for wave propagation analysis in layered media by BEMs, Int. J. Numer. Methods. Engrg. 75 (2008) 1102–1132.
[10] C. Lubich, Convolution quadrature and discretized operational calculus. I, Numer. Math. 52 (1988) 129–145.
[11] C. Lubich, Convolution quadrature and discretized operational calculus. II., Numer. Math. 52 (1988) 413–425.
[12] C. Lubich, R. Schneider, Time discretization of parabolic boundary integral equations, Numer. Math. 63 (1992) 455–481.
[13] C. Lubich, On the multistep time discretization of linear initial-boundary value problems and their boundary integral equations, Numer. Math. 67 (1994) 365–389.
[14] M. Schanz, H. Antes, Application of 'operational quadrature methods' in time domain boundary element methods, Meccanica 32 (1997) 179–186.
[15] M. Schanz, H. Antes, A new visco- and elastodynamic time domain boundary element formulation, Comput. Mech. 20 (1997) 452–459.
[16] M. Schanz, Application of 3-d Boundary Element formulation to wave propagation in poroelastic solids, Engrg. Anal. Bound. Elem. 25 (2001) 363–376.
[17] C. Zhang, Transient elastodynamic antiplane crack analysis in anisotropic solids, Int. J. Solids Struct. 37 (2000) 6107–6130.
[18] F. García-Sánchez, C. Zhang, A. Sáez, 2-d transient dynamic analysis of cracked piezoelectric solids by a time-domain BEM, Comput. Methods Appl. Mech. Engrg. 197 (2008) 3108–3121.
[19] A.I. Abreu, J.A.M. Carrer, W.J. Mansur, Scalar wave propagation in 2D: a BEM formulation based on the operational quadrature method, Engrg. Anal. Bound. Elem. 27 (2003) 101–105.
[20] M. Schanz, Wave Propagation in Viscoelastic and Poroelastic Continua: A Boundary Element Approach, Lecture Notes in Applied Mechanics, vol. 2, Springer-Verlag, Berlin, Heidelberg, New York, 2001.
[21] X. Wang, R.A. Wildman, D.S. Weile, P. Monk, A finite difference delay modeling approach to the discretization of the time domain integral equations of electromagnetics, IEEE Trans. Antennas Propagat. 56 (2008) 2442–2452.
[22] W. Hackbusch, W. Kress, S.A. Sauter, Sparse convolution quadrature for time domain boundary integral formulations of the wave equation by cutoff and panel-clustering, in: M. Schanz, O. Steinbach (Eds.), Boundary Element Analysis: Mathematical Aspects and Applications, Lecture Notes in Applied and Computational Mechanics, vol. 29, Springer-Verlag, Berlin Heidelberg, 2007, pp. 113–134.
[23] T. Saitoh, S. Hirose, T. Fukui, T. Ishida, Development of a time-domain fast multipole BEM based on the operational quadrature method in a wave propagation problem, in: V. Minutolo, M.H. Aliabadi (Eds.), Advances in Boundary Element Techniques VIII, EC Ltd., Eastleigh, 2007, pp. 355–360.
[24] C. Lubich, Convolution quadrature revisited, BIT Numer. Math. 44 (2004) 503–514.
[25] L. Banjai, Multistep and multistage convolution quadrature for the wave equation: Algorithms and experiments, SIAM J. Sci. Comput. 32 (2010) 2964–2994.
[26] L. Banjai, C. Lubich, An error analysis of Runge–Kutta convolution quadrature, BIT Numer. Math. 51 (2011) 483–496.
[27] L. Banjai, C. Lubich, J.M. Melenk, Runge–Kutta convolution quadrature for non-sectorial operators arising in wave propagation, Numer. Math. 119 (2011) 1–20.
[28] M.P. Calvo, E. Cuesta, C. Palencia, Runge–Kutta convolution quadrature methods for well-posed equations with memory, Numer. Math. 107 (2007) 589–614.
[29] L. Banjai, M. Schanz, Wave propagation problems treated with convolution quadrature and BEM, in: U. Langer, M. Schanz, O. Steinbach, W.L. Wendland (Eds.), Fast Boundary Element Methods in Engineering and Industrial Applications, Lecture Notes in Applied and Computational Mechanics, vol. 63, Springer, 2012, pp. 147–187.
[30] E. Hairer, G. Wanner, Solving ordinary differential equations II: stiff and differential-algebraic problems, springer series in computational mathematics, Springer-Verlag, Berlin Heidelberg, 1991.
[31] P.M. Morse, K.U. Ingard, Theoretical Acoustics, Princeton University Press, 1986.
[32] L. Banjai, S. Sauter, Rapid solution of the wave equation in unbounded domains, SIAM J. Numer. Anal. 47 (2009) 227–249.
[33] S. Erichsen, S.A. Sauter, Efficient automatic quadrature in 3-d Galerkin BEM, Comput. Methods Appl. Mech. Engrg. 157 (1998) 215–224.
[34] M. Messner, M. Messner, F. Rammerstorfer, P. Urthaler, Hyperbolic and elliptic numerical analysis BEM library (HyENA), http://www.mech.tugraz.at/HyENA, 2010 (accessed 22.01.10).
[35] M. Frigo, S.G. Johnson, The design and implementation of FFTW3, in: Proceedings of the IEEE 93, 2005, pp. 216–231 (Special issue on Program Generation, Optimization, and Platform Adaptation).
[36] M. Messner, M. Schanz, An accelerated symmetric time-domain boundary element formulation for elasticity, Engrg. Anal. Bound. Elem. 34 (2010) 944–955.
[37] M. Messner, M. Schanz, A regularized collocation boundary element method for linear poroelasticity, Comput. Mech. 47 (2011) 669–680.