Graz University of Technology
Institute of Applied Mechanics

Preprint Series

Preprint No 01/2015

Comparison of the Convolution Quadrature Method and enhanced inverse FFT with application in elastodynamic Boundary Element Method

Martin Schanz
Institute of Applied Mechanics, Graz University of Technology

Wenjing Ye
Department of Mechanical and Aerospace Engineering
Hong Kong University of Science and Technology

Jinyou Xiao
School of Astronautics, Northwestern Polytechnical University

Published in *Computational Mechanics*, 57(4), 523–536
DOI: 10.1007/s00466-015-1237-z

Latest revision: December 3, 2015

Abstract

Transient problems can often be solved with transformation methods, where the inverse transformation is usually performed numerically. Here, the discrete Fourier transform in combination with the exponential window method is compared with the convolution quadrature method formulated as inverse transformation. Both are inverse Laplace transforms, which are formally identical but use different complex frequencies. A numerical study is performed, first with simple convolution integrals and, second, with a boundary element method (BEM) for elastodynamics.

Essentially, when combined with the BEM, the discrete Fourier transform needs less frequency calculations, but finer mesh compared to the convolution quadrature method to obtain the same level of accuracy. If further fast methods like the fast multipole method are used to accelerate the boundary element method the convolution quadrature method is better, because the iterative solver needs much less iterations to converge. This is caused by the larger real part of the complex frequencies necessary for the calculation, which improves the conditions of system matrix.

1 Introduction

Classical integral equation methods for solving wave propagation problems include the direct discretization of the time-domain integral equation [16], the frequency-domain approaches (see, e.g., [5]) and the dual reciprocity approach (see, e.g., [1]). A relatively new approach is to use convolution quadrature developed by Lubich to discretize the time-domain integral equation [14, 15]. The resulting discretized system involves the integration weights determined by one function, usually the kernel function, in the Laplace domain and another function, typically boundary data, in the time domain. Due to its better stability compared to the classical time discretization approach (see, e.g., [21]), convolution quadrature based boundary element method (CQBEM) has attracted much attention and applications of the method can be found in a variety of areas such as, e.g., anisotropic [26] or cracked piezoelectric materials [8].

Methods working in the frequency domain require a suitable inverse transforms, which is commonly performed numerically. The inverse Fourier transform is one popular method. However for systems with no intrinsic damping and mismatched initial and ending responses, the discrete Fourier transform (DFT) fails to produce accurate results. This is why in practical calculations often a small artificial damping is added to the model, either via the exponential window method as suggested by Kausel and Roësset [12] or by introducing a small viscous or hysteretic damping in the system. The application of the exponential window method in an elastodynamic BE formulation has recently been published in [23, 24]. On the other hand, using the Laplace transform systems without damping can be handled without adding artificial damping. As this topic is not new, a lot of literature exists for numerical inverse transformation techniques. In the framework of BEM and dynamic problems there is an old but nevertheless timely comparison from Narayanan and Beskos [18]. Later, for viscoelastic BE formulations a comparison including the CQBEM can be found in [9].

An important step in the CQBEM is the method for solving the linear system resulting from the application of convolution quadrature. In order to avoid the explicit evaluation of the quadrature weights, most of work employs the decoupled approach suggested by Banjai and Sauter [2] and reformulated for elastodynamics in [20]. In this approach, the time-domain function is also casted in a Laplace form, hence, the decoupled approach works as a transformation method. The similarity of the final discretized systems shared by the CQBEM and the DFT based frequency-domain approach motives a close examination of the relationship between the two approaches. In addition, it would be of interest and is practically valuable to compare the efficiency of the two approaches.

In this paper, we present a study of both methods, focusing on the relationship between the two. We first give a brief description of each method, followed by a discussion on the connection between the two methods. The performance of the two approaches is demonstrated on two analytical convolution integrals. Lastly, applications of the methods to elastodynamic problems are presented and performance comparison is given via numerical results.

2 Numerical approximation of a convolution integral

In any computation of time dependent linear problems, there exists the possibility to use integral transformation methods. The most commonly used method is the Fourier transform beside the Laplace transform. Both allow to compute the solution in the transformed domain and via an inverse transformation the time domain response is obtained. In most engineering computations, essentially, a convolution in time

$$y(t) = f * g = \int_0^t f(t-\tau)\, g(\tau)\, \mathrm{d}\tau \tag{1}$$

has to be performed, where one of the functions may be the response of the system computed by some numerical method and the other is the time dependent load. Beside the classical numerical inverse transformation techniques, such a representation as a convolution integral allows to use as well the CQM as inverse transformation, i.e., to solve the convolution integral (1) numerically. To make a comparison of both techniques, the Fourier transform and the CQM, first, the basic formulae are given in a discrete setting. This allows to see the similarities.

In the remaining text, a Fourier transformed function is denoted by $\tilde{y}$ and the Laplace transformed function by $\hat{y}$. Further, for the Laplace parameter s it holds $s \in \mathbb{C}$ s.t. $\Re s > 0$. The notation $\tilde{y}_\ell = \tilde{y}(\omega_\ell)$ indicates that this function is related to the discret frequency ω_ℓ. The analogous shortcut is used for Laplace transformed functions at discrete complex frequencies s_ℓ. It is assumed that the data are zero for negative times, i.e., $y(t) = 0\ \forall t < 0$. The Heaviside function is denoted by $H(t) = 1\ \forall t > 0$ else zero and the Dirac distribution by $\delta(t)$.

2.1 Frequency domain approach

The Fourier transform of a function f is defined via the integrals

$$\tilde{f}(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-\mathrm{i}\omega t}\, \mathrm{d}t \qquad \leftrightarrow \qquad f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \tilde{f}(\omega)\, e^{\mathrm{i}\omega t}\, \mathrm{d}\omega \tag{2}$$

with the frequency ω. For numerical purposes a discrete version is necessary. Discretizing the time period $t \in [0, T]$ in $N+1$ equal intervals of size Δt yields the discrete Fourier transform (DFT)

$$\tilde{f}_\ell = \Delta t \sum_{k=0}^{N} f(k\Delta t)\, \zeta^{-k\ell} \quad \leftrightarrow \quad f(n\Delta t) = \frac{1}{\Delta t\,(N+1)} \sum_{\ell=0}^{N} \tilde{f}_\ell \zeta^{\ell n} \qquad \text{with } \zeta = e^{\mathrm{i}\frac{2\pi}{N+1}}\,. \tag{3}$$

In order to apply the DFT technique, the initial and ending responses within the time period of interest must match owing to the periodic nature of DFT, or otherwise, large 'wrap-around' errors will result. This condition unfortunately cannot be satisfied for most problems. An effective and general method to overcome this issue is the exponential window method (EWM). It is frequently used in signal processing and has been introduced in structural engineering by Kausel and Roësset [12]. In this approach, a "damped" solution $f_D(t) = f(t)\, e^{-\eta t}$ is obtained first with

$\eta > 0$. It is then scaled up to retrieve the original solution. The new continuous definition of the transformation pair is

$$\begin{aligned} \tilde{f}_D(\omega) &= \int_{-\infty}^{\infty} f(t)\, e^{-\eta t} e^{-\mathrm{i}\omega t}\, \mathrm{d}t = \int_{-\infty}^{\infty} f(t)\, e^{-\mathrm{i}\bar{\omega} t}\, \mathrm{d}t = \tilde{f}(\bar{\omega}) \\ f(t) &= f_D(t)\, e^{\eta t} = \frac{1}{2\pi} \int_{-\infty}^{\infty} \tilde{f}(\omega)\, e^{\mathrm{i}\omega t}\, \mathrm{d}\omega = \frac{1}{2\pi} \int_{-\infty-\mathrm{i}\eta}^{\infty-\mathrm{i}\eta} \tilde{f}(\bar{\omega})\, e^{\mathrm{i}\bar{\omega} t}\, \mathrm{d}\bar{\omega}\,. \end{aligned} \tag{4}$$

The corresponding discrete counterpart, i.e., the modified DFT called in the following EWM-DFT , is

$$\tilde{f}_\ell^* = \Delta t \sum_{k=0}^{N} f(k\Delta t)\, \zeta^{-k\ell} e^{-\eta k \Delta t} \quad \leftrightarrow \quad f(n\Delta t) = \frac{1}{\Delta t\,(N+1)} e^{\eta n \Delta t} \sum_{\ell=0}^{N} \tilde{f}_\ell^*\, \zeta^{\ell n}\,. \tag{5}$$

The notation $\tilde{f}_\ell^*$ has been introduced to make clear that these values are related to a complex valued frequency and not to the real valued frequency of (3). Exploring the second formula in (5), i.e., the inverse discrete transformation, numerical problems are obvious when η is chosen too large. Kausel and Roësset [12] developed an empirical formula for the damping parameter

$$\eta < \frac{m \ln 10}{T} \tag{6}$$

if m is the number of significant digits.

Remark 1. A close inspection of the last integrals in each line of (4) shows that a substitution $s = \mathrm{i}\bar{\omega}$ brings up the definition of the usual Laplace transform, if additionally the condition $f(t) = 0$ for $t < 0$ is imposed. Hence, this method mimics the Laplace transform and the EWM-DFT is an approximation of the inverse Laplace transform. Consequently, the numerical inverse Laplace transform is similar. Comparing the formula (5) with the method of Durbin [7] it is found to be the same.

The above sketched method allows to apply the FFT on non-periodic functions. However, the method still suffers from Gibb's phenomenon, i.e., the truncation of high frequency parts of the solution causes the time signal to oscillate at discontinuous points. To suppress these unwelcome parts of the solution a filter can be used. A common one is the Blackman filter, which multiplies the frequency response with

$$w_n = 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right) \quad 0 \le n \le \frac{N}{2} - 1\,. \tag{7}$$

This filter will be used for all calculations in the following. An additional benefit of this filter is that the amplification of the oscillations caused by the exponential weighting $e^{\eta \Delta t}$ in (4) is decreased, which allows a larger m to be used and, hence, a more accurate solution.

Summarizing, to compute the convolution integral (1) both functions $f(t)$ and $g(t)$ are transformed with the first equation in (5). This allows to simply multiply them to obtain the convolution in the transformed domain. Finally, the product is transformed back by the second equation

in (5) using the filter (7), which results in

$$y(n\Delta t) = \frac{\left(e^{\eta\Delta t}\right)^n}{\Delta t\,(N+1)} \sum_{\ell=0}^{N} \tilde{f}_\ell^* \tilde{g}_\ell^* w_\ell \zeta^{\ell n} \,. \tag{8}$$

This final result will be compared with the CQM .

2.2 Convolution quadrature method

The CQM approximates the convolution integral (1) with a quadrature formula, of which integration weights are determined based on the Laplace transform of one of the functions f or g. Let assume as above a discretization of the time interval $t \in [0,T]$ in $N+1$ equal time steps with size Δt. Further, it is assumed that the Laplace transform $\hat{f}$ exists. The final formula of the CQM is

$$y(n\Delta t) = \sum_{k=0}^{n} \frac{\mathcal{R}^{-(n-k)}}{N+1} \sum_{\ell=0}^{N} \hat{f}(s_\ell)\, \zeta^{(n-k)\ell} g(k\Delta t) \quad \text{with } s_\ell = \frac{\gamma\left(\zeta^{-\ell}\mathcal{R}\right)}{\Delta t} \,, \tag{9}$$

with the abbreviation $\gamma(z)$ for the quotient of the characteristic polynomials of the underlying multi-step method. For a Backward Differential formula of order two (BDF 2) this expression is $\gamma(z) = 3/2 - 2z + z^2/2$. The parameter $\mathcal{R}$ must be $0 < \mathcal{R} < 1$ and is usually determined by $\mathcal{R}^{N+1} = \sqrt{\varepsilon_{CQ}}$, with $10^{-20} < \varepsilon_{CQ} < 10^{-4}$ chosen according to the problem. The latter range is a recommendation and not a bound (for studies see [19]). Details of the method can be found in the original papers by Lubich [14, 15] or with an extension to Runge-Kutta methods, e.g., in [3].

As suggested by Banjai and Sauter [2] and reformulated for elastodynamics in [20] the above formula, which works directly in time domain, can be split such that it works like a transformation method. A first step ist to extend the outer sum in (9) to N with zeros. This corresponds to zero integration weights with negative index. A condition which is fulfilled by all causal functions. Then the sums are interchanged to

$$y(n\Delta t) = \frac{\mathcal{R}^{-n}}{N+1} \sum_{\ell=0}^{N} \hat{f}(s_\ell)\, \zeta^{n\ell} \sum_{k=0}^{N} \mathcal{R}^k g(k\Delta t)\, \zeta^{-k\ell} \,, \tag{10}$$

which shows the discrete transformation pair

$$\hat{g}_\ell = \sum_{k=0}^{N} \mathcal{R}^k g(k\Delta t)\, \zeta^{-k\ell} \quad \rightarrow \quad y(n\Delta t) = \frac{\mathcal{R}^{-n}}{N+1} \sum_{\ell=0}^{N} \hat{f}(s_\ell)\, \hat{g}_\ell \zeta^{n\ell} \,. \tag{11}$$

It must be remarked that this expression is not a transformation pair as discussed above but the final result of the convolution combined with the discrete transformation of one function in the convolution.

In the following tests, not only the BDF 2 is used but as well some Runge-Kutta methods. The formula (11) has then the same structure but some details differ. Let a s-stage Runge-Kutta method be given by its Butcher tableau $\begin{array}{c|c} \mathsf{c} & \mathsf{A} \\ \hline & \mathsf{b}^T \end{array}$ with $\mathsf{A} \in \mathbb{R}^{s\times s}$, $\mathsf{b},\mathsf{c} \in \mathbb{R}^s$. Further, let us assume that the method is A-stable and L-stable. The latter requires that $\mathsf{b}^T\mathsf{A}^{-1} = (0,0,\ldots,1)$ holds.

For such kind of Runge-Kutta methods the characteristic function $\gamma(z)$ in (9) changes to a matrix equation

$$\Delta(z) = \left(\mathsf{A} + \frac{z}{1-z}\mathbb{1}\mathsf{b}^T\right)^{-1} \quad \text{with } \mathbb{1} := (1,1,\ldots,1)^T . \tag{12}$$

The formula corresponding to (9) is then

$$y((n+1)\Delta t) = \mathsf{b}^T\mathsf{A}^{-1}\sum_{k=0}^{n}\frac{\mathcal{R}^{-(n-k)}}{N+1}\sum_{\ell=0}^{N}\hat{f}\left(\frac{\Delta\left(\mathcal{R}\zeta^{-\ell}\right)}{\Delta t}\right)\zeta^{(n-k)\ell}\mathsf{g}_k , \tag{13}$$

where g_k is the vector of the function $g(t)$ at the stage's discrete times within each time step k. The same reasoning as above results in the discrete transformation pair

$$\hat{\mathsf{g}}_\ell = \sum_{k=0}^{N}\mathcal{R}^k\mathsf{g}_k\zeta^{-k\ell} \;\rightarrow\; y((n+1)\Delta t) = \mathsf{b}^T\mathsf{A}^{-1}\frac{\mathcal{R}^{-n}}{N+1}\sum_{\ell=0}^{N}\hat{f}\left(\frac{\Delta\left(\mathcal{R}\zeta^{-\ell}\right)}{\Delta t}\right)\hat{\mathsf{g}}_\ell\zeta^{n\ell} . \tag{14}$$

Now the function $\hat{f}$ is dependent on a matrix of frequencies $\Delta(z)$, i.e., in each step the function is evaluated corresponding to the s stages. Details of the derivation and numerical realisation can be found in [3].

3 Comparison of both approaches

For the comparison of both methods an inspection of the formulas for the EWM-DFT (8) and the corresponding one for the CQM (11) is helpful. Note, for this formal comparison the Blackman window will not be considered. There are several similarities and, essentially, two observations can be made:

1. The obtained transformation formulas are obviously weighted FFT's, i.e., the different complex valued frequencies are not equally important for the final result. The weighting factors have a similar structure and the formulas used to determine them are in both cases related to the numerical precision of the computation. For the EWM-DFT it is η or using the estimate (6) m and for the CQM it is ε_{CQ}. The relation is

$$\mathcal{R} \equiv e^{-\eta\Delta t} \quad \Rightarrow\; \varepsilon_{CQ}^{\frac{1}{2(N+1)}} = e^{-\eta\frac{T}{N+1}} \quad \Rightarrow\; \varepsilon_{CQ}^{\frac{1}{2}} = e^{-\eta T} . \tag{15}$$

   Playing with some realistic numbers for the CQM gives the combinations

$$\begin{aligned} \varepsilon_{CQ} = e^{-10} \approx 4.5\cdot 10^{-5} \quad &\Rightarrow\; 5 \;= \eta T < m\ln 10 \quad \Rightarrow\; 2.17 < m \\ \varepsilon_{CQ} = e^{-20} \approx 2\cdot 10^{-9} \quad &\Rightarrow\; 10 = \eta T < m\ln 10 \quad \Rightarrow\; 4.34 < m , \end{aligned}$$

   which results in the first line in values for m comparable to Kausel and Roësset [12]. However, due to Gibbs oscillation, if no filter is employed, the value of m should not exceed 3 in the application of elastodynamic BEM with a Heaviside load as reported in [23].

   Numerical comparisons of the weighting factors $\mathcal{R}$ or $e^{-\eta\Delta t}$ with $\varepsilon_{CQ} = 10^{-5}$ and $m = 3$ show very close behaviors either in size as well as in the course of increasing k. Hence, it can be concluded that the forward transformation is very similar.

2. The inverse transformations are different in two aspects. First, in the EWM-DFT both functions to be multiplied are discrete transformations. In the CQM only one of the two functions is a discrete transformation. The other function is a continuous transformation, which is evaluated at discrete frequencies. Secondly, in the EWM-DFT both functions are multiplied at the same frequency. Contrary, in the CQM the second function is evaluated at different frequencies s_ℓ, determined by the time step size and the underlying multistep method. It may be instructive to study the different frequency distributions in both formulas. In Fig. 1, the used complex frequencies are plotted in the complex plane for both methods using $\Delta t = 0.0391, T = 5, \varepsilon_{CQ} = 10^{-10}$, and $m = 3$. Note, all the time values are assumed to be normalized. Hence no dimension is given. For the CQM different time stepping methods can be used, which influences the values of s_ℓ. That is why in Fig. 1 beside the used complex frequencies of the EWM-DFT three curves for the CQM are given. They are indicated by the names of the multistep methods (BDF 1 and BDF 2) or of the Runge-Kutta methods (Radau IIA and Lobatto IIIC, both in the 3-stage version, see appendix A for the Butcher tableau). The influence of the time stepping method on the CQM results is not discussed in this paper and can be found in [3].

   A different time step size obviously changes the used complex frequencies in the classical way, i.e., a smaller time step size results in larger frequencies and vice versa. However, important for this study here is the relation between the EWM-DFT and the CQM . The comparison shows that the Radau IIA method uses frequencies with imaginary parts larger than the EWM-DFT . The other methods uses smaller or similar frequencies in the Lobatto IIIC case. In Fig. 1b, the distance to the imaginary axis is shown, which is a measure of the used damping in the EWM-DFT . Clearly, the CQM is further away for this choice of parameters and the following study will confirm that the results are more damped. This distance is obviously influenced by the way the values for ε_{CQ} and m are set. Hence, the choice may even produce a larger damping in EWM-DFT compared to the CQM . This choice is problem dependent but in the EWM-DFT it can only be shifted to higher values using the Blackman window.

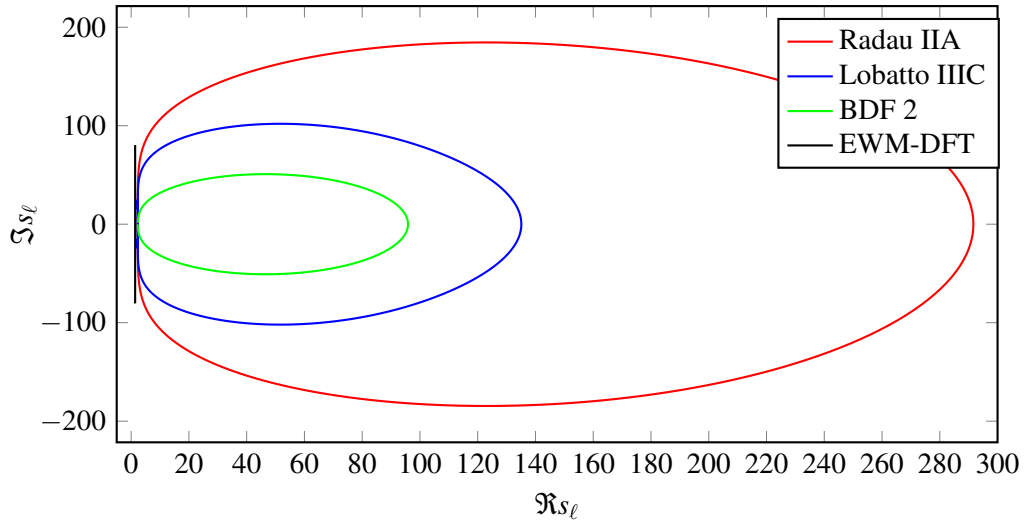
Next, the two methods are tested for two sets of functions. In both cases the function $f(t) = \delta(t-1)$ is chosen. This choice is related to the kernel function in a BE formulation of acoustics, which has the same temporal behavior for $r/c = 1$ but is divided by $4\pi r$. It is a prototype for any kernel function describing waves. The load function $g(t)$ is chosen differently, either

- Heaviside function: $g(t) = H(t) - H(t-b)$. This results in the exact solution of

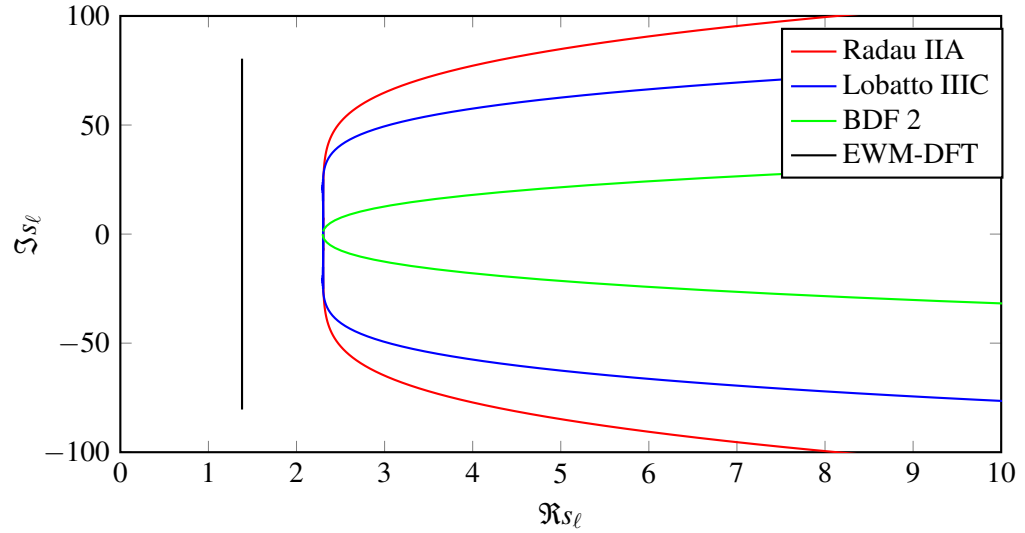
$$f * g = \int_0^t \delta(t-\tau-1)\,[H(\tau) - H(\tau-b)]\,\mathrm{d}\tau = H(t-1) - H(t-(b+1)) \tag{16}$$

- Sine function: $g(t) = \sin(Tt)$. This results in the exact solution of

$$f * g = \int_0^t \delta(t-\tau-1)\sin(T\tau)\,\mathrm{d}\tau = \sin(T(t-1))\,H(t-1) \tag{17}$$



(a) Overall picture



(b) Zoom for small real parts

Figure 1: Real part versus the imaginary part of the used complex frequencies s_ℓ for $\Delta t = 0.0391$

For all tests an overall observation time $T = 5$ is used and $\varepsilon_{CQ} = 10^{-10}$. The parameter for the EWM-DFT is chosen for the Heaviside load with $m = 3$ and for the Sine load with $m = 6$. Both have found to be the good choice by a trial and error search. It must be remarked that only the usage of the Blackman filter allows to use high values of m. Taking $N = 128$ results in a time steps size of $\Delta t = 0.0391$. In Fig. 2, the results for the Heaviside load with $b = 3$ are presented with and without the Blackman window. For the results without the Blackman window the value $m = 2$ has to be chosen, else the shown oscillations would be much higher. In Fig. 3, the results for the Sine load are displayed. From now on all presented results are obtained using the Blackman window. In both load cases all results are acceptable. The CQM has oscillations at the jumps but they decrease within a small distance, where the Runge-Kutta method performs best. For the Sine load no jumps have to be resolved and, consequently, the results are much nicer. The CQM with the BDF 2 smoothes the kink in the beginning and all CQM solutions show a slight nearly not visible phase-shift in the first bow of the Sine. The EWM-DFT shows no shift. Both methods show a small offset from zero for $t < 1$. For EWM-DFT , this offset depends on the value of m. For $m \approx 3$ this offset would be $\approx 10^{-6}$. It decreases to $\approx 10^{-8}$ if $m = 6$ is set. For CQM , the offset is $\approx 10^{-6}$ and is relatively independent of ε_{CQ}. If the original version of the CQM as proposed by Lubich [14] is used, i.e., not reformulated as transformation and utilizing the integration weights in time domain, a much smaller nearly zero value can be obtained. It must be remarked again that only with the Blackman window such results can be obtained and the value of m has to be set to high values. The above results show as well that a normal DFT without the EWM and the Blackman window is not suitable for such functions as above.

The above study shows that both methods work well. Next, the error will be quantatively measured. For the CQM a discrete ℓ^2 error estimate exists for the above used kernel function [3]. It is based on the assumption

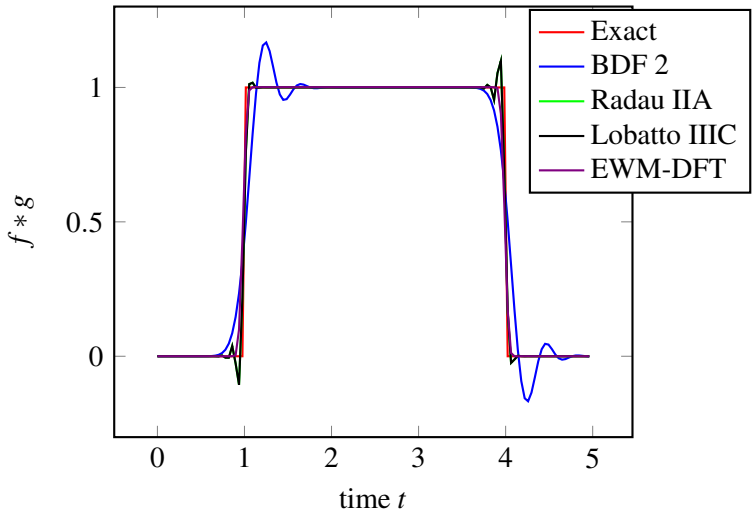
$$g(t) \in L^2(\mathbb{R}) \;\text{ with } g(t) \equiv 0 \;\forall t < 0 \quad \text{and} \quad |\hat{g}| \leq C\,|s|^{-\mu} \;\forall \Re s > 0 \;\text{ with } \mu > \frac{1}{2}\,.$$

Under this assumption the CQM converges for a Runge-Kutta method of order p with $\mathscr{O}(\Delta t^{\alpha})$ and

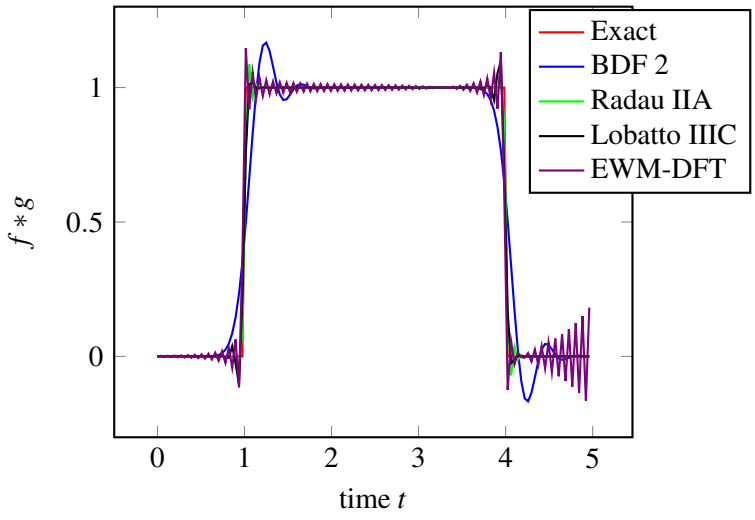
$$\alpha = \min\left\{ \frac{(2\mu - 1)\,p}{2\,(p+1)}, p \right\}\,. \tag{18}$$

The smoothness assumption on $g(t = 0)$ is violated by the above used functions, but the study presented in [3] has shown that the convergence order can be shown numerically also for such non-smooth functions. For the Heaviside function $\mu = 1$ and for the Sine $\mu = 2$ holds. Both 3-stage Runge-Kutta methods used are A-stable and have order $p = 5$ or $p = 4$ for the Radau IIA and Lobatto IIIC rule, respectively. With this in mind the relative error

$$\varepsilon^{\Delta t} = \left(\frac{\sum\limits_{n=0}^{N} \left(y(t_n) - y_n^{\Delta t} \right)^2}{\sum\limits_{n=0}^{N} y(t_n)^2} \right)^{\frac{1}{2}} \tag{19}$$



(a) With Blackman window



(b) Without Blackman window

Figure 2: Solution with different multistep methods and EWM-DFT : Heaviside function

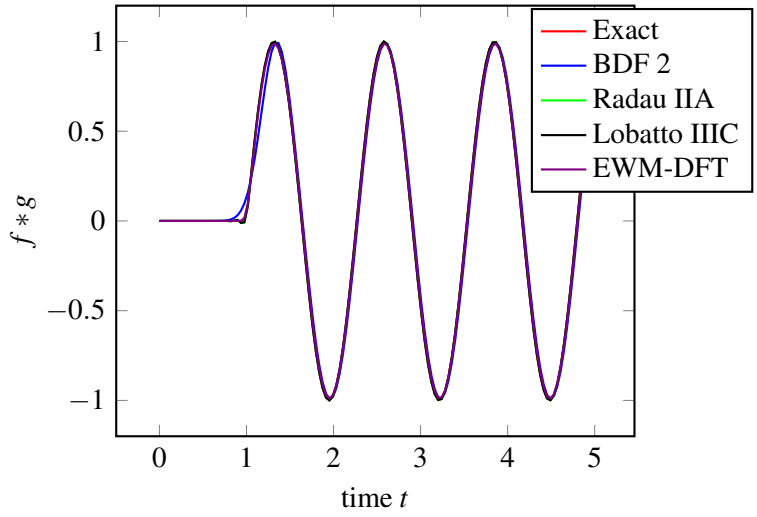


Figure 3: Solution with different multistep methods and EWM-DFT : Sine function

is computed. In (19), $y(t_n)$ denotes the exact solution at time $t_n = n\Delta t$ and $y_n^{\Delta t}$ is the approximated solution by either the CQM or EWM-DFT . This error is displayed in Fig. 4 and Fig. 5 versus the time step size in a logarithmic scale for the Heaviside and Sine load. The differences in the error plots in Fig. 4 and Fig. 5 is, whether the second jump visible in Fig. 2 is included in the observation time or not. This is governed by the parameter b for the Heaviside load in (16). For the results in Fig. 4 $b = 10$ is used and in Fig. 5 $b = 3$. The solution presented in Fig. 2 corresponds to the latter case. Additionally, the convergence rates of a BDF 1 and the 3-stage Radau IIA method computed with (18) are displayed by the two straight lines. It can be observed that the CQM has the predicted convergence rate. The EWM-DFT seems to have the similar convergence rate as that of the 3-stage Runge-Kutta method.

The same error study is performed for the Sine load and displayed in Fig. 6. The same tendency as before is observed, where the EWM-DFT seems to have for this function a higher convergence rate than the Radau IIA. The fast convergence rate of the EWM-DFT might be explained by the link between EWM-DFT and CQM . If one sets the quotient of the characteristic polynomial of the underlying multi-step method to be $\gamma(\zeta) = -\log\zeta$, one can show that the formula of the EWM-DFT (8) is equivalent to that of the CQM (11). However, it must be remarked that such a choice of the characteristic polynomial is not possible in CQM in defining the weights (i.e., the necessary series expansion does not exist). By neglecting this aspect, EWM-DFT can be regarded as CQM with a special time stepping method, the quotient of which is $\gamma(\zeta) = -\log\zeta$. It is well known that the accuracy of the stable linear multistep scheme is measured by how well the quotient of the generating polynomial approximates $\log\zeta$ as ζ approaches 1. This may explain the more accurate solutions obtained by EWM-DFT at least for smooth functions. For non-smooth functions, such as the Heaviside function, the convergence rate of EWM-DFT may be reduced due to Gibbs oscillations.

For both methods the error does not go beyond a certain value. This value is larger for the CQM than for the EWM-DFT , as long as a large $m = 6$ is used. For $m = 3$ the plateau for the

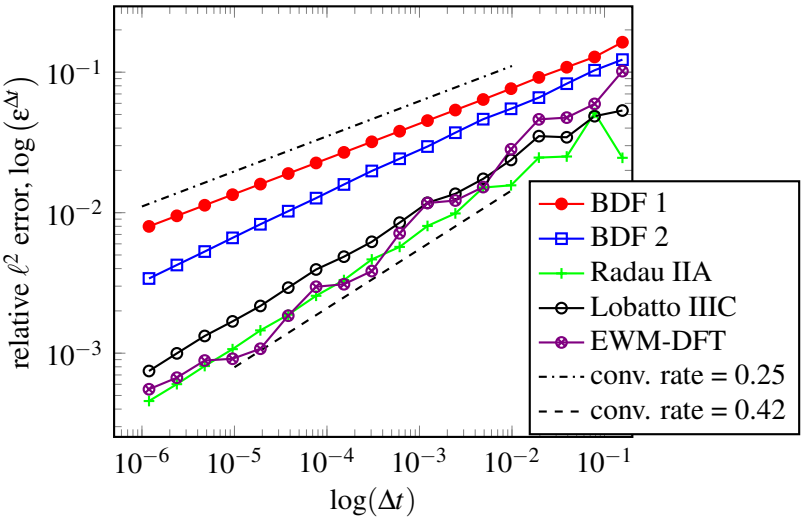


Figure 4: Log-log-plot of the relative ℓ^2 error: Heaviside function with one jump ($b = 10$)

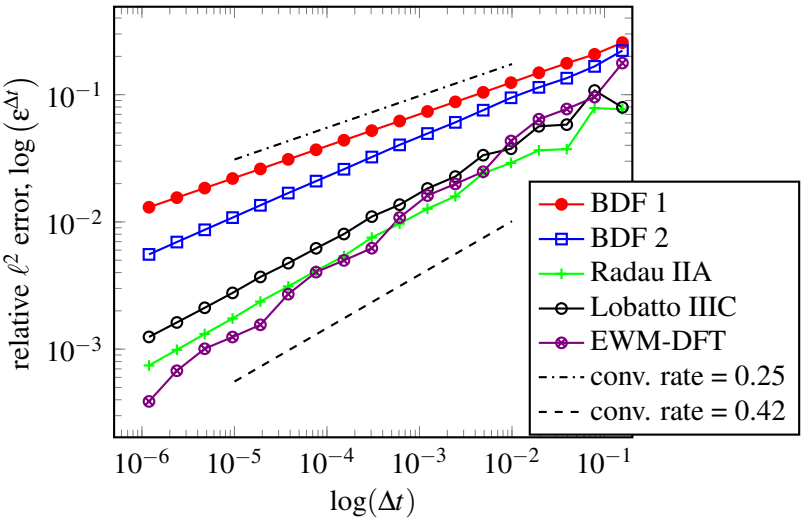


Figure 5: Log-log-plot of the relative ℓ^2 error: Heaviside function with two jumps ($b = 3$)

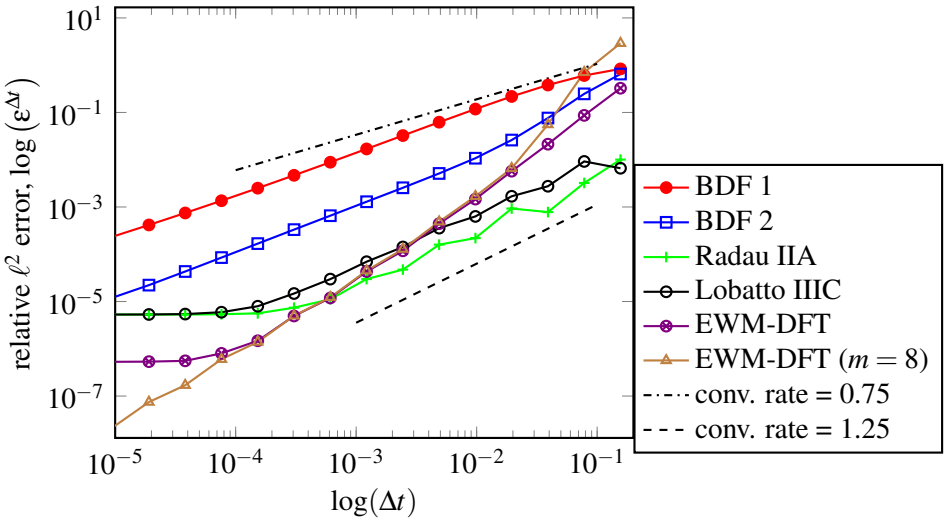


Figure 6: Log-log-plot of the relative ℓ^2 error: Sine

EWM-DFT would be at $\approx 0.5 \cdot 10^{-3}$. The reason for this stagnation of the error in case of the EWM-DFT is the approximation of the initial zeros and, hence, can be minimized by a large value of m. In fact, for $m = 8$, there is no plateau within the time step sizes considered as shown in Fig. 6. For the CQM this plateau is mainly caused by the slight phase-shift in the first bow of the results. Playing with ε_{CQ} this shift can be affected but not avoided. On the other hand the value of ε_{CQ} influences the zero offset as well. For the presented study it has been tried to find good values. However, it must be clearly stated that the value of m influences the EWM-DFT much more than the ε_{CQ} the CQM .

Above the accuracy of the methods has been compared. The missing point is a discussion on the numerical effort. The efficiency of the method can be defined as the numerical effort necessary to obtain a defined error level. As in the above examples the numerical effort is determined by a function evaluation for each necessary complex frequency, it is sufficient to find the necessary number of frequencies, i.e., the number N to obtain a certain error. The method with the lowest number of evaluations is the most efficient one. In Tab. 1, the numbers N for different error levels and the used loading functions are presented. However, care must be taken with the Runge-Kutta methods. In the above, 3-stage methods are used, which means within one time step 3 function evaluations are necessary. That is why in the lines of Tab. 1 the numbers for the Runge-Kutta methods are multiplied with 3. The result confirms the observations of the above studies. Within the CQM the Radau IIA method performs best. In comparison with the EWM-DFT the CQM is not such efficient, i.e., it needs more frequency evaluations particularly at small time steps.

The above study has shown that both methods yield good results, whereas the EWM-DFT seems to be the most efficient one, as long as the parameter m is chosen well and a Blackman window is used for non-smooth functions. Without both windowing techniques the DFT would fail for the tested non-periodic functions. In some sense the CQM can be declared to be the more robust method as only the physical accesible time step size must be chosen. But, this nice

Error	EWM-DFT	CQM BDF 2	CQM Lobatto IIIC	CQM Radau IIA
	Heaviside load $b = 10$			
0.05	290	650	170*3=510	120*3=360
0.01	7450	140000	7300*3=21900	3400*3=10200
	Heaviside load $b = 3$			
0.05	510	3740	380*3=1140	300*3=900
0.01	12750	660000	20600*3=61800	9900*3=29700
	Sine load			
0.001	670	5350	530*3=1590	370*3= 1110
0.0005	1050	10100	950*3=2850	630*3=1890

Table 1: Necessary function evaluations for a given accuracy

property comes along with more function evaluations. Note, the observations may be different if different functions are tested but the above choice should give a good indicator for the intended application in BEM.

4 Boundary element method

The application of the above shown transformation methods to BE formulations is presented next. The model problem chosen is linear elastodynamics. Obviously, the transformation techniques can be applied to any linear problem whose time domain description involves convolution integrals.

4.1 Governing equations

The assumption of linear elasticity and a linear geometry description results in the Lamè-Navier equation for the displacement field $\mathbf{u}(\mathbf{x},t)$

$$\begin{aligned} c_1^2\nabla(\nabla\cdot\mathbf{u}(\mathbf{x},t)) - c_2^2\nabla\times(\nabla\times\mathbf{u}(\mathbf{x},t)) &= \frac{\partial^2\mathbf{u}}{\partial t^2}(\mathbf{x},t) \quad (\mathbf{x},t)\in\Omega\times(0,T) \\ \mathbf{u}(\mathbf{y},t) &= \mathbf{g}_D(\mathbf{y},t) \quad (\mathbf{y},t)\in\Gamma_D\times(0,T) \\ \mathbf{t}(\mathbf{y},t) &= \mathbf{g}_N(\mathbf{y},t) \quad (\mathbf{y},t)\in\Gamma_N\times(0,T) \\ \mathbf{u}(\mathbf{x},0) = \frac{\partial\mathbf{u}}{\partial t}(\mathbf{x},0) &= \mathbf{0} \quad (\mathbf{x},t)\in\Omega\times(0)\,. \end{aligned} \tag{20}$$

In this hyperbolic partial differential equation the position in the three-dimensional Euclidean space $\mathbb{R}^3$ is denoted by $\mathbf{x}$ and the time by $t\in(0,T)$. The material properties of the solid are represented by the wave speeds

$$c_1 = \sqrt{\frac{E\,(1-\nu)}{\rho\,(1-2\nu)\,(1+\nu)}} \qquad c_2 = \sqrt{\frac{E}{\rho 2\,(1+\nu)}}\,, \tag{21}$$

with the material data Young's modulus E, Poisson's ration ν, and the mass density ρ. The partial differential equation in (20) is given in the spatial domain Ω for all times $0<t<T$. The

boundary Γ of the domain Ω is subdivided into two disjoint sets Γ_D and Γ_N at which boundary conditions are prescribed. The Dirichlet boundary condition is the second statement of (20) and assigns a given datum $\mathbf{g}_D$ to the displacement $\mathbf{u}$ on the part Γ_D of the boundary. Similarly, the Neumann boundary condition is the third statement in which the datum $\mathbf{g}_N$ is assigned to the surface traction $\mathbf{t}$, which is defined by

$$\mathbf{t}(\mathbf{y},t) = (\mathcal{T}\mathbf{u})(\mathbf{y},t) = \lim_{\Omega\ni\mathbf{x}\to\mathbf{y}\in\Gamma}[\sigma(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{y})]\,. \tag{22}$$

In (22), σ is the stress tensor depending on the displacement field $\mathbf{u}$ according to the strain-displacement relationship and Hooke's law. For later purposes the traction operator $\mathcal{T}$ is defined, which maps the displacement field $\mathbf{u}$ to the surface traction $\mathbf{t}$. The boundary conditions have to hold for all times and may be also prescribed in each direction by different types, e.g., roller bearings. Finally, in the last statement of (20) the condition of a quiescent past is given which implies homogeneous initial conditions.

The integral representation formula may be derived from the dynamic reciprocal identity [22] or also from a weighted residual statement. Using the fundamental solution $\mathbf{U}(\mathbf{x}-\mathbf{y},t-\tau)$ of equation (20) (see for instance [11]) and taking the boundary trace of the representation formula, the boundary integral equation for elastodynamics can be obtained (see, e.g., [5, 19]). Using operator notation, this boundary integral equation is

$$(\mathcal{V}*\mathbf{t})(\mathbf{x},t) = \mathcal{C}(\mathbf{x})\mathbf{u}(\mathbf{x},t) + (\mathcal{K}*\mathbf{u})(\mathbf{x},t) \qquad (\mathbf{x},t)\in\Gamma\times(0,\infty)\,. \tag{23}$$

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

$$(\mathcal{V}*\mathbf{t})(\mathbf{x},t) = \int_0^t\int_\Gamma \mathbf{U}(\mathbf{x}-\mathbf{y},t-\tau)\mathbf{t}(\mathbf{y},\tau)\,\mathrm{d}s_\mathbf{y}\,\mathrm{d}\tau \tag{24a}$$

$$\mathcal{C}(\mathbf{x}) = I + \lim_{\varepsilon\to0}\int_{\partial B_\varepsilon(\mathbf{x})\cap\Omega} (\mathcal{T}_\mathbf{y}\mathbf{U})^\top(\mathbf{x}-\mathbf{y},0)\,\mathrm{d}s_\mathbf{y} \tag{24b}$$

$$(\mathcal{K}*\mathbf{u})(\mathbf{x},t) = \lim_{\varepsilon\to0}\int_0^t\int_{\Gamma\setminus B_\varepsilon(\mathbf{x})} (\mathcal{T}_\mathbf{y}\mathbf{U})^\top(\mathbf{x}-\mathbf{y},t-\tau)\mathbf{u}(\mathbf{y},\tau)\,\mathrm{d}s_\mathbf{y}\,\mathrm{d}\tau\,. \tag{24c}$$

In these expressions, $B_\varepsilon(\mathbf{x})$ denotes a ball of radius ε centered at $\mathbf{x}$ and $\partial B_\varepsilon(\mathbf{x})$ is its surface. Note that the single layer operator (24a) involves a weakly singular integral and the double layer operator (24c) has to be understood in the sense of a Cauchy principal value.

4.2 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

$$\Gamma_h = \bigcup_{e=1}^{N_e}\tau_e\,. \tag{25}$$

τ_e denote boundary elements, e.g., surface triangles as in this work, and their total number is N_e. Now, the boundary functions $\mathbf{u}$ and $\mathbf{t}$ are approximated with shape functions φ_i or ψ_j, which are defined with respect to the geometry partitioning (25), and time dependent coefficients u_k^i and t_k^j. This yields for the k-th component of the data

$$u_k(\mathbf{y},t) = \sum_{i=1}^{N} u_k^i(t)\varphi_i(\mathbf{y}) \qquad \text{and} \qquad t_k(\mathbf{y},t) = \sum_{j=1}^{M} t_k^j(t)\psi_j(\mathbf{y})\,. \tag{26}$$

Inserting these spatial shape functions in the boundary integral equation (23) and a collocation method results in the semi-discrete equation system

$$\mathsf{V} * \mathsf{t} = \mathsf{C}\mathsf{u} + \mathsf{K} * \mathsf{u}\,. \tag{27}$$

In Equation (27), 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. Hence, the matrices V,K and C are the discrete version of the single layer and double layer potential and the integral free term in (24). The vectors t and u collects the tractions and displacements at all nodes.

4.3 Application of transformation methods

In principle, the remaining task is to perform a time discretisation or to use an integral transform with respect to time. Choosing the latter, the hyperbolic problem is reduced to an elliptic problem. Discretising the convolution in (27) with (8) results in the EWM-DFT version and with (11) or (14) gives the CQM version. Independently of the used method, the solution of (27) is reduced to the solution of N elliptic problems for the complex 'frequency' $s_\ell, \ell = 0,1,\ldots,N-1$

$$\hat{\mathsf{V}}(s_\ell)\,\mathsf{t}_\ell^* - \hat{\mathsf{K}}(s_\ell)\,\mathsf{u}_\ell^* = \mathsf{C}\mathsf{u}_\ell^*\,. \tag{28}$$

The difference, as mentioned in section 3, lies in the used complex frequencies s_ℓ. Recalling their definition it is

$$\text{for EWM-DFT } s_\ell = \eta - \mathrm{i}\frac{2\pi}{\Delta t\,(N+1)}\ell \tag{29a}$$

$$\text{for CQM } s_\ell = \frac{\gamma\left(\zeta^{-\ell}\mathcal{R}\right)}{\Delta t}\,. \tag{29b}$$

The matrices and vectors in (28) are the transformed counterparts of (27). For the CQM it remains to specify the underlying time stepping method $\gamma(z)$. In case of a Runge-Kutta method $\gamma(z)$ is replaced by the matrix equation (12). With these operations the time stepping procedure is reduced to the solution of decoupled Laplace domain problems.

4.4 Numerical solution and fast methods

The remaining part is the numerical realisation of the above given procedure. The following test have two parts. One is concerned with the efficiency of the transformation methods in relation to

the accuracy. For this a normal BE formulation without any fast techniques or iterative solvers is used to study only the influence of the transformation method. The other test is concerned with the overall efficiency of the transformation methods in connection with a fast BE formulation. By virtue of the available codes, two different numerical realisations are used.

Normal BE formulation All regular integrals are performed with Gaussian quadrature formulas. The strong singular integral is regularized with partial integration (see, e.g., [13]). The resulting weakly singular integrals are solved using Duffy coordinates [6]. The geometrical discretization is done with linear triangles and the Dirichlet data are approximated by piecewise linear shape functions and the Neuman data by piecewise constant shape functions. For the solution a direct solver is used.

Fast BE formulation A so-called 'black-box' fast multipole method (FMM) following the paper of Ying et al. [25] is used for the test. Details of the realisation can be found in [4]. As above, triangles are used for the geometrical discretization. The Cauchy data are approximated with non-conforming quadratic shape functions and GMRES is used to solve the system of equations. A block diagonal pre-conditioner is applied as well.

5 Numerical BE results

To study the behavior of both transformation methods a standard example for time domain BE formulations is used. It is a 3d-column with boundary and material data set such that the results can be compared to the available analytical solution.

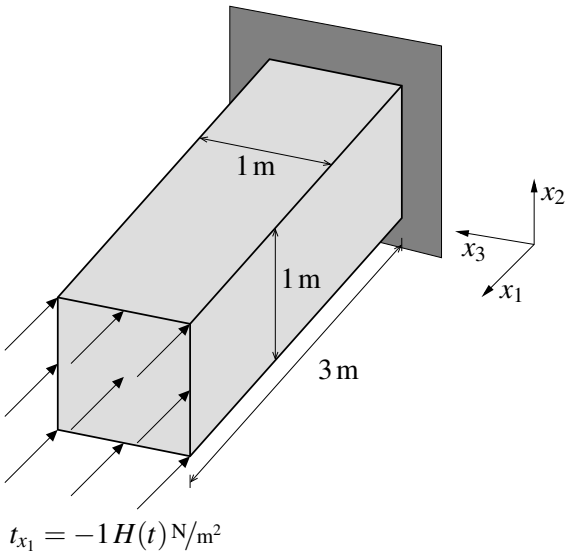
This 3-d column has the size $\ell_1 = 3.0\,\mathrm{m}$ and $\ell_2 = \ell_3 = 1.0\,\mathrm{m}$, as depicted in Fig. 7. It has zero displacements on one end and on the other end the normal traction $t_{x_1} = -1\,H(t)\,\mathrm{N/m^2}$ is prescribed. The material parameters of steel ($E = 2.11 \cdot 10^{11}\,\mathrm{N/m^2}, \nu = 0, \rho = 7850\,\mathrm{kg/m^3}$) are taken. Note, Poisson's ratio is artificially set to zero to allow the comparison with the 1-d analytical solution [10]. In order to determine a suitable time step size, the dimensionless value

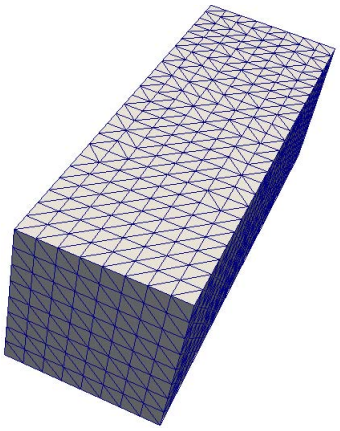
$$\beta = \frac{c_1 \Delta t}{h} \tag{30}$$

is introduced, which relates the spatial (mesh size h) to the temporal discretization (time step size Δt). This CFL-condition like number is studied for the CQM calculations in several publications, e.g., [19] or for the Runge-Kutta based CQM [3]. Please note, the time step size used in case of Runge-Kutta methods is that of the stages and not the real time step size. This is made to allow the comparison of the numerical effort with the one-stage methods. A similar number can not be defined for the results obtained by the EWM-DFT , but the following relation

$$\Delta\omega = \frac{2\pi}{\Delta t\,(N+1)} \tag{31}$$

hold. Beside, to have a reliable computation a minimum of 3 elements per wave length is required which results in a $\omega_{max} = {}^{c2\pi}/_{3h}$. It must be remarked that this lower limit is very small and highly discussable but it is sufficient for the subsequent tests.





mesh with 1792 elements on 899 nodes
$h = 0.125\,\mathrm{m}$

Figure 7: System, boundary conditions, and mesh

h[m]	Elements	Nodes	N for CQM			for EWM-DFT	
			$\beta = 1.2$	$\beta = 0.6$	$\beta = 0.3$	$\omega_{max}[\mathrm{s}^{-1}]$	N_{max}
0.25	448	227	256	512	1024	43433	256 <
0.125	1792	899	512	1024	2048	86869	512 <
0.0625	7168	3587	1024	2048	4096	173734	1024 <

Table 2: Parameters for the different meshes (for ω_{max} three elements per wave length are set)

The column shown in Fig. 7 is discretised with 1792 triangular boundary elements of mesh size $h = 0.125\,\mathrm{m}$ on 899 nodes. Beside the displayed mesh a finer one (7168 elements) and a coarser one (448 elements) are used. In Tab. 2 the necessary parameters are listed. In principle the mesh size is halfed from one mesh to the other. Beside ω_{max} the amount of necessary time steps for the different β-values are given under the assumption of the same total time T. These values are needed for the comparison of the numerical effort in the next section. Further, in all calculations in the following studies the value $m = 3$ for the determination of the damping factor in (6) is chosen.

5.1 Comparison of results

For the above given problem the results are compared using a standard BEM without fast techniques. The traction solution in the middle of the fixed end is observed. As mentioned above, these data are approximated with constant shape functions, hence, the values can not be given at the exact mid point but on one of the elements which corner coincides with the mid point. For all BE calculations the same element is chosen and compared with the 1-d analytical solution (see [10]). The motivation to compare the traction solution is simply that the course of

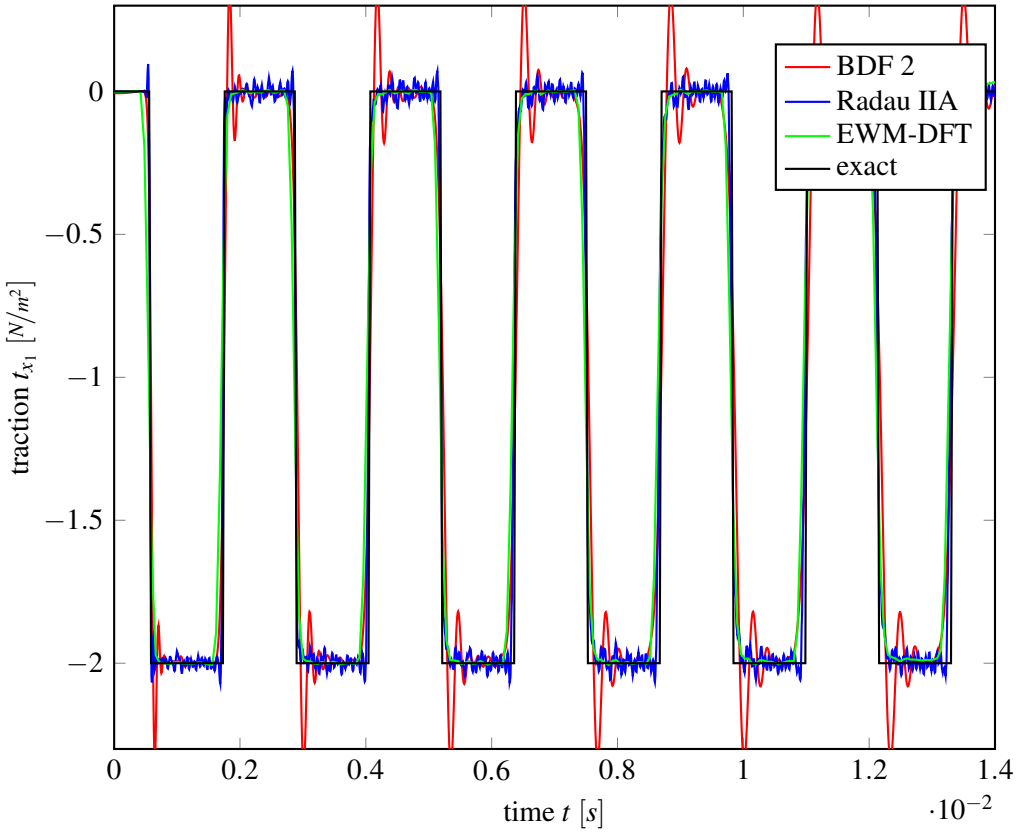


Figure 8: Traction solution t_{x_1} at the fixed end versus time: EWM-DFT and CQM solutions

this solution is more difficult to be approximated as it consists, essentially, of jumps. In Fig. 8, the solutions for the CQM (using BDF 2 and Radau IIA) with a $\beta = 0.3$ and the solutions for the EWM-DFT with 256 frequency steps, i.e., a $\omega_{max} = 52359\,\mathrm{s}^{-1}$, are displayed using the mesh with $h = 0.125\,\mathrm{m}$. The analytical solution is given as well and the only obvious conclusion is that all methods produce the correct solution. The BDF 2 solution shows the known overshooting at the jumps, whereas the Radau IIA method decreases them significantly.

In Tab. 3, the traction error computed with (19) is shown. The analytical 1-d solution has been taken as the exact solution. The necessary number of time steps is given in Tab. 2 for the different β-values of the different meshes. Additionally, the error of the displacements at the oposite point, i.e., at the free end, is given in Tab. 4. For EWM-DFT in the displacement results the value at the last time step has been canceled because it was obviously an overshoot. This shows the sensitivity of the method. Also indicated in the table, the result of the EWM-DFT for mesh 448 and $N = 512$ is not good, i.e., the error increases compared to that of the case with $N = 256$, whereas the same happens for mesh 1792 with $N = 1024$ (not shown in the tables). This reflects the fact that for the EWM-DFT an upper limit of computable frequencies ω_{max} exist for a given spatial discretization. In this problem, the upper limit seems to be consistent with the maximal frequency resulted from the requirement of 3 elements per wave length because in both

	BDF 2			Radau IIA			EWM-DFT		
h[m]	$\beta = 1.2$	$\beta = 0.6$	$\beta = 0.3$	$\beta = 1.2$	$\beta = 0.6$	$\beta = 0.3$	$N = 128$	$N = 256$	$N = 512$
0.25	0.311	0.274	0.214	0.251	0.222	0.206	0.289	0.232	0.838
0.125	0.288	0.227	0.176	0.180	0.147	0.139	0.266	0.192	0.163
0.0625	0.234	0.188	0.148	0.126	0.103	0.087	0.260	0.178	0.139

Table 3: Comparison of the error of the different methods: Traction solution

	BDF 2			Radau IIA			EWM-DFT		
h[m]	$\beta = 1.2$	$\beta = 0.6$	$\beta = 0.3$	$\beta = 1.2$	$\beta = 0.6$	$\beta = 0.3$	$N = 128$	$N = 256$	$N = 512$
0.25	0.081	0.056	0.039	0.095	0.064	0.049	0.150	0.084	0.128
0.125	0.057	0.030	0.017	0.043	0.026	0.018	0.139	0.067	0.035
0.0625	0.031	0.016	0.009	0.020	0.011	0.007	0.137	0.063	0.029

Table 4: Comparison of the error of the different methods: Displacement solution

discussed cases, $N = 256$ in case of mesh 448 and $N = 512$ in case of mesh 1792 exceeds the maximal frequency determined by the requirement of 3 elements per wave length. The CQM does not seem to be restricted by this requirement but the time step size should not be too small either, i.e., $\beta \gtrsim 0.1$. An inspection of the used frequencies show that the frequencies with the high imaginary part has a much larger real part than those of a comparable EWM-DFT . This seems to weight somehow the influence of the large frequencies such that the essential part of the frequency response is still included in the time domain solution without blowing it up. The price to pay is that more frequencies have to be used to get qualitatively the same result. This can be concluded from the above tables. If, e.g., an error of $\varepsilon \approx 0.18$ in the traction solution should be obtained the EWM-DFT would require approximately $N \approx 350$ and the CQM with BDF 2 $N \approx 2000$. With the Radau IIA method this value can be decreased to $N \approx 512$. This observation holds if the mesh with $h = 0.125\,\mathrm{m}$ is considered. However, if an error level of $\varepsilon \approx 0.139$ should be hold, the CQM with the Radau IIA needs the mesh with $h = 0.125\,\mathrm{m}$ and $N = 2048$ and the EWM-DFT needs the finer mesh with $h = 0.0625\,\mathrm{m}$ and $N = 512$. Having in mind that the complexity of a standard BEM, i.e., without acceleration, is quadratic in the degrees of freedom and linear in the amount of frequencies, the more efficient solution seems to be the CQM .

Summarizing, the EWM-DFT needs less frequencies to be computed compared to the CQM to obtain the same error level up to a certain error level. For small errors the overall efficiency of the CQM with Runge Kutta methods is better. Consequently, with a fixed mesh size the CQM can obtain a smaller error. However, it must be clearly stated that EWM-DFT requires a good adjustment of m and the usage of the Blackman window for non-smooth functions to justify these conclusions. In contrast, the determination of Δt for the CQM is straight forward given with $0.1 \lesssim \beta \lesssim 0.6$, whereas these limits can be in-/decreased, if fine meshes are used (see above the results for the finest mesh).

	EWM-DFT	CQM
No frequencies	201	450
displacement error	0.030	0.0052
traction error	0.157	0.120
CPU time	24.5 hours	14.1 hours
No iterations	25779	13608

Table 5: Comparison of the performance for EWM-DFT and CQM with 3-stage Radau IIA

5.2 Overall efficiency in combination with fast methods

In modern BE formulations fast methods are used to reduce complexity and storage to an almost linear order, which may lead to the conclusion that the overall efficiency of the EWM-DFT would be comparable to that of the CQM with Runge Kutta methods. However, most formulations require an iterative equation solver and the efficiency is, largely determined by the amount of iterations. It is well known that the iteration number grows with the increasing frequency. Hence, the performance of BE formulations based on transformation techniques is strongly influenced by the used frequencies. This motivates a study with a fast BEM comparing the EWM-DFT and the CQM .

Contrary to the above study, now an fixed error level is set to be $\varepsilon < 0.16$, the mesh with 448 quadratic elements is used, and only the CQM with Radau IIA (3-stage) is compared with the EWM-DFT . The total observation time is set to $T = 0.0155\,\mathrm{s}$ and the stopping criterion for the GMRES is set to $\varepsilon_{GMRES} = 10^{-4}$. In Tab. 5, the CPU time is compared besides the number of frequencies and iterations. The results show a clear advantage of the CQM formulation despite it needs more frequencies to be calculated. But the error of the CQM is obviously better and the iteration numbers are smaller. This results in a better overall efficiency for a fixed mesh size. The error of the EWM-DFT can be improved if for the higher frequencies a high frequency version of the FMM is used. But this would presumably also reduce the error in the CQM but not to the same extend. However, the bigger problem is the high iteration number of the equation solver. In Fig. 9, the number of these iterations are plotted versus the imaginary part of the frequency. The figure shows the dramatical increase of iteration numbers for the EWM-DFT , whereas the CQM stays nearly constant after an initial high value. The explanation can be found in the influence of the real part of the frequency. In Fig. 1, the used complex frequencies are plotted in the complex plane showing the constant real part of the EWM-DFT and the growing one for the CQM . The effect of the real part is to decrease the condition number of the matrices and, consequently, the iteration number. This effect has already been reported for the CQM in [17] and for the EWM-DFT in [23] in a study concerning an improved iterative equation solver. If a good pre-conditioner can be found for the EWM-DFT , this strong difference may be reduced. However, up to now such a pre-conditioner has not be presented in literature.

It may be argued that the example made here is not representative as it is not a large problem where an FMM is necessary. The used discretization results in 8064 degrees of freedom due to the non-conforming quadratic elements, which could also be solved with a normal BEM. Indeed, a more systematic study of larger problems with a robust fast method should be conducted. Nevertheless, the tendency will likely not change if the problem size is increased, especially the

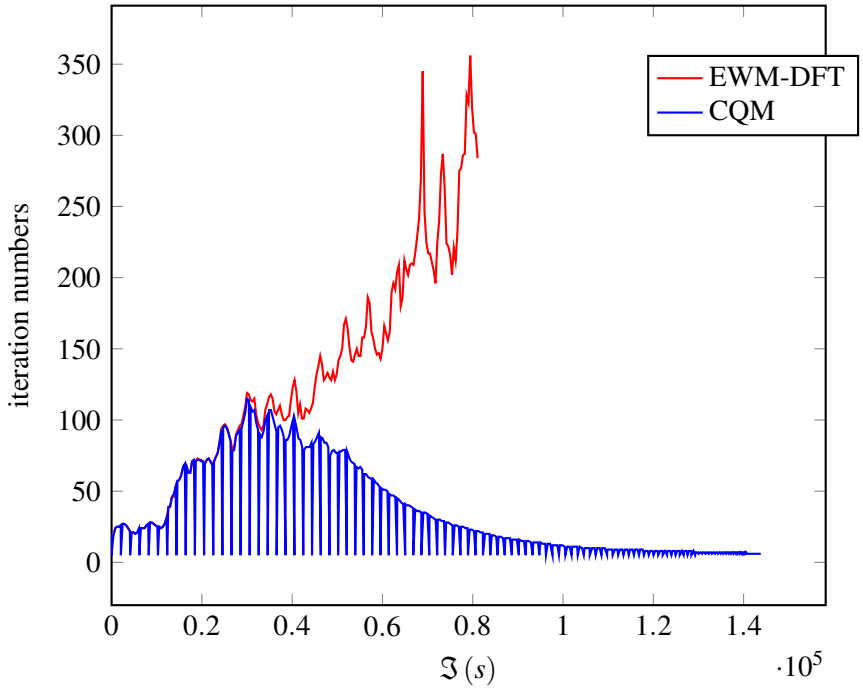


Figure 9: Iteration numbers versus the imaginary part of the complex frequency

large difference in the iteration number, which indicates that this is a systematical advantage of the CQM .

6 Conclusions

The CQM can be formulated as inverse Laplace transformation. It results in a simular formula as the inverse discrete Fourier transformation in combination with the exponential window method. In the paper, both formulations are recalled and the similarities are presented. Essentially, the same inverse transformation is made, however, the used complex frequencies are different. In the EWM-DFT approach a constant real part and continuously distributed imaginary parts of the complex frequency realises a straight integration path parallel to the imaginary axis in the complex plane. In the CQM the integration path is an ellipse in the complex plane. These differences determine the different numerical behavior of both methods. With the aid of a convolution integral of analytical functions related to the BEM for wave propagation the basic behavior has been studied. For the CQM the convergence order is mainly determined by the used time stepping method. A-stable Runge-Kutta methods give the best results. For the EWM-DFT the convergence behavior depends on the studied functions. For non-smooth functions, with a filtering technique, EWM-DFT can be as good as the CQM with high order Runge-Kutta methods. For smooth functions, the convergence rate of the EWM-DFT is slightly better.

Both methods can be used in the framework of BEM to obtain time domain results. Here, the problem under study is elastodynamics but the results can certainly be transfered to other

problem classes. The numerical study shows that the EWM-DFT can be very efficient compared to the CQM as long as not to high precessions, i.e., small errors, are requested. Essentially, the EWM-DFT needs less complex frequencies to be calculated and, consequently, the EWM-DFT is faster. However, the Runge-Kutta based CQM allows much smaller error levels for a fixed mesh size. The picture changes if fast BE formulations are used. In this study a kernel independent fast multipole method has been used. For this formulation the Runge-Kutta CQM is more efficient than the EWM-DFT because much less iterations are needed in the equation solver. This is caused by the larger real part of the complex frequencies in the CQM compared to the EWM-DFT , which improves the condition of the resulting system matrix. However, it must be remarked that these conclusions are based on numerical experiments and not on a mathematical proof.

Acknowledgement The first author gratefully acknowledges the hospitality and support by The Hong Kong University of Science and Technology during his sabbatical leave. The second author acknowledges Hong Kong Research Grants Council for supporting this work through Competitive Earmarked Research Grant 621411. The third author acknowledges the supports from the National Science Foundations of China under Grants 11102154 and 11472217 and the Alexander von Humboldt Foundation (AvH) to support his fellowship research at the Chair of Structural Mechanics, University of Siegen, Germany.

A Butcher tableaus for the used Runge-Kutta methods

In the test two 3-stage Runge-Kutta methods have been used. The respective Butcher tableaus are:

- 3-stage Radau IIA

$\frac{4-\sqrt{6}}{10}$	$\frac{88-7\sqrt{6}}{360}$	$\frac{296-169\sqrt{6}}{1800}$	$\frac{-2+3\sqrt{6}}{225}$
$\frac{4+\sqrt{6}}{10}$	$\frac{296+169\sqrt{6}}{1800}$	$\frac{88+7\sqrt{6}}{360}$	$\frac{-2-3\sqrt{6}}{225}$
1	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$
	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$

- 3-stage Lobatto IIIC

0	$\frac{1}{6}$	$-\frac{1}{3}$	$\frac{1}{6}$
$\frac{1}{2}$	$\frac{1}{6}$	$\frac{5}{12}$	$-\frac{1}{12}$
1	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

References

[1] M. H. Aliabadi. *The Boundary Element Method: Applications in Solids and Structures*, volume 2. J. Wiley & Sons, 2002.

[2] L. Banjai and S. Sauter. Rapid solution of the wave equation in unbounded domains. *SIAM J. Numer. Anal.*, 47(1):227–249, 2008.

[3] L. Banjai, M. Messner, and M. Schanz. Runge-Kutta convolution quadrature for the boundary element method. *Comput. Methods Appl. Mech. Engrg.*, 245–246:90–101, 2012. doi: 10.1016/j.cma.2012.07.007. URL http://www.sciencedirect.com/science/article/pii/S0045782512002277.

[4] Yanchuang Cao, Junjie Rong, Lihua Wen, and Jinyou Xiao. A kernel-independent fast multipole BEM for large-scale elastodynamic analysis. *Eng. Comp.*, page (in press), 2015.

[5] J. Domínguez. *Boundary Elements in Dynamics*. Computational Mechanics Publication, Southampton, 1993.

[6] M. G. Duffy. Quadrature over a pyramid or cube of integrands with a singularity at a vertex. *SIAM J. Numer. Anal.*, 19(6):1260–1262, 1982.

[7] F. Durbin. Numerical inversion of Laplace transforms: an efficient improvement to Dubner and Abate's method. *The Computer Journal*, 17(4):371–376, 1974.

[8] F. García-Sánchez, Ch. Zhang, and A. Sáez. 2-d transient dynamic analysis of cracked piezoelectric solids by a time-domain BEM. *Comput. Methods Appl. Mech. Engrg.*, 197 (33-40):3108–3121, 2008.

[9] L. Gaul and M. Schanz. A comparative study of three boundary element approaches to calculate the transient response of viscoelastic solids with unbounded domains. *Comput. Methods Appl. Mech. Engrg.*, 179(1-2):111–123, 1999. doi: 10.1016/S0045-7825(99)00032-8.

[10] D. Graffi. Über den Reziprozitätssatz in der Dynamik elastischer Körper. *Ingenieur Archiv*, 22:45–46, 1954.

[11] E. Kausel. *Fundamental Solutions in Elastodynamics*. Cambridge University Press, 2006.

[12] E. Kausel and J. Roësset. Frequency domain analysis of undamped systems. *J. Engrg. Mech., ASCE*, 118(4):721–734, 1992. doi: 10.1061/(ASCE)0733-9399(1992)118:4(721). URL http://dx.doi.org/10.1061/(ASCE)0733-9399(1992)118:4(721).

[13] L. Kielhorn and M. Schanz. Convolution quadrature method based symmetric Galerkin boundary element method for 3-d elastodynamics. *Int. J. Numer. Methods. Engrg.*, 76(11): 1724–1746, 2008.

[14] C. Lubich. Convolution quadrature and discretized operational calculus. I. *Numer. Math.*, 52(2):129–145, 1988.

[15] C. Lubich. Convolution quadrature and discretized operational calculus. II. *Numer. Math.*, 52(4):413–425, 1988.

[16] W. J. Mansur. *A Time-Stepping Technique to Solve Wave Propagation Problems Using the Boundary Element Method*. Phd thesis, University of Southampton, 1983.

[17] M. Messner and M. Schanz. An accelerated symmetric time-domain boundary element formulation for elasticity. *Eng. Anal. Bound. Elem.*, 34(11):944–955, 2010. doi: 10.1016/j.enganabound.2010.06.007.

[18] G. V. Narayanan and D. E. Beskos. Numerical operational methods for time-dependent linear problems. *Int. J. Numer. Methods. Engrg.*, 18(12):1829–1854, 1982. doi: 10.1002/nme.1620181207.

[19] M. Schanz. *Wave Propagation in Viscoelastic and Poroelastic Continua: A Boundary Element Approach*, volume 2 of *Lecture Notes in Applied Mechanics*. Springer-Verlag, Berlin, Heidelberg, New York, 2001. doi: 10.1007/978-3-540-44575-3.

[20] M. Schanz. On a reformulated convolution quadrature based boundary element method. *CMES Comput. Model. Eng. Sci.*, 58(2):109–128, 2010. doi: 10.3970/cmes.2010.058.109.

[21] M. Schanz and H. Antes. Application of 'operational quadrature methods' in time domain boundary element methods. *Meccanica*, 32(3):179–186, 1997.

[22] L. T. Wheeler and E. Sternberg. Some theorems in classical elastodynamics. *Arch. Rational Mech. Anal.*, 31:51–90, 1968.

[23] J. Xiao, W. Ye, Y. Cai, and J. Zhang. Precorrected FFT accelerated BEM for large-scale transient elastodynamic analysis using frequency-domain approach. *Int. J. Numer. Methods. Engrg.*, 90(1):116–134, 2012. ISSN 1097-0207. doi: 10.1002/nme.3316. URL `http://dx.doi.org/10.1002/nme.3316`.

[24] Jinyou Xiao, Wenjing Ye, and Lihua Wen. Efficiency improvement of the frequency-domain BEM for rapid transient elastodynamic analysis. *Comput. Mech.*, 52(4): 903–912, 2013. ISSN 0178-7675. doi: 10.1007/s00466-013-0852-9. URL `http://dx.doi.org/10.1007/s00466-013-0852-9`.

[25] L. Ying, G. Biros, and D. Zorin. A kernel-independent adaptive fast multipole algorithm in two and three dimensions. *J. Comput. Phys.*, 196(2): 591–626, 2004. ISSN 0021-9991. doi: 10.1016/j.jcp.2003.11.021. URL `http://www.sciencedirect.com/science/article/pii/S0021999103006090`.

[26] Ch. Zhang. Transient elastodynamic antiplane crack analysis in anisotropic solids. *Internat. J. Solids Structures*, 37(42):6107–6130, 2000.