Potential evaluation in space-time BIE formulations of 2D wave equation problems. *

G. Monegato†, L. Scuderi‡

This is the authors' post-print version of an article published on *Journal of Computational and Applied Mathematics*, Volume 243, Number 1 (2013), pp. 60-79, DOI: 10.1016/j.cam.2012.10.032.§

Abstract

In this paper we examine the computation of the potential generated by space-time BIE representations associated with Dirichlet and Neumann problems for the 2D wave equation. In particular, we consider the efficient evaluation of the (convolution) time integral that appears in the potential representation. For this, we propose two simple quadrature rules which appear more efficient than the currently used ones. Both are of Gaussian type: the classical Gauss-Jacobi quadrature rule in the case of a Dirichlet problem, and the classical Gauss-Radau quadrature rule in the case of a Neumann problem. Both of them give very accurate results by using a few quadrature nodes, as long as the potential is evaluated at a point not very close to the boundary of the PDE problem domain. To deal with this latter case, we propose an alternative rule, which is defined by a proper combination of a Gauss-Legendre formula with one of product type, the latter having only 5 nodes.

The proposed quadrature formulas are compared with the second order BDF Lubich convolution quadrature formula and with two higher order Lubich formulas of Runge-Kutta type. An extensive numerical testing is presented. This shows that the new proposed approach is very competitive, from both the accuracy and the efficiency points of view.

KEY WORDS: wave equation; space-time boundary integral equations; potential evaluation; quadrature rules

*This work was supported by the Ministero dell'Istruzione, dell'Università e della Ricerca of Italy, under the research program PRIN09 - Boundary element methods for time dependent problems.

†Dipartimento di Scienze Matematiche, Politecnico di Torino, Italy. Email: giovanni.monegato@polito.it

‡Dipartimento di Scienze Matematiche, Politecnico di Torino, Italy. Email: letizia.scuderi@polito.it

1 Introduction

Many papers are devoted to the numerical solution of Dirichlet or Neumann wave problems by means of boundary element methods (BEM). Very recently in [**?**] and in [**?**] we have proposed a BEM to solve 2D (and also 3D) exterior problems for the scalar non homogeneous wave equation with a Dirichlet or Neumann condition and in general with non trivial data. For these problems, we have derived corresponding space-time boundary integral equations (BIE), for whose solution we have proposed a BEM. This is based on a second order Lubich convolution quadrature for the discretization of the time integral, coupled with a space collocation or Galerkin boundary element method. However, in our papers, as well as in those devoted to the numerical solution of space-time BIE, the main task is the efficient and accurate numerical solution of the boundary integral equation. The efficient computation of the original PDE solution by means of its single or double layer representation becomes secondary or it is not at all taken into consideration.

Therefore, in this paper we will focus our attention on the computation of the solution of the original 2D PDE problem, by starting from the numerical approach described in [**?**], [**?**]. We recall that the numerical evaluation of the space integral has been already examined, being analogous to that one encounters in the case of elliptic problems; thus, we will examine only the evaluation of the (convolution) time integral. Since this is a trivial matter in the 3D case, we will consider only the 2D one. In particular, for it we will propose two rules which appear to be more efficient than the Lubich's quadratures, which are currently the mostly used formulas. Both are of Gaussian type: the classical Gauss-Jacobi quadrature rule in the case of a Dirichlet problem, and the classical Gauss-Radau quadrature rule in the case of a Neumann problem. Both of them give very accurate results by using a few quadrature nodes, as long as the potential is evaluated at a point not too close to the boundary of the problem domain. To deal with this latter case, we propose an alternative rule, which is defined by a proper combination of a Gauss-Legendre formula with one of product type.

We compare the quadrature formulas we have proposed with the second order Lubich convolution quadrature formula used in the numerical solution of the BIE (which represents the most natural choice for a numerical approach and hence the widely used formula in the literature) and with two higher order Lubich formulas of Runge-Kutta type.

In Section 2 we state the PDE problems we aim to solve, and recall the single and double layer BIE representations of their solutions. In Sections 3, 4 we define the quadrature formulas we propose for the time convolution integrals involved in the above representations, and compared them with those derived by Lubich. Finally, in Section 5, first we solve the BIE formulations of two test problems, and then we evaluate the associated potentials by using the formulas presented in Sections 3, 4. The numerical testing we have performed shows that indeed our formulas are very competitive, from both the accuracy and efficiency points of view.

2 Potential representation of the solution of 2D Dirichlet and Neumann wave problems

Let $\Omega^i \subset \mathbb{R}^2$ be an open bounded domain with a sufficiently smooth boundary Γ; define $\Omega^e = \mathbb{R}^2 \backslash \bar{\Omega}^i$. We identify Ω with Ω^i, if the problem we are solving is the interior one, or with Ω^e in the case of the exterior problem.

We consider the following (interior or exterior) homogeneous problem for the classical wave equation:

$$\begin{cases} \Delta u(\mathbf{x},t) - u_{tt}(\mathbf{x},t) &= 0 \quad \text{in } \Omega \times (0,T) \\ u(\mathbf{x},0) &= 0 \quad \text{in } \Omega \\ u_t(\mathbf{x},0) &= 0 \quad \text{in } \Omega, \end{cases} \tag{1}$$

with a Dirichlet boundary condition

$$u(\mathbf{x},t) = g^D(\mathbf{x},t), \quad \text{in } \Gamma \times (0,T)$$

or a Neumann boundary condition

$$\frac{\partial u}{\partial \mathbf{n_x}}(\mathbf{x},t) = g^N(\mathbf{x},t), \quad \text{in } \Gamma \times (0,T)$$

where $\frac{\partial u}{\partial \mathbf{n_x}}$ denotes the normal derivative pointing outside the chosen domain Ω.

In this paper we assume that the boundary Γ of the domain Ω and the boundary data satisfy the regularity and compatibility properties which guarantee the existence and uniqueness of the (classical) solution of the problem in $C^2([0,T], C^2(\bar{\Omega}))$. Moreover, for simplicity, we assume that the equation is homogeneous as well as the initial conditions; the non homogeneous case can be treated similarly. Indeed, this latter case differs from the homogeneous one only for (at most) the three additional "volume" integral terms generated by the non homogeneous data. These are involved both in the known term of the BIE and in the representation of the potential; see [**?**], [**?**]. Since for their numerical computation some efficient formulas have already been proposed in [**?**], for the Dirichlet case, and in [**?**] for the Neumann case, here we will only consider the (full) homogeneous problem.

Let $G(\mathbf{x},t)$ denote the wave equation fundamental solution

$$G(\mathbf{x},t) = \frac{1}{2\pi} \frac{H(t - \|\mathbf{x}\|)}{\sqrt{t^2 - \|\mathbf{x}\|^2}}, \qquad \mathbf{x} \in \mathbb{R}^2 \tag{2}$$

$H(\cdot)$ being the well known Heaviside function. For the Dirichlet problem, in [**?**] (see eq. (2.7)) we have derived the following single-layer potential representation of u

$$u(\mathbf{x},t) = \int_\Gamma \int_0^t G(\mathbf{x}-\mathbf{y}, t-\tau) \varphi^D(\mathbf{y},\tau)\, d\tau\, d\Gamma_{\mathbf{y}} \quad \mathbf{x} \in \Omega, \tag{3}$$

where φ^D is solution of the following BIE

$$\int_0^t \int_\Gamma G(\mathbf{x}-\mathbf{y}, t-\tau) \varphi^D(\mathbf{y},\tau) \Gamma_{\mathbf{y}} d\tau = g^D(\mathbf{x},t), \quad \mathbf{x} \in \Gamma, \tag{4}$$

and $\varphi^D(\mathbf{y},\tau) := [\partial_n u(\mathbf{y},\tau)]$ denotes the normal derivative jump of u along Γ.

Analogously, for the Neumann problem in [**?**] (see Proposition 2.1 and equation (20)) we have obtained the following double-layer potential representations of u:

$$u(\mathbf{x},t) = \int_\Gamma \int_0^t \frac{\partial G}{\partial \mathbf{n_y}}(\mathbf{x}-\mathbf{y},t-\tau)\varphi^N(\mathbf{y},\tau)\,d\tau\,d\Gamma_\mathbf{y} \quad \mathbf{x}\in\Omega, \tag{5}$$

$$u(\mathbf{x},t) = \int_\Gamma \int_0^t \frac{\partial G}{\partial \mathbf{n_y}}(\mathbf{x}-\mathbf{y},t-\tau)\varphi^N(\mathbf{y},\tau)\,d\tau\,d\Gamma_\mathbf{y} - \frac{1}{2}\varphi^N(\mathbf{x},t), \quad \mathbf{x}\in\Gamma, \tag{6}$$

where $\varphi^N(\mathbf{y},\tau) := [u(\mathbf{y},\tau)]$ denotes the jump of u along Γ and is the solution of the hypersingular BIE

$$\int_0^t \oint\!\!\!\!\!\!= \int_\Gamma \frac{\partial^2 G}{\partial \mathbf{n_x}\partial \mathbf{n_y}}(\mathbf{x}-\mathbf{y},t-\tau)\varphi^N(\mathbf{y},\tau)\,d\Gamma_\mathbf{y}\,d\tau = g^N(\mathbf{x},t),\ \mathbf{x}\in\Gamma. \tag{7}$$

Here and in the following, the symbol $\oint\!\!\!\!\!\!=$ means that the integral is defined in the finite part sense (see [**?**]).

Let now assume that we have already solved the BIE we have associated with Problem (**??**), i.e., we have determined its unknown $\varphi^D(\mathbf{y},\tau)$ or $\varphi^N(\mathbf{y},\tau)$. Since there is an extensive literature on the numerical evaluation of the (space) integral defined on Γ, in the next sections we will concentrate our attention to the numerical evaluation of the time convolution integral. In particular, we will propose a new approach, which appears more efficient than those currently used.

3 Convolution integrals in 2D Dirichlet wave problems

In this section and the next one, we consider the following convolution integral:

$$\mathrm{I}(\varphi;t) := \int_0^t K(t-\tau)\varphi(\tau)\,d\tau, \quad t>0 \tag{8}$$

with the kernel $K(t)$ first defined by

$$K(t) = G(r,t) = \frac{1}{2\pi}\frac{H(t-r)}{\sqrt{t^2-r^2}} \tag{9}$$

and then by

$$K(t) = \frac{\partial G(r;t)}{\partial \mathbf{n_y}} \tag{10}$$

being $r = ||\mathbf{x}-\mathbf{y}||$ the distance between $\mathbf{x}$ and $\mathbf{y}$, and $H(t)$ the Heaviside step function. As described in Section **??**, one deals with integral (**??**) with $K(t)$ defined by (**??**) or (**??**) when solving (2D) Dirichlet or Neumann wave problems, respectively. In particular, integrals of the above type appear in the single or double-layer representation of the wave PDE solution (see (**??**), (**??**)) .

We first consider (**??**) with $K(t)$ defined by (**??**) and $t>r>0$, that is,

$$\mathrm{I}(\varphi;t) = \frac{1}{2\pi}\int_0^{t-r} \frac{\varphi(\tau)}{\sqrt{(t-\tau)^2-r^2}}. \tag{11}$$

Since in this section we will define two quadrature rules for the evaluation of this convolution integral, that we will then compared with those proposed by Lubich in [**?**], for the reader convenience we briefly describe these latter.

Let assume that we have to evaluate (**??**) at a given instant $t = T$, or in the interval $(0, T]$. In both cases, to construct Lubich rules we ought to partition this interval into N subintervals of equal length $\Delta_t = \mathrm{T}/N$, by means of the equidistant points $t_j = j\Delta_t$, $j = 0, ..., N$.

Among the possible Lubich discrete convolution formulas, we will consider three of them: one of (convergence) order 2, based on the 2-step BDF method for ODEs, and two of order 3 and 5, respectively, based on two Radau IIA Runge-Kutta methods.

The first one takes the following form:

$$BDF(\varphi; t_n) = \sum_{j=0}^{n} \omega_{n-j}(\Delta_t)\varphi(t_j), \quad n = 0, \ldots, N \tag{12}$$

with

$$\omega_m(\Delta_t) = \frac{1}{2\pi \mathrm{i}} \int_{|z|=\rho} \widehat{K}\left(\frac{\gamma(z)}{\Delta_t}\right) z^{-(m+1)}\, dz \tag{13}$$

where $\widehat{K}$ is the Laplace transform of K and ρ is the radius of a circle in the domain of analyticity of $\widehat{K}(\gamma(z)/\Delta_t)$. The function $\gamma(z) = 3/2 - 2z + 1/2z^2$ is a characteristic polynomial associated with the chosen 2-step BDF method.

When $K(t)$ is defined by (**??**), we have:

$$\widehat{K}(s) = \widehat{K}(r, s) = \frac{1}{2\pi} K_0(rs), \tag{14}$$

where K_0 denotes the modified Bessel function of the second kind of order 0.

All the $N + 1$ quadrature weights $\{\omega_n\}$ are simultaneously computed by means of the trapezoidal rule

$$\omega_n(\Delta_t) = \omega_n(\Delta_t; r) \approx \frac{\rho^{-n}}{L} \sum_{l=0}^{L-1} \widehat{K}\left(r, \frac{\gamma(\rho e^{\frac{l2\pi \mathrm{i}}{L}})}{\Delta_t}\right) e^{-\frac{nl2\pi \mathrm{i}}{L}}, \quad n = 0, \ldots, N \tag{15}$$

combined with the Fast Fourier Transform, with $O(N \log N)$ flops. In [**?**] Lubich has proved that if the values of $\widehat{K}$ are computed with a relative accuracy bounded by ε, then the choices $L = 2N$ and $\rho^N = \sqrt{\varepsilon}$ yield an error in (**??**) of size $O(\sqrt{\varepsilon})$.

This numerical approach is quite general, although it requires the use of complex arithmetic. Moreover, with the above choices of L and ρ, and using the double precision arithmetic, according to Lubich's results the coefficients ω_n are computed with a relative accuracy of order (at least) $1.0E-07$.

Remark 3.1 *Concerning the quadrature error behavior, in [**?**] Lubich has proved that for a function $\varphi \in C^4([0, \mathrm{T}])$ with $\varphi(0) = \varphi'(0) = \varphi''(0) = \varphi'''(0) = 0$, we have $|\mathrm{I}(\varphi; t_n) - BDF(\varphi; t_n)| = O(\Delta_t^2)$, uniformly for $0 \le t_n \le \mathrm{T}$. We notice however that to obtain this rate of convergence we need to require the vanishing at the origin of the derivatives of φ only up to order 2. Moreover, when this latter condition is not satisfied,*

to restore the convergence order $O(\Delta_t^2)$, it is sufficient to modify the quadrature formula (??) as follows:

$$\overline{BDF}(\varphi; t_n) = BDF(\varphi; t_n) + \sum_{j=0}^{2} w_{jn}\varphi(t_j), \tag{16}$$

where the coefficients w_{jn} are determined by requiring the formula to integrate exactly all polynomials of degree (at most) 2.

To prove the above statements, we only need to write the following Taylor expansion of $\varphi(\tau)$:

$$\varphi(\tau) = \varphi(0) + \varphi'(0)\tau + \frac{1}{2}\varphi''(0)\tau^2 + \frac{1}{6}\varphi'''(0)\tau^3 + \psi(\tau)$$

and apply estimates (i) and (ii) of Theorem 3.1 in [?].

Incidentally, we remark that Lubich formulas associated with BDF methods of order greater than 2 cannot be used to compute our integrals, because they are not A-stable. In this case, some s-stage Runge-Kutta methods, satisfying appropriate stability properties, have been used. Among these, we recall, for example, the Radau IIA methods, which have been considered in [?], [?] for the approximation of convolution integrals within the numerical solution of parabolic equations. More recently, they have been used in [?], [?] and [?] to solve space-time BIE formulations for wave propagation problems.

If we identify these Runge-Kutta methods with the Butcher tableau

$$\begin{array}{c|c} c & \mathcal{Q} \\ \hline & \\ & b^{\mathrm{T}} \end{array}$$

the associated Lubich convolution quadrature takes the following form:

$$RK_s(\varphi; t_n) = \Delta_t \sum_{\nu=1}^{s} \sum_{j=0}^{n-1} w_{n-1-j,\nu}(\Delta_t)\varphi(t_j + c_\nu \Delta_t) \tag{17}$$

Its weights $w_{m,1}, \ldots, w_{m,s}$ coincide with the corresponding elements of the last row of the $s \times s$ matrix W_m defined as follows:

$$W_m = W_m(\Delta_t, r) = \frac{1}{\Delta_t 2\pi \mathrm{i}} \int_{|z|=\rho} \widehat{K}\left(r, \frac{\Delta(z)}{\Delta_t}\right) z^{-(m+1)}\, dz, \quad m = 0, \ldots, N-1$$

where

$$\Delta(z) = \left(\mathcal{Q} + \frac{z}{1-z}\mathbf{1}b^{\mathrm{T}}\right)^{-1}$$

and $\mathbf{1} = (1, \ldots, 1)^{\mathrm{T}}$. Likely to the BDF methods, the above Cauchy integrals can be approximated by the trapezoidal rule

$$W_m \approx \frac{\rho^{-m}}{\Delta_t L} \sum_{l=0}^{L-1} \widehat{K}\left(r, \frac{\Delta(\rho e^{\frac{l2\pi\mathrm{i}}{L}})}{\Delta_t}\right) e^{-\frac{ml2\pi\mathrm{i}}{L}}, \quad m = 0, \ldots, N-1$$

and the choices $L = 2N$ and $\rho = \varepsilon^{1/N}$ guarantee that the error in $\Delta_t W_m$ is $O(\sqrt{\varepsilon})$ if the values of $\widehat{K}$ are computed with accuracy ε. Using the FFT, also these weights are simultaneously computed with $O(N \log N)$ flops.

In the sequel we will consider the (order 3) 2-stage and the (order 5) 3-stage Radau IIA, defined by the tableau:

$$\begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array} \qquad \text{and} \qquad \begin{array}{c|ccc} \frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ 1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array}$$

respectively.

To these two rules we can apply the convergence result proved in [**?**] (see Theorem 4.1). In the case of our 2D wave equation, this states that, having defined $p = 2s - 1$ and assuming $\varphi \in C^p([0,T])$ with $\varphi(0) = \varphi'(0) = \cdots = \varphi^{(s)}(0) = 0$, there exists a $\bar{\Delta} > 0$ such that for $0 < \Delta_t \leq \bar{\Delta}$ and $t \in [0,T]$ we have

$$|I(\varphi; t_n) - RK_s(\varphi; t_n)| = O(\Delta_t^{\alpha})$$

uniformly with respect to $t_n \in [0,T]$, where $\alpha = \min\{2s-1, s+3/2\}$. Notice that for the 3-stage (order 5) RK method we only have $O(\Delta_t^{9/2})$.

As for the BDF method case (see Remark **??**), when the required vanishing conditions at the origin are not satisfied, to restore the above rate of convergence it is sufficient to modify the convolution quadrature by adding s additional terms, whose coefficients are determined by requiring the new rule to be exact for all polynomials of degree up to $s-1$. Obviously, these coefficients will depend on n and r (see (**??**), (**??**)).

We further remark that in the numerical solution of a space-time BIE formulation of wave equation problems, in particular in the time discretization of the convolution integral, the Lubich quadrature formula which is largely employed is that corresponding to the 2-step BDF method. Only very recently the Lubich quadrature formula associated with proper Runge Kutta methods have been considered for wave propagation problems (see [**?**], [**?**], [**?**]). A possible drawback of these latter rules is that they requires the computation of the unknown function at intermediate points of the original uniform grid. However they show very good dissipation/dispersion properties.

In the computation of the potential integrals, both the BDF and the Radau IIA methods can be used. The latter methods are preferable because they have a higher convergence rate. We notice however that, if the BIE has been solved using the 2-step BDF Lubich discrete convolution, to compute the associated potential by means of a Radau IIA method one has to interpolate φ with respect to the time variable. This is not the case if the BIE has been solved using the same RK method.

Remark 3.2 *To apply Lubich's convolution quadrature, the time interval of interest* $[0,T]$ *ought to be subdivided into* N *equal parts of length* Δ_t*, with* Δ_t *sufficiently small to guarantee the required accuracy. This also when we are interested in computing the integral* $I(\varphi;t)$ *only for a very few values of* t*; for example, only at* $t=T$*. Moreover, if* r $(<t)$ *is close to* t*, one cannot take advantage of the property that the integrand function in (**??**) vanishes in* $(t-r,t)$*.*

*A consequence of the latter drawback is, for example, that when in (**??**) we replace the time integral by the chosen* n*-point discrete convolution formula, we have nevertheless to compute* n *space integrals.*

To speed up the potential evaluation, it is thus important to evaluate the time integral with the required accuracy, by using a rule with a low number of nodes. To this end, since $\mathrm{I}(\varphi;t)=0$ whenever $t\le r$, in the following, to evaluate $\mathrm{I}(\varphi;t)$ when $t>r$, we propose two alternative approaches to the above mentioned Lubich quadratures.

The first one is simply given by the following m-point Gauss-Jacobi (GJ_m) rule:

$$\begin{aligned}\mathrm{I}(\varphi;t) &= \frac{1}{2\pi}\int_0^{t-r}\frac{\varphi(\tau)}{\sqrt{t+r-\tau}}\frac{d\tau}{\sqrt{t-r-\tau}} \\ &= \frac{1}{2\pi}\int_{-1}^{1}\frac{\varphi\left(\frac{t-r}{2}(\xi+1)\right)}{\sqrt{\frac{t+3r}{t-r}-\xi}}\frac{d\xi}{\sqrt{1-\xi}} \\ &\approx \frac{1}{2\pi}\sum_{i=1}^{m}\lambda_{im}^{GJ}\frac{\varphi\left(\frac{t-r}{2}(\xi_{im}^{GJ}+1)\right)}{\sqrt{\frac{t+3r}{t-r}-\xi_{im}^{GJ}}}\end{aligned} \tag{18}$$

where λ_{im}^{GJ} and ξ_{im}^{GJ} are the weights and nodes of the GJ_m quadrature rule defined by the weight function $(1-\xi)^{-\frac{1}{2}}$. This is the best choice when an analytic expression of φ is known and r is away from zero. Indeed, the numerical tests reported below show that this GJ_m formula is more efficient than the Lubich formulas. The convergence behavior of the Gauss-Jacobi quadratures is well-known for several classes of integrand functions; see, for example, [**?**]. When an approximation of $\varphi(t)$ is known only at equidistant instants $\{t_j\}$, to apply the above GJ_m rule we need to interpolate first, by a not-a-knot cubic spline function, for example, the given known values. We recall that when $\varphi\in C^4[0,T]$, this interpolation introduces an additional error term of order $O(\Delta_t^4)$.

Unfortunately, when $r\approx 0$, i.e., when $\mathbf{x}$ is very close to the boundary of the problem domain, the accuracy of the Gaussian rule decreases. This is due to the behavior of the integrand function, which becomes nearly singular as $(t+3r)/(t-r)\approx 1$ (notice that $(t+3r)/(t-r)>1$). Actually, this phenomenon also appears whenever $t>>r$. In this case a different numerical approach is required.

To deal with the latter situation we could evaluate the above integral $I(\varphi;t)$ by means of the m-point Gaussian rule associated with the (positive) weight function

$$w(\xi)=\frac{1}{\sqrt{(1-\xi)(\frac{t+3r}{t-r}-\xi)}}$$

However to construct this rule we would need some special software; moreover, for each value of t and of r one has to determine the corresponding set of nodes and

weights. Thus, except for the case one has to evaluate the integral for a single value of t/r, this approach is highly expensive.

To avoid this inefficiencies, we propose to use a new rule of composite type. To construct it, first we split the interval of integration into two parts: $(0, t-R)$ and $(t-R, t-r)$, with $R : r < R < t$ properly chosen. Then, we apply a m-point Gauss-Legendre (GL_m) rule to the first subinterval, and a ν-point formula of product type (P_ν), with ν very small, to the second one, after having reduced also this latter integral to the reference interval $(-1, 1)$. For this second rule we have taken as abscissas the Chebyshev nodes of the first kind, that is,

$$\xi_{i\nu}^{GC} = \cos(2i-1)\frac{\pi}{2\nu}, \quad i = 1, \ldots, \nu.$$

Thus, in this case, we approximate $\mathrm{I}(\varphi; t)$ as follows:

$$\begin{aligned}
\mathrm{I}(\varphi; t) &= \frac{1}{2\pi}\left[\int_0^{t-R} + \int_{t-R}^{t-r}\right] \frac{\varphi(\tau)}{\sqrt{(t-\tau)^2 - r^2}}\, d\tau \\
&= \frac{1}{2\pi}\left[\int_{-1}^{1} \frac{\varphi\left(\frac{t-R}{2}(\xi+1)\right)}{\sqrt{(\frac{t+R-2r}{t-R} - \xi)(\frac{t+R+2r}{t-R} - \xi)}}\, d\xi + \int_{-1}^{1} \frac{\varphi(\frac{R-r}{2}(\xi+1) + t - R)}{\sqrt{(1-\xi)\left(\frac{R+3r}{R-r} - \xi\right)}}\, d\xi\right] \\
&\approx \frac{1}{2\pi}\left[\sum_{i=1}^{m} \lambda_{im}^{GL} \frac{\varphi\left(\frac{t-R}{2}(\xi_{im}^{GL}+1)\right)}{\sqrt{(\frac{t+R-2r}{t-R} - \xi_{im}^{GL})(\frac{t+R+2r}{t-R} - \xi_{im}^{GL})}}\right. \\
&\quad \left. + \sum_{i=1}^{\nu} v_{i\nu}\varphi\left(\frac{R-r}{2}(\xi_{i\nu}^{GC}+1) + t - R\right)\right].
\end{aligned} \tag{19}$$

The integer ν is generally chosen small (in our experimental tests, for example, we have taken $\nu = 5$). The corresponding coefficients $v_{i\nu}$ are then defined by requiring the formula to be exact whenever φ is a polynomial of degree $\leq \nu - 1$.

More precisely, if we assume that φ is known only at the (equidistant) points $\{t_n, n = 0, ..., N\}$ defined above, as in the case they are obtained by solving our BIE, then we choose $0 \leq t - R \equiv t_{n_0-1} < t_{n_0} < t - r < t_{n_0+1} \leq \mathrm{T}$ if $t - r > t_1$; otherwise, if $t - r \leq t_1$ we set $t - R = 0$ and apply only the P_ν rule. In our testing, since we are mainly interested in evaluating the potential associated with a BIE, in (**??**) we have chosen $\nu = 5$ and $x_{i\nu}$ coinciding with the above defined Chebyshev nodes. This choice of ν is justified by the property that the remainder term behavior of this rule, as $\Delta_t \to 0, r > 0$ being fixed, is $O(\Delta_t^5)$. Indeed, this is negligible with respect to that of the most accurate Lubich convolution quadrature RK_3, which is $O(\Delta_t^{9/2})$.

The coefficients $v_{i\nu}$ are defined by requiring the formula P_ν to be exact whenever $\varphi(\tau)$ is a polynomial of degree $\leq \nu - 1$. Setting $v := \frac{R+3r}{R-r}$, the (weighted) moments associated with the standard interval $(-1, 1)$, which are needed to determine the weights $v_{i\nu}$ by solving the associated 5×5 linear system, are given by the following expressions:

$$
\begin{aligned}
\mu_{0\nu}^D &= 2\log\frac{\sqrt{2}+\sqrt{1+v}}{\sqrt{v-1}} \\
\mu_{1\nu}^D &= \frac{1}{2}\left((1+v)\mu_{0\nu}^D - \sqrt{8(1+v)}\right) \\
\mu_{2\nu}^D &= \frac{1}{4}\left(\frac{3v^2+2v+3}{2}\mu_{0\nu}^D - \sqrt{2(1+v)}(3v+1)\right) \\
\mu_{3\nu}^D &= \frac{1}{24}\left(\frac{15v^3+9v^2+9v+15}{2}\mu_{0\nu}^D - \sqrt{2(1+v)}(15v^2+4v+13)\right) \\
\mu_{4\nu}^D &= \frac{1}{192}\left(\frac{3(35v^4+20v^3+18v^2+20v+35)}{2}\mu_{0\nu}^D\right. \\
&- \left.\sqrt{2(1+v)}(105v^3+25v^2+83v+43)\right)
\end{aligned}
\tag{20}
$$

The 2-norm condition number of the above linear system is ≈ 19.6.

In Table **??** we have reported, for some value of r, the absolute errors and the corresponding estimated orders of convergence (EOC), associated with the computation of integral (**??**), when $\varphi(\tau) = \sin(2\tau)^2\tau^2\exp(-\tau)$. The reference values are computed using the most accurate quadrature formula for each value of r, after having taken $N = N_{ref} = 1024$; these are: GJ_N for $r \geq 0.01$ and $GL_N + P_\nu$ otherwise. In this table, as well as in all the following ones, the symbol "$--$" means that the double precision accuracy has been achieved.

In all the following tables, the total number of nodes n_t, hence of function evaluations, required by the above defined time integration formulas BDF, RK_2, RK_3, GJ_N, $GLP_{N+\nu} = GL_N + P_\nu$, will be: $N+1, 2N, 3N, N, N+5$, respectively, where the values of N are defined in the first column.

In Table **??**, the behavior of the Lubich formulas does not depend significantly on r; on the contrary, that of the Gaussian rules varies significantly, as $r \to 0$. We also notice that for $r \geq 1$ the Gauss-Jacobi formula is the most accurate one; while, as r decreases towards zero, the GLP rule become more efficient than the RK_3 and GJ approaches (see Fig. 1).

Concerning the first formula, we have to remark that, even if the coefficient matrix W_m can be computed by the FFT algorithm with $O(N\log N)$ flops, it requires the evaluation of a matrix function and the use of the complex arithmetic. Finally, to achieve an accuracy greater than $1.E-08$ by means of the RK_s when $r = 0.01, 0.0001$ and N is large, we have to increase the number of the trapezoidal quadrature nodes, for the computation of the convolution coefficients, up to $L = 4N$ or $L = 6N$. Otherwise, as some of the following tables show, an accuracy barrier will soon appear.

We finally recall that, except for the BDF Lubich formula, in all the others formulas the function φ ought to be evaluated also at abscissas which are different from those of the chosen uniform partition of $[0, \mathrm{T}]$.

In Table **??** we have reported the absolute errors given by the computation of integral (**??**) with kernel (**??**), where $\varphi(\tau) = \sin(2\tau)^2\tau^2\exp(-\tau)$ is known only at κ equidistant points of the integration interval $[0, \mathrm{T}]$. Therefore, except for the BDF rule, to apply the other quadrature formulas, for simplicity we have approximated

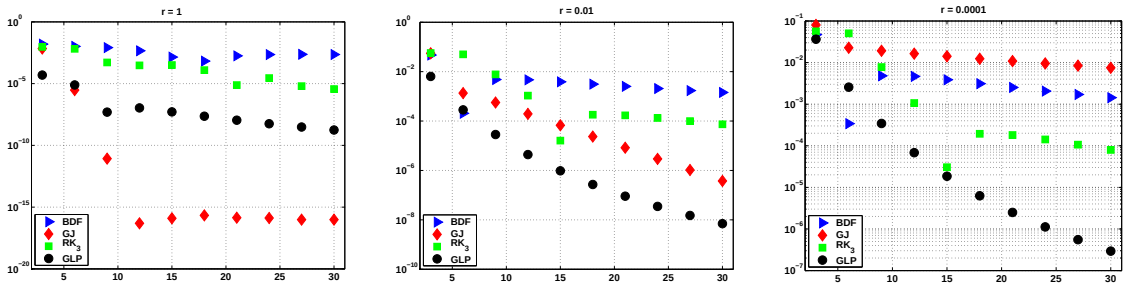
$\varphi(\tau)$ by the not-a-knot spline $S_3(\varphi)$ interpolating it at those κ points. We recall that this latter introduces an extra error term of order $O(\Delta_t^4)$. This choice has introduced the accuracy barrier shown by the results reported in the corresponding tables.

In Table **??**, we have fixed $r = 1$ and chosen $\kappa = 16, 32, 128$. The accuracy barrier generated by the spline interpolation appears very soon in the last three columns.

The reference values are computed by the GJ_N quadrature formula with $N = 1024$ and assuming $\varphi(\tau)$ known everywhere.

In Figure **??** we compare the performance of the approaches BDF, RK_3, GJ and GLP, when they are applied to integral (**??**) with kernel (**??**), $T = 3$ and $\varphi(\tau) = \sin(2\tau)^2\tau^2\exp(-\tau)$. For a fair comparison, the approaches use the same number of function evaluations. We have plotted the absolute errors obtained by taking the number of function evaluations reported on the x-axis. The reference values are those obtained by applying the GJ_{64} rule when $r = 1, 0.01$, and the GLP_{64+5} formula in the case $r = 0.0001$.

Figure 1: Absolute errors for (**??**) with (**??**) and $\varphi(\tau) = \sin(2\tau)^2\tau^2\exp(-\tau)$.



Finally, we have considered the case of a higher oscillating function: $\varphi(\tau) = \tau^3 e^{-\tau}\sin(100\tau)$. The performances of the chosen rules are reported in Tables **??**-**??**. From these, it emerges that in this case the BDF method is completely unsatisfactory, while the Gaussian approach appears very efficient.

Table 1: Absolute errors and EOCs for (**??**) with (**??**). $\varphi(\tau) = \sin(2\tau)^2\tau^2\exp(-\tau)$ and $t = \mathrm{T} = 3$.

N	BDF	EOC	RK_2	EOC	RK_3	EOC	GJ_N	EOC	GLP_{N+5}	EOC
					$r = 2$					
4	$1.19e-02$		$8.99e-03$		$3.79e-03$		$3.53e-06$		$8.67e-08$	
		$4.4e-1$		$1.1e+0$		$4.0e+0$		$2.6e+1$		
8	$1.26e-02$		$4.32e-03$		$2.33e-04$		$3.88e-14$		$6.71e-07$	
		$1.1e+0$		$2.4e+0$		$4.9e+0$		$1.2e+1$		$7.9e+0$
16	$9.28e-03$		$8.37e-04$		$8.00e-06$		$--$		$2.84e-09$	
		$2.0e+0$		$2.9e+0$		$5.0e+0$				$4.6e+0$
32	$4.30e-03$		$1.11e-04$		$2.55e-07$		$--$		$1.15e-10$	
		$2.2e+0$		$3.0e+0$		$5.0e+0$				$1.3e+1$
64	$1.07e-03$		$1.39e-05$		$7.93e-09$		$--$		$2.05e-14$	
		$2.1e+0$		$3.0e+0$		$5.0e+0$				
128	$2.32e-04$		$1.72e-06$		$2.47e-10$		$--$		$--$	
		$2.1e+0$		$3.0e+0$		$5.0e+0$				
256	$5.35e-05$		$2.14e-07$		$7.69e-12$		$--$		$--$	
		$2.1e+0$		$3.0e+0$		$5.1e+0$				
512	$1.29e-05$		$2.66e-08$		$2.30e-13$		$--$		$--$	
					$r = 1$					
4	$8.91e-03$		$5.58e-03$		$3.02e-04$		$9.77e-04$		$2.70e-05$	
		$4.0e+0$		$2.5e+0$		$3.4e+0$		$2.1e+1$		$8.0+0$
8	$9.26e-03$		$9.71e-04$		$2.81e-05$		$6.45e-10$		$1.08e-07$	
		$1.4e+0$		$2.7e+0$		$9.4e+0$		$2.6e+1$		$2.7e+0$
16	$5.98e-04$		$1.50e-04$		$4.27e-08$		$--$		$1.70e-08$	
		$1.8e+1$		$2.5e+0$		$2.1e+0$				$6.9e+0$
32	$2.18e-03$		$2.61e-05$		$9.94e-09$		$--$		$1.41e-10$	
		$1.9e+1$		$2.8e+0$		$4.3e+0$				$3.7e+0$
64	$8.41e-04$		$3.85e-06$		$5.03e-10$		$--$		$1.08e-11$	
		$2.0e+1$		$2.9e+0$		$4.8e+0$				$7.7e+0$
128	$2.36e-04$		$5.21e-07$		$1.82e-11$		$--$		$7.03e-14$	
		$1.9e+1$		$3.0e+0$		$2.4e+0$				
256	$6.17e-05$		$6.75e-08$		$3.48e-12$		$--$		$--$	
		$2.0e+1$		$3.0e+0$						
512	$1.57e-05$		$8.59e-09$		$7.71e-12$		$--$		$--$	
					$r = 0.01$					
4	$1.77e-02$		$2.56e-02$		$1.07e-03$		$9.44e-03$		$6.53e-04$	
		$2.1e+0$		$2.7e+0$		$3.0e+0$		$3.6e+0$		$3.5e+0$
8	$4.25e-03$		$4.07e-03$		$1.33e-04$		$8.01e-04$		$5.86e-05$	
		$2.4e-1$		$2.9e+0$		$3.1e+0$		$4.1e+0$		$6.6e+0$
16	$3.60e-03$		$5.44e-04$		$1.52e-05$		$4.75e-05$		$6.16e-07$	
		$1.5e-1$		$2.9e+0$		$4.0e+0$		$8.0e+0$		$7.1e+0$
32	$1.29e-03$		$7.41e-05$		$9.49e-07$		$1.89e-07$		$4.40e-09$	
		$1.8e+0$		$2.9e+0$		$5.0e+0$		$1.6e+1$		$7.2e+0$
64	$3.63e-04$		$1.0e-05$		$3.01e-08$		$3.93e-12$		$3.02e-11$	
		$1.9e+0$		$2.9e+0$		$5.3e+0$		$1.3e+1$		$7.0e+0$
128	$9.47e-05$		$1.32e-06$		$7.53e-10$		$6.66e-16$		$2.29e-13$	
		$2.0e+0$		$3.0e+0$		$5.0e+0$				$7.1e+0$
256	$2.41e-05$		$1.68e-07$		$2.37e-11$		$--$		$1.64e-15$	
		$2.0e+0$		$3.0e+0$						
512	$6.06e-06$		$2.10e-08$		$2.68e-10$		$--$		$--$	
					$r = 0.001$					
4	$1.75e-02$		$2.56e-02$		$1.07e-03$		$6.99e-04$		$2.57e-03$	
		$2.0e+0$		$2.6e+0$		$2.9e+0$				$3.1e+0$
8	$4.30e-03$		$4.11e-03$		$1.42e-04$		$8.33e-03$		$2.98e-04$	
		$2.6e-1$		$2.9e+0$		$3.1e+0$		$1.4e+0$		$5.9e+0$
16	$3.60e-03$		$5.54e-04$		$1.70e-05$		$3.22e-03$		$4.92e-06$	
		$1.5e+0$		$2.9e+0$		$3.7e+0$		$2.7e+0$		$6.3e+0$
32	$1.29e-03$		$7.68e-05$		$1.30e-06$		$5.04e-04$		$6.18e-08$	
		$1.8e+0$		$2.8e+0$		$3.9e+0$		$5.2e+0$		$6.4e+0$
64	$3.62e-04$		$1.07e-05$		$8.76e-08$		$1.37e-05$		$7.15e-10$	
		$1.9e+0$		$2.9e+0$		$4.0e+0$		$1.0e+1$		$6.5e+0$
128	$9.44e-05$		$1.47e-06$		$5.36e-09$		$1.32e-08$		$7.67e-12$	
		$2.0e+0$		$2.9e+0$		$4.0e+0$		$2.0e+1$		$6.7e+0$
256	$2.40e-05$		$2.00e-07$		$3.40e-10$		$1.47e-14$		$7.49e-14$	
		$2.0e+0$		$2.9e+0$		$6.6e-1$		$5.1e+0$		$5.3e+0$
512	$6.05e-06$		$2.64e-08$		$2.15e-10$		$--$		$1.92e-15$	

Table 2: Absolute errors for (**??**) with (**??**). $\varphi(\tau) = S_3(\sin(2\tau)^2\tau^2\exp(-\tau))$, $r = 1$, $t = \mathrm{T} = 3$.

N	BDF	RK_2	RK_3	GJ_N	GLP_{N+5}
			$\kappa = 16$		
4	$8.91e-03$	$5.58e-03$	$3.10e-04$	$9.76e-04$	$6.08e-05$
8	$9.26e-03$	$9.76e-04$	$2.38e-05$	$1.55e-05$	$1.55e-06$
16	$5.98e-04$	$1.55e-04$	$7.69e-06$	$8.91e-06$	$8.80e-06$
32	$2.17e-03$	$3.34e-05$	$7.09e-06$	$7.02e-06$	$7.21e-06$
64	$8.33e-04$	$1.08e-05$	$7.13e-06$	$7.07e-06$	$7.22e-06$
128	$2.29e-04$	$7.71e-06$	$7.06e-06$	$7.07e-06$	$7.21e-06$
256	$5.65e-05$	$7.15e-06$	$7.07e-06$	$7.07e-06$	$7.21e-06$
512	$8.96e-06$	$7.08e-06$	$7.07e-06$	$7.07e-06$	$7.21e-06$
			$\kappa = 32$		
4	$8.91e-03$	$5.58e-03$	$3.02e-04$	$9.76e-04$	$5.54e-04$
8	$9.26e-03$	$9.71e-04$	$2.66e-05$	$1.37e-06$	$2.82e-06$
16	$5.98e-04$	$1.52e-04$	$6.81e-07$	$6.94e-07$	$1.24e-06$
32	$2.18e-03$	$2.69e-05$	$7.86e-07$	$6.58e-07$	$4.45e-07$
64	$8.40e-04$	$4.63e-06$	$7.45e-07$	$7.24e-07$	$7.06e-07$
128	$2.36e-04$	$1.28e-06$	$7.15e-07$	$7.16e-07$	$7.15e-07$
256	$6.10e-05$	$7.89e-07$	$7.15e-07$	$7.15e-07$	$7.15e-07$
512	$1.49e-05$	$7.25e-07$	$7.15e-07$	$7.15e-07$	$7.15e-07$
			$\kappa = 128$		
4	$8.91e-03$	$5.58e-03$	$3.02e-04$	$9.77e-04$	$2.61e-03$
8	$9.26e-03$	$9.71e-04$	$2.81e-05$	$2.60e-09$	$2.09e-04$
16	$5.98e-04$	$1.50e-04$	$4.24e-08$	$2.45e-09$	$2.69e-06$
32	$2.18e-03$	$2.61e-05$	$1.41e-08$	$3.29e-09$	$3.46e-09$
64	$8.41e-04$	$3.86e-06$	$3.04e-09$	$2.64e-09$	$3.32e-09$
128	$2.36e-04$	$5.23e-07$	$2.37e-09$	$2.46e-09$	$1.48e-09$
256	$6.17e-05$	$6.99e-08$	$2.34e-09$	$2.32e-09$	$2.32e-09$
512	$1.57e-05$	$1.09e-08$	$2.33e-09$	$2.32e-09$	$2.33e-09$

Table 3: Absolute errors for (**??**) with (**??**). $\varphi(\tau) = \tau^3\exp(-\tau)\sin(100\tau)$, $r = 0.1$

N	BDF	RK_2	RK_3	GJ_N	GLP_{N+5}
4	$5.04e-02$	$4.49e-02$	$7.20e-04$	$9.53e-03$	$3.42e-02$
8	$3.82e-02$	$4.57e-02$	$2.94e-02$	$8.30e-03$	$3.58e-02$
16	$3.80e-02$	$3.24e-02$	$2.41e-02$	$6.85e-02$	$1.97e-02$
32	$3.84e-02$	$4.06e-02$	$1.85e-02$	$3.11e-02$	$3.72e-03$
64	$9.10e-04$	$3.37e-03$	$1.57e-03$	$5.16e-03$	$1.22e-03$
128	$2.53e-03$	$2.15e-03$	$1.96e-03$	$3.91e-03$	$2.99e-03$
256	$1.13e-03$	$1.41e-03$	$1.40e-04$	$4.27e-04$	$3.78e-04$
512	$1.13e-03$	$2.40e-04$	$1.33e-05$	$8.13e-15$	$1.03e-07$
1024	$9.21e-04$	$2.44e-05$	$5.27e-08$	$--$	$8.70e-09$
2048	$8.95e-04$	$7.02e-06$	$7.81e-09$	$--$	$1.08e-10$
4096	$4.61e-04$	$1.25e-06$	$1.35e-09$	$--$	$2.84e-13$
8192	$1.32e-04$	$1.82e-07$	$8.20e-09$	$--$	$7.09e-15$

Table 4: Numerical values of (**??**) with (**??**). $\varphi(\tau) = \tau^3\exp(-\tau)\sin(100\tau)$, $r = 0.1$

N	BDF	RK_3	GJ_N
4	$-4.9258391756840318e-02$	$1.8526314626703317e-03$	$1.0662933997184008e-02$
8	$-3.7099190421807995e-02$	$-2.8243170024519964e-02$	$-7.1704612941749304e-03$
16	$-3.6904459553958657e-02$	$2.5251039912798992e-02$	$6.9661962227335325e-02$
32	$-3.7221518426081811e-02$	$-1.7359999509674866e-02$	$3.2194085256150778e-02$
64	$2.2343306941026951e-04$	$-4.3226789790465228e-04$	$6.2922252904591743e-03$
128	$-1.3954519384163627e-03$	$-8.2490343517672917e-04$	$-2.7777376284759334e-03$
256	$-4.5438541903695603e-08$	$1.2734074801274515e-03$	$1.5600513773026828e-03$
512	$1.4883659860869328e-07$	$1.1464515972069253e-03$	$1.1331073596789311e-03$
1024	$2.1170594429162012e-04$	$1.1331600781978295e-03$	$1.1331073596808022e-03$
2048	$2.0283184016331932e-03$	$1.1330995502252515e-03$	$1.1331073596784051e-03$
4096	$1.5942520855116926e-03$	$1.1331060104438717e-03$	$1.1331073596798104e-03$
8192	$1.2650028711678699e-03$	$1.1330991614223684e-03$	$1.1331073596817470e-03$

Table 5: Absolute errors for (**??**) with (**??**). $\varphi(\tau) = \tau^3 \exp(-\tau)\sin(100\tau),\ r = 8$

N	BDF	RK_2	RK_3	GJ_N	GLP_{N+5}
4	$4.29e-02$	$1.27e-02$	$4.25e-02$	$3.23e-02$	$3.97e-02$
8	$4.21e-02$	$5.75e-03$	$4.09e-02$	$2.13e-02$	$3.97e-02$
16	$4.17e-02$	$4.04e-02$	$4.99e-02$	$5.41e-03$	$1.27e-02$
32	$4.17e-02$	$1.39e-02$	$6.40e-04$	$1.15e-02$	$1.30e-02$
64	$7.35e-03$	$7.35e-03$	$7.72e-03$	$6.53e-09$	$1.19e-02$
128	$7.35e-03$	$7.37e-03$	$6.34e-03$	$--$	$6.48e-04$
256	$7.35e-03$	$7.35e-03$	$7.34e-03$	$--$	$4.20e-06$
512	$7.35e-03$	$7.35e-03$	$7.06e-03$	$--$	$4.48e-07$
1024	$7.35e-03$	$7.35e-03$	$6.83e-04$	$--$	$9.70e-08$
2048	$7.35e-03$	$5.37e-03$	$2.27e-05$	$--$	$2.77e-07$
4096	$7.66e-03$	$1.11e-03$	$7.10e-07$	$--$	$3.55e-07$
8192	$9.71e-03$	$1.48e-04$	$2.21e-08$	$--$	$1.98e-07$

Table 6: Numerical values of (**??**) with (**??**). $\varphi(\tau) = \tau^3 \exp(-\tau)\sin(100\tau),\ r = 8$

N	BDF	RK_3	GJ_N
4	$-5.0207682444498476e-02$	$-4.9884484824954570e-02$	$-3.9625903758146203e-02$
8	$-4.9463238785970254e-02$	$-4.8281683452237115e-02$	$1.3962432086426143e-02$
16	$-4.9059329012229022e-02$	$4.2504560009072268e-02$	$-1.9400898916913959e-03$
32	$-4.9005449083994755e-02$	$-7.9908648153352805e-03$	$-1.8816738489447649e-02$
64	$1.2700021556968921e-08$	$3.7277474883842891e-04$	$-7.3510492267599634e-03$
128	$9.1185596740039898e-08$	$-1.0124958276289228e-03$	$-7.3510557577548315e-03$
256	$1.8603700469894938e-10$	$-1.2636054586908445e-05$	$-7.3510557577547968e-03$
512	$-4.5012234879075047e-11$	$-2.9151456150994140e-04$	$-7.3510557577545765e-03$
1024	$-4.5887367998180114e-11$	$-6.6679554174551821e-03$	$-7.3510557577547170e-03$
2048	$-4.6591937642576309e-11$	$-7.3283152131975796e-03$	$-7.3510557577544057e-03$
4096	$3.0903969312288476e-04$	$-7.3503456761484310e-03$	$-7.3510557577549365e-03$
8192	$2.3568928453271288e-03$	$-7.3510336523228527e-03$	$-7.3510557577543450e-03$

4 Convolution integrals in 2D Neumann wave problems

In this section we consider (**??**) with $K(t)$ given by (**??**). In this case the Lubich quadrature formula associated with the BDF or Radau IIA methods can be defined in a similar way; the only difference with respect to the Dirichlet case is the Laplace transform $\widehat{K}$, which now is given by

$$\widehat{K}(r,s) = -\frac{1}{2\pi} s K_1(rs) \frac{\partial r}{\partial \mathbf{n_y}}, \tag{21}$$

where K_1 denotes the modified Bessel function of the second kind of order 1.

For the BDF rule, assuming $\varphi \in C^5([0,T])$, with $\varphi(0) = \cdots = \varphi'''(0) = 0$, from [**?**] we obtain the following pointwise bound:

$$|\mathrm{I}(\varphi; t_n) - BDF(\varphi; t_n)| = O(\Delta_t^2)$$

uniformly for $0 \leq t_n \leq \mathrm{T}$.

For the two s-stage Runge-Kutta methods, we apply Theorem 5.1 in [**?**]. Thus, assuming $\varphi \in C^{s+2}([0,T])$, with $\varphi(0) = \varphi'(0) = \cdots = \varphi^{(s+1)}(0) = 0$, there exists a $\bar{\Delta} > 0$ such that for $0 < \Delta_t \leq \bar{\Delta}$ and $t_n \in [0,T]$ we have

$$\left(\Delta_t \sum_{n=0}^{N} |I(\varphi; t_n) - RK_s(\varphi; t_n)|^2 \right)^{1/2} = O(\Delta_t^{s+1/2})$$

Notice that for our two RK_2, RK_3 rules the above bound becomes $O(\Delta_t^{5/2})$ and $O(\Delta_t^{7/2})$, respectively. These two bounds are of lower order if compared with those we have in the Dirichlet case.

To apply our Gaussian formula, we need to use the following more explicit representation of $\mathrm{I}(\varphi; t)$, which is more suitable for its evaluation.

Proposition 4.1 *Let $r = ||\mathbf{x} - \mathbf{y}|| > 0$, $r < t$, be given, and assume $\varphi \in C^1[0,t]$. Then we we have:*

$$\mathrm{I}(\varphi; t) = \frac{r}{2\pi} \frac{\partial r}{\partial \mathbf{n_y}} \neq \int_0^{t-r} \frac{\varphi(\tau)}{[(t-\tau)^2 - r^2]^{3/2}} \, d\tau. \tag{22}$$

Proof. Recalling the relationship (see [**?**] Lemma 2.4)

$$\frac{\partial G}{\partial \mathbf{n_y}}(\|\mathbf{x} - \mathbf{y}\|, t) = -\frac{1}{2\pi} \left[\frac{\partial}{\partial t} \frac{H(t-r)}{\sqrt{t^2 - r^2}} + \frac{(t-r)H(t-r)}{(t^2 - r^2)^{3/2}} \right] \frac{\partial r}{\partial \mathbf{n_y}},$$

for the corresponding integral (**??**) we obtain the following representation:

$$\mathrm{I}(\varphi; t) = -\frac{1}{2\pi} \frac{\partial r}{\partial \mathbf{n_y}} \left[\frac{\partial}{\partial t} \int_0^{t-r} \frac{\varphi(\tau)}{\sqrt{(t-\tau)^2 - r^2}} \, d\tau + \int_0^{t-r} \frac{(t-\tau-r)\varphi(\tau)}{((t-\tau)^2 - r^2)^{3/2}} \, d\tau \right].$$

Then, we take $\epsilon > 0$ sufficiently small, and rewrite this expression as follows:

$$\mathrm{I}(\varphi; t) = -\frac{1}{2\pi} \frac{\partial r}{\partial \mathbf{n_y}} \lim_{\epsilon \to 0} \left[\frac{\partial}{\partial t} \int_0^{t-r-\epsilon} \frac{\varphi(\tau)}{\sqrt{(t-\tau)^2 - r^2}} \, d\tau + \int_0^{t-r-\epsilon} \frac{(t-\tau-r)\varphi(\tau)}{((t-\tau)^2 - r^2)^{3/2}} \, d\tau \right].$$

By computing explicitly the derivative of the first integral, and subsequently the limit value, as $\epsilon \to 0$, of the square bracket parenthesis, we obtain:

$$\mathrm{I}(\varphi;t) = \frac{r}{2\pi}\frac{\partial r}{\partial \mathbf{n_y}}\left[\int_0^{t-r} \frac{\varphi(\tau)-\varphi(t-r)}{((t-\tau)^2-r^2)^{3/2}}\,d\tau - \frac{t}{r^2}\frac{\varphi(t-r)}{\sqrt{t^2-r^2}}\right].$$

Finally, recalling the definition of finite part integral (see [**?**]), we obtain representation (**??**).□

Proceeding further, for the integral $\mathrm{I}(\varphi;t)$ defined in (**??**) we obtain

$$\begin{aligned}\mathrm{I}(\varphi;t) &= \frac{r}{2\pi}\frac{\partial r}{\partial \mathbf{n_y}} \,\text{⨍}_0^{t-r} \frac{1}{\sqrt{t-r-\tau}}\frac{\frac{\varphi(\tau)}{(t+r-\tau)^{\frac{3}{2}}}}{t-r-\tau}\,d\tau \\ &= \frac{2r}{\pi(t-r)^2}\frac{\partial r}{\partial \mathbf{n_y}} \,\text{⨍}_{-1}^{1} \frac{1}{\sqrt{1-\xi}}\frac{\frac{\varphi\left(\frac{(t-r)}{2}(\xi+1)\right)}{[(t+3r)/(t-r)-\xi]^{\frac{3}{2}}}}{1-\xi}\,d\xi \end{aligned} \tag{23}$$

We recall that in the above finite part integrals the change of variable is allowed, even if the hypersingularity is at an endpoint, because the order of the singularity is not an integer (see [**?**], p.13). When $r \geq 1$, to compute this finite part integral we use the Radau type Gaussian rule described in [**?**], that here takes the form:

$$\text{⨍}_{-1}^{1} \frac{1}{\sqrt{1-\xi}}\frac{\Phi(\xi)}{1-\xi}d\xi \approx a_0\Phi(1) + \sum_{i=1}^{m} \frac{\lambda_{im}^{GJ}}{1-\xi_{im}^{GJ}}\ \Phi(\xi_{im}^{GJ})$$

where

$$a_0 = -\sqrt{2} - \sum_{i=1}^{m} \frac{\lambda_{im}^{GJ}}{1-\xi_{im}^{GJ}} < 0.$$

For the convergence behavior of this rule see [**?**].

When $0 < r < 1$, proceeding as in the Dirichlet case, we propose the following numerical approach:

$$\begin{aligned}\mathrm{I}(\varphi;t) &= \frac{r}{2\pi}\frac{\partial r}{\partial \mathbf{n_y}}\left[\int_0^{t-R} + \text{⨍}_{t-R}^{t-r}\right] \frac{\varphi(\tau)}{[(t-\tau)^2-r^2]^{\frac{3}{2}}}\,d\tau \\ &= \frac{2r}{\pi}\frac{\partial r}{\partial \mathbf{n_y}}\left[\frac{1}{(t-R)^2}\int_{-1}^{1} \frac{\varphi\left(\frac{(t-R)}{2}(\xi+1)\right)}{\left[(\frac{t+R-2r}{t-R}-\xi)(\frac{t+R+2r}{t-R}-\xi)\right]^{\frac{3}{2}}}\,d\xi\right. \\ &\quad + \frac{1}{(R-r)^2}\left(\int_{-1}^{1} \frac{\varphi(\frac{R-r}{2}(\xi+1)+t-R)-\varphi(t-R)}{1-\xi}\frac{1}{\sqrt{1-\xi}(\frac{R+3r}{R-r}-\xi)^{\frac{3}{2}}}\,d\xi\right. \\ &\quad \left.\left. + \varphi(t-R)\,\text{⨍}_{-1}^{1} \frac{1}{\left[(1-\xi)(\frac{R+3r}{R-r}-\xi)\right]^{\frac{3}{2}}}\,d\xi\right)\right]\end{aligned}$$

Then, by applying the Gauss-Legendre rule to the first integral, and the corresponding product type rule, based on the Chebyshev nodes, to the second one, we obtain the following composite integration rule:

$$\mathrm{I}(\varphi;t) \approx \frac{2r}{\pi}\frac{\partial r}{\partial \mathbf{n_y}}\left[\frac{1}{(t-R)^2}\sum_{i=1}^{m}\lambda_{im}^{GL}\frac{\varphi\left(\frac{t-R}{2}(\xi_{im}^{GL}+1)\right)}{\left[(\frac{t+R-2r}{t-R}-\xi_{im}^{GL})(\frac{t+R+2r}{t-R}-\xi_{im}^{GL})\right]^{\frac{3}{2}}}\,d\xi\right.$$
$$\left.+\frac{1}{(R-r)^2}\left(\sum_{i=1}^{\nu}\tilde{v}_{i\nu}\frac{\varphi(\frac{R-r}{2}(\xi_{i\nu}^{GC}+1)+t-R)}{1-\xi_{i\nu}^{GC}}+\tilde{v}_{0\nu}\varphi(t-R)\right)\right]$$

with

$$\tilde{v}_{0\nu} = -\frac{R}{4r^2}\frac{(R-r)^{\frac{3}{2}}}{(R+r)^{\frac{1}{2}}} - \sum_{i=1}^{\nu}\tilde{v}_{i\nu}\frac{1}{1-\xi_{i\nu}^{GC}}.$$

As in the Dirichlet case, the corresponding coefficients $\widetilde{v}_{i\nu}$ are defined by requiring the formula to be exact for all polynomials of degree $\leq \nu-1$. To this end, setting $v := \frac{R+3r}{R-r}$, for $\nu = 5$ the required moments are given by the following expressions:

$$\begin{aligned}
\mu_{0\nu}^{N} &= \frac{\sqrt{8(1+v)}}{v^2-1}\\
\mu_{1\nu}^{N} &= v\mu_{0\nu}^{N} - \mu_{0\nu}^{D}\\
\mu_{2\nu}^{N} &= \frac{3v^2-1}{2}\mu_{0\nu}^{N} - \frac{3v+1}{2}\mu_{0\nu}^{D}\\
\mu_{3\nu}^{N} &= \frac{15v^3+v^2-7v-1}{8}\mu_{0\nu}^{N} - \frac{15v^2+6v+3}{8}\mu_{0\nu}^{D}\\
\mu_{4\nu}^{N} &= \frac{105v^4+10v^3-44v^2-10v-13}{48}\mu_{0\nu}^{N} - \frac{35v^3+15v^2+9v+5}{16}\mu_{0\nu}^{D}
\end{aligned} \tag{24}$$

where $\mu_{0,\nu}^{D}$ is defined in (**??**).

In Table **??** we have reported, for some values of r, the relative errors and the corresponding EOC associated with the computation of integral (**??**) with kernel (**??**), where $\varphi(\tau) = \tau^5\exp(-\tau)$, $t = \mathrm{T} = 10$. For each value of r, the reference values are computed by using the most accurate quadrature formula, after having set $N = 1024$; this is: the GJ_N rule when $r \geq 1$, and the GLP_{N+5} one otherwise. As in the Dirichlet case, for the BDF and RK methods an accuracy barrier appears; this is due to the error generated by the trapezoidal rule we have used to compute their coefficients.

Table 7: Relative errors and EOCs for (**??**) with (**??**). $\varphi(\tau) = \tau^5 \exp(-\tau)$ and $t = \mathrm{T} = 10$.

N	BDF	EOC	RK_2	EOC	RK_3	EOC	GJ_N	EOC	GLP_{N+5}	EOC
					$r = 5$					
4	$3.34e-01$		$3.55e-03$		$2.90e-02$		$2.09e-05$		$1.43e-05$	
		$2.5e+0$				$5.4e+0$		$1.8e+1$		$4.2e+0$
8	$5.76e-02$		$2.42e-02$		$6.81e-04$		$8.32e-11$		$7.92e-07$	
				$3.0e+0$		$5.1e+0$		$8.5e+0$		$7.5e+0$
16	$1.14e-01$		$2.93e-03$		$2.00e-05$		$2.33e-13$		$4.48e-09$	
		$1.2e+0$		$3.0e+0$		$4.9e+0$				$7.7e+0$
32	$5.10e-02$		$3.60e-04$		$6.64e-07$		$2.27e-13$		$2.12e-11$	
		$2.2e+0$		$3.0e+0$		$5.0e+0$				$3.6e+0$
64	$1.10e-02$		$4.43e-05$		$2.13e-08$		$2.42e-13$		$1.76e-12$	
		$2.1e+0$		$3.0e+0$		$5.0e+0$				$2.7e+0$
128	$2.66e-03$		$5.48e-06$		$6.75e-10$		$2.86e-13$		$2.72e-13$	
		$2.0e+0$		$3.0e+0$		$5.0e+0$				
256	$6.56e-04$		$6.81e-07$		$2.13e-11$		$2.64e-13$		$1.50e-13$	
		$2.0e+0$		$3.0e+0$		$1.9e+0$				
512	$1.63e-04$		$8.48e-08$		$5.77e-12$		$2.04e-13$		$7.19e-14$	
					$r = 1$					
4	$3.67e-01$		$5.07e-02$		$5.79e-03$		$8.71e-04$		$3.02e-07$	
		$2.5e+0$		$3.7e+0$		$9.5e+0$		$9.7e+0$		
8	$6.59e-02$		$3.96e-03$		$7.88e-06$		$1.03e-06$		$3.27e-06$	
		$1.6e+0$		$3.0e+0$		$1.9e+0$		$2.0e+1$		$8.7e+0$
16	$2.23e-02$		$4.83e-04$		$2.10e-06$		$1.16e-12$		$7.86e-09$	
		$2.1e+0$		$3.0e+0$		$4.4e+0$				$1.0e+1$
32	$5.21e-03$		$5.86e-05$		$9.66e-08$		$5.43e-13$		$7.43e-12$	
		$2.0e+0$		$3.0e+0$		$4.9e+0$				$4.3e+0$
64	$1.27e-03$		$7.15e-06$		$3.33e-09$		$5.41e-13$		$3.88e-13$	
		$2.0e+0$		$3.0e+0$		$2.0e+0$				
128	$3.13e-04$		$8.79e-07$		$8.07e-10$		$5.99e-13$		$5.16e-13$	
		$2.0e+0$		$3.0e+0$						
256	$7.78e-05$		$1.09e-07$		$3.13e-09$		$4.82e-13$		$3.74e-13$	
		$2.0e+0$		$3.1e+0$						
512	$1.94e-05$		$1.26e-08$		$3.84e-09$		$3.93e-13$		$2.91e-13$	
					$r = 0.01$					
4	$1.50e-04$		$3.28e-05$		$9.65e-06$		$7.03e-01$		$1.75e-06$	
		$2.3e+0$		$6.1e+0$		$3.4e+0$		$6.3e-1$		$5.6e+0$
8	$3.17e-05$		$4.88e-07$		$9.21e-07$		$4.55e-01$		$3.61e-08$	
		$2.6e+0$				$2.3e+0$		$1.6e+0$		$5.1e+0$
16	$5.20e-06$		$1.76e-06$		$1.89e-07$		$1.51e-01$		$1.09e-09$	
		$2.3e+0$		$1.1e+0$		$2.4e+0$		$3.7e+0$		$5.6e+0$
32	$1.09e-06$		$8.54e-07$		$3.69e-08$		$1.13e-02$		$2.28e-11$	
		$2.1e+0$		$1.4e+0$		$2.5e+0$		$7.9e+0$		$6.2e+0$
64	$2.47e-07$		$3.24e-07$		$6.75e-09$		$4.79e-05$		$1.23e-12$	
		$2.1e+0$		$1.7e+0$		$2.3e+0$		$1.6e+1$		$3.1e+0$
128	$5.89e-08$		$1.03e-07$		$1.42e-09$		$6.84e-10$		$9.00e-13$	
		$2.1e+0$		$2.0e+0$		$1.7e+0$		$9.0e+0$		
256	$1.42e-08$		$2.64e-08$		$4.39e-10$		$4.29e-13$		$8.97e-13$	
		$2.0e+0$		$2.4e+0$						
512	$3.56e-09$		$5.05e-09$		$4.50e-10$		$1.26e-12$		$8.96e-13$	
					$r = 0.0001$					
4	$3.58e-08$		$7.93e-09$		$1.61e-09$		$9.70e-01$		$2.26e-08$	
		$2.7e+0$		$5.1e+0$		$4.0e+0$		$4.2e-2$		$5.3e+0$
8	$5.37e-09$		$2.29e-10$		$1.04e-10$		$9.42e-01$		$5.65e-10$	
		$2.6e+0$						$8.9e-2$		$4.7e+0$
16	$9.00e-10$		$2.08e-10$		$7.11e-11$		$8.86e-01$		$2.23e-11$	
		$2.9e+0$						$1.9e-1$		$5.0e+0$
32	$1.24e-10$		$2.98e-10$		$6.45e-11$		$7.75e-01$		$6.95e-13$	
		$7.2e-1$						$4.6e-1$		$5.1e+0$
64	$7.56e-11$		$1.82e-10$		$6.87e-11$		$5.64e-01$		$2.08e-14$	
		$1.5e+0$		$1.3e+0$				$1.2e+0$		$5.0e+0$
128	$2.65e-11$		$7.20e-11$		$1.33e-10$		$2.50e-01$		$--$	
								$2.9e+1$		
256	$4.79e-11$		$1.68e-10$		$2.34e-11$		$3.40e-02$		$--$	
		$1.9e+0$						$6.2e+0$		
512	$1.30e-11$		$8.92e-11$		$1.04e-10$		$4.52e-04$		$--$	

In Table **??** we have reported the relative errors given by the computation of integral (**??**) with kernel (**??**), assuming that $\varphi(\tau) = \tau^5 \exp(-\tau)$ is known only at κ equidistant points of the integration interval $[0, \mathrm{T}]$. Therefore, as in Table **??**, to apply our quadrature formulas we have approximated $\varphi(\tau)$ by the not-a-knot spline $S_3(\varphi)$ of order 3, interpolating it at κ abscissas. In Table **??**, we have fixed $r = 1$ and chosen $\kappa = 16, 32, 128$.

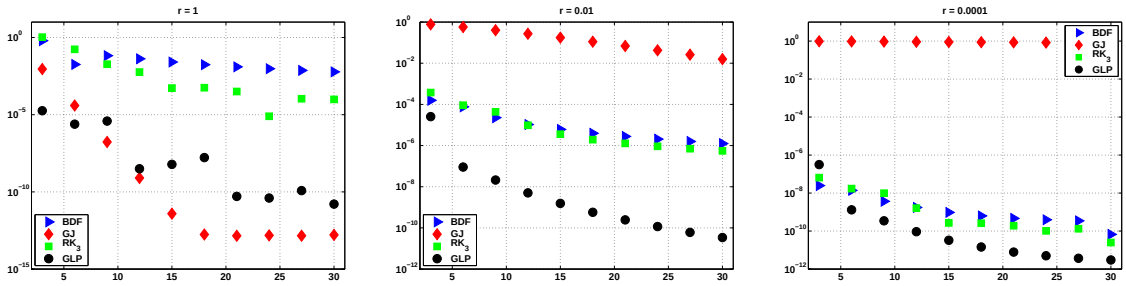
The reference values are computed by the GJ_N quadrature formula, with $N = 1024$ and assuming $\varphi(\tau)$ known everywhere.

Table 8: Relative errors for (**??**) with (**??**). $\varphi(\tau) = S_3(\tau^5 \exp(-\tau))$, $r = 1$, $t = \mathrm{T} = 10$.

N	BDF	RK_2	RK_3	GJ_N	GLP_{N+5}
			$\kappa = 16$		
4	$3.67e-01$	$5.07e-02$	$5.67e-03$	$7.36e-03$	$2.76e-04$
8	$6.59e-02$	$3.88e-03$	$6.78e-05$	$1.95e-04$	$2.57e-04$
16	$2.23e-02$	$4.87e-04$	$1.68e-04$	$1.13e-05$	$2.52e-04$
32	$5.21e-03$	$8.67e-05$	$2.48e-04$	$8.47e-06$	$2.55e-04$
64	$1.19e-03$	$2.23e-04$	$2.57e-04$	$1.17e-05$	$2.55e-04$
128	$1.80e-04$	$2.53e-04$	$2.55e-04$	$3.59e-05$	$2.55e-04$
256	$1.58e-04$	$2.55e-04$	$2.55e-04$	$7.82e-05$	$2.55e-04$
512	$2.38e-04$	$2.55e-04$	$2.55e-04$	$1.31e-04$	$2.55e-04$
			$\kappa = 32$		
4	$3.67e-01$	$5.07e-02$	$5.79e-03$	$7.36e-03$	$3.72e-06$
8	$6.59e-02$	$3.96e-03$	$1.01e-05$	$1.91e-04$	$2.00e-06$
16	$2.23e-02$	$4.82e-04$	$1.46e-06$	$2.04e-06$	$3.87e-06$
32	$5.21e-03$	$5.83e-05$	$1.69e-07$	$6.83e-07$	$4.20e-06$
64	$1.27e-03$	$7.22e-06$	$1.68e-06$	$1.14e-06$	$4.06e-06$
128	$3.13e-04$	$1.70e-06$	$4.02e-06$	$2.49e-06$	$4.07e-06$
256	$7.68e-05$	$3.67e-06$	$4.10e-06$	$4.16e-06$	$4.07e-06$
512	$1.85e-05$	$4.08e-06$	$4.07e-06$	$5.73e-06$	$4.07e-06$
			$\kappa = 128$		
4	$3.67e-01$	$5.07e-02$	$5.79e-03$	$7.36e-03$	$3.14e-07$
8	$6.59e-02$	$3.96e-03$	$7.88e-06$	$1.90e-04$	$3.26e-06$
16	$2.23e-02$	$4.83e-04$	$2.10e-06$	$1.83e-06$	$3.39e-08$
32	$5.21e-03$	$5.86e-05$	$9.34e-08$	$3.97e-08$	$3.37e-08$
64	$1.27e-03$	$7.14e-06$	$1.04e-09$	$3.61e-09$	$2.12e-08$
128	$3.13e-04$	$8.77e-07$	$1.28e-09$	$8.80e-09$	$2.50e-08$
256	$7.78e-05$	$1.07e-07$	$5.59e-10$	$1.18e-08$	$2.54e-08$
512	$1.94e-05$	$1.16e-08$	$1.47e-08$	$1.70e-08$	$2.54e-08$

As for the Dirichlet kernel, in Figure **??** we fairly compare the performance of the approaches BDF, RK_3, GJ and GLP for the evaluation of the integral (**??**) with kernel (**??**), $T = 10$ and $\varphi(\tau) = \tau^5 \exp(-\tau)$. We have plotted the relative errors obtained by the number of function evaluations reported on x-axis. The reference values are those obtained using the approaches GJ for $r = 1, 0.01$ and GLP when $r = 0.0001$.

Figure 2: Relative errors of (**??**) with (**??**) and $\varphi(\tau) = \tau^5 \exp(-\tau)$.



5 BEMs for wave problems and evaluation of the potential

In this section we apply the quadrature rules, we have examined in the previous sections, to compute the potentials defined by (**??**) and (**??**). In particular, to compare the performance of the approaches BDF, RK_3, GJ and GLP, we evaluate the potential given by (**??**) with kernel (**??**), when the density function $\varphi^D(\mathbf{x},\tau)$ has been obtained by solving the boundary integral equation (**??**) with $g^D(\mathbf{x},t) = t^4\exp(-2t)$. The boundary Γ coincides with that of the unit disc, parameterized by $(\cos(\theta),\sin(\theta))$, $\theta \in [-\pi,\pi)$.

To obtain an approximant of $\varphi^D(\mathbf{x},\tau)$, we have applied the collocation BEM described in [**?**], after subdividing the above parametrization interval into $M = 32$ equal subintervals, and considering the associated (space) continuous piecewise linear function. This is represented as a linear combination of the standard finite element Lagrangian basis $\{N_k(\theta), k = 1,\ldots,M\}$ of local degree 1, whose coefficients depend on τ.

Since in this case the density function is constant with respect to the space variable, this value of M causes an error which is negligible with respect to that due to the time integration. The matrix entries of the final linear system have been computed by applying the 32-point Gauss-Legendre rule to each boundary element.

To evaluate the space integral defined on the parametrization interval of the boundary Γ, for simplicity we have integrated each shape function N_k by applying a m-point Gauss-Legendre formula to each element of its support, with a computational cost of $2m$ function evaluations. The total cost due to the integration of our (space) approximant is thus $n_s = 2mM$.

As we have remarked in the previous sections, the approach GLP outperforms GJ, only when r is sufficiently small. Therefore, in the next numerical testing, since $r = ||\mathbf{x}-\mathbf{y}||$ varies when $\mathbf{y}$ moves on Γ, the GLP approach has to be reinterpreted as follows: we use the GLP rule only when $r < 0.1$, and the GJ one otherwise. This value of r has been chosen (both in the Dirichlet and Neumann case) after performing some experimental testing.

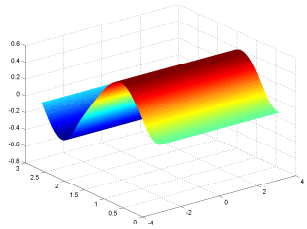
In Table **??** we have reported the (estimated) L^2-norm absolute errors we have obtained at each chosen instant t_n of the time interval $[0,T]$. As reference solution we have chosen the approximant obtained by taking $M = 32$, $N = 1024$; this is plotted in Figure **??**.

Table 9: Dirichlet wave problem. L^2-norm absolute errors for φ^D. $T = 3$

N	$T/4$	$T/2$	$3T/4$	T
16	$3.73E-02$	$1.46E-02$	$1.80E-01$	$3.03E-01$
32	$1.31E-02$	$8.06E-04$	$1.07E-01$	$1.38E-01$
64	$3.58E-03$	$4.71E-06$	$5.34E-02$	$2.29E-02$
128	$9.17E-04$	$2.76E-05$	$1.84E-02$	$5.43E-03$
256	$2.21E-04$	$1.08E-05$	$2.93E-03$	$1.25E-03$
512	$4.46E-05$	$2.60E-06$	$5.99E-04$	$2.47E-04$

Remark 5.1 *In the following testing, to compute the potential, for simplicity we have taken as density function φ^D the numerical solution obtained by taking $M = 32$ and*

Figure 3: Dirichlet wave problem. Density function φ^D.



$N = 1024$. *Since its relative accuracy at* $t = T/4, T/2, 3T/4, T$ *should be of order* $1.1E-5, 6.5E-7, 1.5E-4, 6.1E-5$, *respectively, in the next tables only the first 4-5 figures should be significant. Nevertheless, the* RK_3 *and* GJ_N *rules will converge, very fast the second one, to some values, whose figures beyond the above ones in general do not have any meaning. To show this phenomenon, we have preferred to report the potential values we have obtained, rather than their errors.*

Furthermore, to apply the latter two quadratures, we have interpolated the above (discrete) reference solution by the not-a-knot cubic spline defined by the subset of 128 equidistant time instants. This choice guarantees an interpolation relative accuracy less than 10^{-6}, *which is sufficient for the applications we will consider next.*

Table 10: Dirichlet wave problem. Numerical values of (**??**). $\mathbf{x} = (2, 0)$, $T = 3$, $n_s = 128$.

N	BDF	RK_3	GJ_N
		$T/4$	
4	$1.18833168138717e-02$	$-4.42966837794846e-04$	$0.00000000000000e+00$
8	$2.57002678946833e-03$	$3.13800269640450e-05$	
16	$3.66084839310169e-04$	$2.05742878950548e-06$	
32	$3.09059206744296e-05$	$-1.28683051234476e-08$	
64	$1.15160917485056e-06$	$-1.57582457279020e-11$	
128	$1.08735083693501e-08$	$-9.37427701556057e-16$	
256	$8.74634440323318e-12$	$2.52891342902737e-16$	
512	$3.11627226807381e-16$	$2.53625595617632e-16$	
1024	$2.43307376211427e-16$	$2.28497020769812e-16$	
		$T/2$	
4	$5.62631579984236e-02$	$1.82142277256695e-02$	$1.63792831658736e-02$
8	$3.53209481760269e-02$	$1.64244716166179e-02$	$1.63792573065788e-02$
16	$2.36258297329252e-02$	$1.63799702749361e-02$	$1.63792626260307e-02$
32	$1.83572261632710e-02$	$1.63786687268855e-02$	$1.63792623000755e-02$
64	$1.66882918352872e-02$	$1.63792756887661e-02$	$1.63792623845715e-02$
128	$1.64340271244956e-02$	$1.63792642276929e-02$	$1.63792624029582e-02$
256	$1.63902630024813e-02$	$1.63792653066198e-02$	$1.63792624027901e-02$
512	$1.63821523686572e-02$	$1.63792625394531e-02$	$1.63792624028210e-02$
1024	$1.63799128074817e-02$	$1.63792623886099e-02$	$1.63792624028213e-02$
		$3T/4$	
4	$1.21120234362169e-01$	$1.44279384749307e-01$	$1.44749763922269e-01$
8	$1.28782398658362e-01$	$1.44758074687492e-01$	$1.44740727691830e-01$
16	$1.36934829462159e-01$	$1.44740617394548e-01$	$1.44740731416758e-01$
32	$1.42822014943537e-01$	$1.44740105348607e-01$	$1.44740728076369e-01$
64	$1.44286632956065e-01$	$1.44740766182184e-01$	$1.44740728338813e-01$
128	$1.44629710418047e-01$	$1.44740753317944e-01$	$1.44740728169859e-01$
256	$1.44712524449859e-01$	$1.44740730994197e-01$	$1.44740728167727e-01$
512	$1.44733622306847e-01$	$1.44740728158353e-01$	$1.44740728167831e-01$
1024	$1.44739149888294e-01$	$1.44740728169527e-01$	$1.44740728167833e-01$
		T	
4	$1.59763485035794e-01$	$2.18037144506171e-01$	$2.16375950478717e-01$
8	$1.99922429181080e-01$	$2.16222691683145e-01$	$2.16185495038349e-01$
16	$2.15451803595472e-01$	$2.16177931826062e-01$	$2.16185314257246e-01$
32	$2.16745516181832e-01$	$2.16182942182543e-01$	$2.16185318955480e-01$
64	$2.16520512530719e-01$	$2.16184930061895e-01$	$2.16185318947306e-01$
128	$2.16314345978590e-01$	$2.16185261456589e-01$	$2.16185319157254e-01$
256	$2.16226711335958e-01$	$2.16185311897528e-01$	$2.16185319067697e-01$
512	$2.16197395699578e-01$	$2.16185317371127e-01$	$2.16185319072228e-01$
1024	$2.16188648587549e-01$	$2.16185318605181e-01$	$2.16185319072399e-01$

In Tables **??** - **??** we have reported the numerical values of the potential (**??**) at $\mathbf{x} = (2, 0)$, $(1.01, 0)$ and $(1.0001, 0)$, respectively, obtained at the intermediate instants $t_n = T/4, T/2, 3T/4$ and at $t_N = T = 3$. In this table, and in the following ones, n_s denotes the total number of nodes required by the quadrature rules we have used to compute the space integral. The chosen number of n_s is probably higher than that really needed; but for simplicity, we have applied the same Gaussian rule to each space element.

About the latter point, we remark that a point $\mathbf{x}$ can be very close at most to very few boundary elements. For example, for a convex domain, at most to two of them. Therefore, only the integration over these elements requires to use a Gauss-Legendre formula with a higher number of nodes. For the other elements, much fewer nodes are needed to achieve the same accuracy.

Table 11: Dirichlet wave problem. Numerical values of (**??**). $\mathbf{x} = (1.01, 0)$, $T = 3$, $n_s = 512$.

N	BDF	RK_3	GJ_N	GLP_{N+5}
		$T/4$		
4	$1.18340562288500e-01$	$6.84533592236506e-02$	$6.75745160310835e-02$	$6.79497148627066e-02$
8	$8.39331275818846e-02$	$6.79771041632198e-02$	$6.79403938828129e-02$	$6.79503719834775e-02$
16	$7.22746870410588e-02$	$6.79546119469360e-02$	$6.79537409626076e-02$	$6.79537732573932e-02$
32	$6.89952034706979e-02$	$6.79537709082315e-02$	$6.79537784953879e-02$	$6.79537781017959e-02$
64	$6.81972306614378e-02$	$6.79537789488834e-02$	$6.79537784839093e-02$	$6.79537785600856e-02$
128	$6.80118453763416e-02$	$6.79537783692746e-02$	$6.79537785014120e-02$	$6.79537785025887e-02$
256	$6.79679244287207e-02$	$6.79537775049628e-02$	$6.79537784996178e-02$	$6.79537784996308e-02$
512	$6.79572755451240e-02$	$6.79537781471645e-02$	$6.79537784996438e-02$	$6.79537784996450e-02$
1024	$6.79546562761094e-02$	$6.79537784316790e-02$	$6.79537784996431e-02$	$6.79537784996431e-02$
		$T/2$		
4	$2.45021178961290e-01$	$2.49048137205597e-01$	$2.48561359028582e-01$	$2.49262321448355e-01$
8	$2.46257542732829e-01$	$2.49294249696800e-01$	$2.49190219956435e-01$	$2.49245614767995e-01$
16	$2.47717192578355e-01$	$2.49246097970846e-01$	$2.49244492018470e-01$	$2.49245191691185e-01$
32	$2.48735649345189e-01$	$2.49245190826094e-01$	$2.49245185466403e-01$	$2.49245185698566e-01$
64	$2.49104975783849e-01$	$2.49245186498248e-01$	$2.49245186070236e-01$	$2.49245186094148e-01$
128	$2.49209424545216e-01$	$2.49245185072466e-01$	$2.49245185988381e-01$	$2.49245185987069e-01$
256	$2.49236238303980e-01$	$2.49245185019407e-01$	$2.49245185989947e-01$	$2.49245185989934e-01$
512	$2.49242962766508e-01$	$2.49245185594579e-01$	$2.49245185989244e-01$	$2.49245185989245e-01$
1024	$2.49244642345259e-01$	$2.49245185983706e-01$	$2.49245185989228e-01$	$2.49245185989228e-01$
		$3T/4$		
4	$2.09192590150734e-01$	$2.88559375812882e-01$	$2.86470351662232e-01$	$2.84470235829572e-01$
8	$2.61742287239026e-01$	$2.84561834808123e-01$	$2.84381021125066e-01$	$2.84173229686853e-01$
16	$2.78996566030938e-01$	$2.84186596795743e-01$	$2.84177578259405e-01$	$2.84174392610424e-01$
32	$2.82966505070642e-01$	$2.84173422554609e-01$	$2.84173668344773e-01$	$2.84173667879746e-01$
64	$2.83717135503490e-01$	$2.84173652650941e-01$	$2.84173655138096e-01$	$2.84173656799667e-01$
128	$2.83935419174377e-01$	$2.84173654349835e-01$	$2.84173656457059e-01$	$2.84173656346848e-01$
256	$2.84088228553954e-01$	$2.84173656599063e-01$	$2.84173656404341e-01$	$2.84173656395791e-01$
512	$2.84152077556819e-01$	$2.84173656445069e-01$	$2.84173656399093e-01$	$2.84173656399759e-01$
1024	$2.84168164249791e-01$	$2.84173656399354e-01$	$2.84173656399586e-01$	$2.84173656399598e-01$
		T		
4	$1.48067399761436e-01$	$2.01380160573841e-01$	$2.06555065645013e-01$	$2.04696756299882e-01$
8	$1.69201123862936e-01$	$2.01424097596908e-01$	$2.01383994471550e-01$	$2.01417024899041e-01$
16	$1.89182951834863e-01$	$2.01434445475681e-01$	$2.01430509072387e-01$	$2.01432028989638e-01$
32	$2.00039725942662e-01$	$2.01433354671908e-01$	$2.01433014603763e-01$	$2.01433107116383e-01$
64	$2.01554153026134e-01$	$2.01433200057443e-01$	$2.01433199228379e-01$	$2.01433203249936e-01$
128	$2.01487049534764e-01$	$2.01433199422968e-01$	$2.01433198063678e-01$	$2.01433197668107e-01$
256	$2.01449352882435e-01$	$2.01433199255094e-01$	$2.01433199238878e-01$	$2.01433199242600e-01$
512	$2.01437555708266e-01$	$2.01433199373647e-01$	$2.01433199279934e-01$	$2.01433199274538e-01$
1024	$2.01434333421078e-01$	$2.01433199308600e-01$	$2.01433199269347e-01$	$2.01433199269511e-01$

Finally, in Table **??** we have reported the number of seconds required by the listed approaches to compute the potential at $t_N = T$, using the same values of n_t and n_s of Table **??**.

We recall that the maximum relative accuracy we obtain when we solve the space-time BIE is 4-5 digits. Thus, when we compute the associated potential it does not make sense to require a higher accuracy. From Table **??** we notice that this accuracy is achieved by the BDF method when we take $N = 1024$, by the RK_3 rule with $N = 32$, and by the GJ_N rule for $N = 4, 8$. Thus in this case the last approach is about 22 times faster than the BDF rule, and about 100 times faster than the RK_3 rule. This ratios

Table 12: Dirichlet wave problem. Numerical values of (**??**). $\mathbf{x} = (1.0001, 0)$, $T = 3$, $n_s = 1280$.

N	BDF	RK_3	GJ_N	GLP_{N+5}
		$T/4$		
4	$1.21225771294076e-01$	$7.10679954748972e-02$	$6.93312388870119e-02$	$7.05593751957595e-02$
8	$8.66252655419394e-02$	$7.05896473607653e-02$	$7.02737171017611e-02$	$7.05601411369528e-02$
16	$7.49085521170552e-02$	$7.05666548267117e-02$	$7.05189517454947e-02$	$7.05657482140206e-02$
32	$7.16126619909731e-02$	$7.05657563450486e-02$	$7.05635635149598e-02$	$7.05657574774015e-02$
64	$7.08105325839569e-02$	$7.05657587122216e-02$	$7.05657509023407e-02$	$7.05657576259265e-02$
128	$7.06241468010138e-02$	$7.05657575937760e-02$	$7.05657576018889e-02$	$7.05657576089552e-02$
256	$7.05799831000380e-02$	$7.05657575936911e-02$	$7.05657576024445e-02$	$7.05657576025926e-02$
512	$7.05692745407019e-02$	$7.05657575914667e-02$	$7.05657576024402e-02$	$7.05657576024264e-02$
1024	$7.05666402909867e-02$	$7.05657575808773e-02$	$7.05657576024401e-02$	$7.05657576024402e-02$
		$T/2$		
4	$2.47778638804660e-01$	$2.51804248859010e-01$	$2.50572571706995e-01$	$2.52043556170846e-01$
8	$2.49075094535462e-01$	$2.52063702040176e-01$	$2.51628066033966e-01$	$2.52016313658419e-01$
16	$2.50514850723671e-01$	$2.52016471329877e-01$	$2.51934707124594e-01$	$2.52015546413738e-01$
32	$2.51514223620446e-01$	$2.52015540894367e-01$	$2.52007419597373e-01$	$2.52015535540877e-01$
64	$2.51877251299786e-01$	$2.52015536939503e-01$	$2.52015412285153e-01$	$2.52015535608276e-01$
128	$2.51980251571864e-01$	$2.52015535581528e-01$	$2.52015535540988e-01$	$2.52015535579294e-01$
256	$2.52006710590354e-01$	$2.52015535574818e-01$	$2.52015535579133e-01$	$2.52015535579242e-01$
512	$2.52013344324304e-01$	$2.52015535574769e-01$	$2.52015535578994e-01$	$2.52015535578991e-01$
1024	$2.52015000780787e-01$	$2.52015535587110e-01$	$2.52015535578994e-01$	$2.52015535578994e-01$
		$3T/4$		
4	$2.08426980039922e-01$	$2.89039649000014e-01$	$2.88150458440077e-01$	$2.85001655149622e-01$
8	$2.61534084998639e-01$	$2.85079327537979e-01$	$2.85499002871046e-01$	$2.84710326407848e-01$
16	$2.79131083900853e-01$	$2.84724752200223e-01$	$2.84884948866574e-01$	$2.84718289349363e-01$
32	$2.83290365518751e-01$	$2.84716879306908e-01$	$2.84739727950543e-01$	$2.84717009649342e-01$
64	$2.84152459833272e-01$	$2.84716995457180e-01$	$2.84717698141250e-01$	$2.84716998517297e-01$
128	$2.84439512809547e-01$	$2.84716997633633e-01$	$2.84716998713031e-01$	$2.84716997637774e-01$
256	$2.84624172253983e-01$	$2.84716997657224e-01$	$2.84716997705518e-01$	$2.84716997703896e-01$
512	$2.84693588160572e-01$	$2.84716997683170e-01$	$2.84716997706870e-01$	$2.84716997706722e-01$
1024	$2.84711068612504e-01$	$2.84716997706780e-01$	$2.84716997706947e-01$	$2.84716997706941e-01$
		T		
4	$1.47582486759578e-01$	$2.00843614849573e-01$	$2.06357888299412e-01$	$2.04097350208474e-01$
8	$1.68857459069513e-01$	$2.00775585860146e-01$	$2.00982469615980e-01$	$2.00767526402444e-01$
16	$1.88962328539564e-01$	$2.00787431411152e-01$	$2.00865109125368e-01$	$2.00784810765300e-01$
32	$1.99635766219673e-01$	$2.00786239306330e-01$	$2.00799714794961e-01$	$2.00785975561733e-01$
64	$2.00952441658070e-01$	$2.00786079432614e-01$	$2.00786725312432e-01$	$2.00786079025809e-01$
128	$2.00851031913936e-01$	$2.00786078697589e-01$	$2.00786080631752e-01$	$2.00786078695438e-01$
256	$2.00804945400590e-01$	$2.00786078828592e-01$	$2.00786078700909e-01$	$2.00786078699835e-01$
512	$2.00791106439779e-01$	$2.00786078754494e-01$	$2.00786078729716e-01$	$2.00786078731219e-01$
1024	$2.00787379423050e-01$	$2.00786078757343e-01$	$2.00786078727731e-01$	$2.00786078727711e-01$

remain the same when we compute the potential at several space points $\mathbf{x}$, for the same time instant t. In this case we have taken advantage of the MATLAB vectorial operation features.

For the last two rules, the CPU time includes the cubic spline interpolation, but not the a priori determination of the spline coefficients.

We ought however to remark that the BDF and RK rules simultaneously give, even if we are not interested in, and with a little overhead, an approximation of the potential at all the interior points of the chosen partition of the time interval $[0, T]$. Therefore, the computational cost for the potential evaluation at each one of these abscissas is very close to that needed to evaluate it at the final instant T. This is not the case of our approach, which requires the application of our rules at each chosen instant. Thus, when one needs to compute the potential at all the abscissas of the time interval partition, with no more than 4 significant digits, the BDF method in general appears superior.

Remark 5.2 *From the results reported in Tables* **??**-**??** *it emerges that the GJ approach is by far the most efficient one, when one has to evaluate the potential at a few time instants t, or at instants which do not coincide with the abscissas of the time interval partition defined by the BDF or RK methods. To compute the potential with 4-5 significant digits, which is the maximum accuracy we have achieved by applying our collocation boundary element method, the three rules BDF, RK_3 and GJ_N require 1024, 32 and 4-8 (time) abscissas, respectively.*

Table 13: CPU time required by (**??**) in Table **??**, $t = T$.

N	BDF	RK_3	GJ_N
4	$3.66e-02$	$1.87e-01$	$2.61e-02$
8	$2.03e-02$	$3.38e-01$	$1.35e-02$
16	$2.29e-02$	$6.63e-01$	$1.84e-02$
32	$2.89e-02$	$1.32e+00$	$2.81e-02$
64	$3.98e-02$	$2.63e+00$	$4.74e-02$
128	$5.94e-02$	$5.27e+00$	$8.66e-02$
256	$9.72e-02$	$1.05e+01$	$1.70e-01$
512	$1.67e-01$	$2.11e+01$	$3.41e-01$
1024	$3.01e-01$	$4.22e+01$	$7.21e-01$

The overall CPU time confirms the superiority of the GJ approach when one has to compute the potential for a few time instants, or when one requires a high accuracy. We recall that for the BDF and RK rules we have taken full advantage of the FFT algorithm, to minimize the computational cost.

The same testing we have performed above for the Dirichlet case, have been repeated for the corresponding Neumann problem with the data $g^N(\mathbf{x}, t) = t^4 \exp(-t)$. The corresponding tables and figures are reported below.

Table 14: Neumann wave problem. L^2-norm absolute errors for φ^N. $T = 3$

N	$T/4$	$T/2$	$3T/4$	T
16	$2.82E-02$	$1.29E-01$	$3.10E-01$	$4.70E-01$
32	$8.31E-03$	$3.44E-02$	$8.48E-02$	$1.15E-01$
64	$2.28E-03$	$8.89E-03$	$2.25E-02$	$2.84E-02$
128	$5.91E-04$	$2.24E-03$	$5.75E-03$	$6.99E-03$
256	$1.44E-04$	$5.38E-04$	$1.38E-03$	$1.66E-03$
512	$2.92E-05$	$1.08E-04$	$2.78E-04$	$3.32E-04$

Figure 4: Neumann wave problem. Density function φ^N.

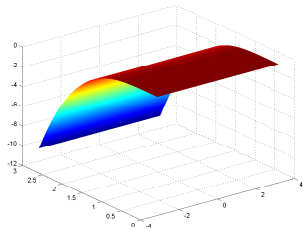


Table 15: Neumann wave problem. Numerical values of (**??**). $\mathbf{x} = (2, 0)$, $T = 3$, $n_s = 128$.

N	BDF	RK_3	GJ_N
		$T/4$	
4	$1.82524171757543e-03$	$2.51146256384322e-05$	$0.00000000000000e+00$
8	$3.57933714235021e-04$	$8.81934012850308e-06$	
16	$3.70443555200476e-05$	$2.79694804815003e-08$	
32	$2.01950528694472e-06$	$3.08655176383750e-10$	
64	$4.56197773548274e-08$	$5.68467370754284e-13$	
128	$2.50493949812220e-10$	$1.87332516284753e-16$	
256	$1.14055005962149e-13$	$2.13830911511849e-18$	
512	$7.90962778662871e-19$	$3.37358607396418e-18$	
1024	$1.79350863456579e-19$	$-3.83084783973906e-18$	
		$T/2$	
4	$3.64912145972709e-02$	$2.76343624673378e-03$	$2.81374986904464e-03$
8	$1.73626312314471e-02$	$2.81548970433493e-03$	$2.81374511598731e-03$
16	$8.27449535700557e-03$	$2.81375538104931e-03$	$2.81374583970067e-03$
32	$4.65892365478012e-03$	$2.81374800036567e-03$	$2.81374600104602e-03$
64	$3.35227851914943e-03$	$2.81374887913900e-03$	$2.81374598784438e-03$
128	$2.95458430438014e-03$	$2.81374885825422e-03$	$2.81374599198724e-03$
256	$2.84963264678060e-03$	$2.81374865518362e-03$	$2.81374599174361e-03$
512	$2.82279450334010e-03$	$2.81374682932586e-03$	$2.81374599174228e-03$
1024	$2.81602090992270e-03$	$2.81374589065857e-03$	$2.81374599174345e-03$
		$3T/4$	
4	$2.15868973889516e-01$	$1.40381842748111e-01$	$1.40346087400202e-01$
8	$1.71851135682669e-01$	$1.40354912993451e-01$	$1.40346163568479e-01$
16	$1.49019468417731e-01$	$1.40346445893027e-01$	$1.40346164143731e-01$
32	$1.42201214950169e-01$	$1.40346174558372e-01$	$1.40346163976448e-01$
64	$1.40762396414990e-01$	$1.40346164062650e-01$	$1.40346163980728e-01$
128	$1.40444028531596e-01$	$1.40346163689835e-01$	$1.40346163983584e-01$
256	$1.40369838509592e-01$	$1.40346163720306e-01$	$1.40346163984222e-01$
512	$1.40351989115818e-01$	$1.40346163897040e-01$	$1.40346163984221e-01$
1024	$1.40347607223095e-01$	$1.40346163978963e-01$	$1.40346163984241e-01$
		T	
4	$7.19408956089235e-01$	$7.58185822981593e-01$	$7.57573297379296e-01$
8	$7.25898773387120e-01$	$7.57619364041917e-01$	$7.57577177662955e-01$
16	$7.43481150589564e-01$	$7.57579886006520e-01$	$7.57577178513247e-01$
32	$7.53291756804713e-01$	$7.57577324010335e-01$	$7.57577178682857e-01$
64	$7.56410654707519e-01$	$7.57577189452832e-01$	$7.57577178652365e-01$
128	$7.57273581425075e-01$	$7.57577181064931e-01$	$7.57577178657306e-01$
256	$7.57499777703799e-01$	$7.57577179576298e-01$	$7.57577178659336e-01$
512	$7.57557640758120e-01$	$7.57577178586678e-01$	$7.57577178659384e-01$
1024	$7.57572271346404e-01$	$7.57577178590766e-01$	$7.57577178659526e-01$

Table 16: Neumann wave problem. Numerical values of (**??**). $\mathbf{x} = (1.01, 0)$, $T = 3$, $n_s = 2048$.

N	BDF	RK_5	GJ_N	GLP_{N+5}
		$T/4$		
4	$1.81765605119403e-02$	$2.26146923202208e-02$	$1.99791236987627e-02$	$2.24835061670171e-02$
8	$2.12761253111361e-02$	$2.24977562807619e-02$	$2.22980968128789e-02$	$2.24835713850215e-02$
16	$2.21022216361343e-02$	$2.24848893840553e-02$	$2.24825721741689e-02$	$2.24835313910462e-02$
32	$2.23613275449855e-02$	$2.24836503860784e-02$	$2.24835314022480e-02$	$2.24835313213386e-02$
64	$2.24475845846437e-02$	$2.24835390789837e-02$	$2.24835314320454e-02$	$2.24835314808325e-02$
128	$2.24736981441444e-02$	$2.24835317506027e-02$	$2.24835314322031e-02$	$2.24835314343194e-02$
256	$2.24809554993220e-02$	$2.24835313907102e-02$	$2.24835314321801e-02$	$2.24835314322528e-02$
512	$2.24828724242493e-02$	$2.24835314018172e-02$	$2.24835314321777e-02$	$2.24835314321786e-02$
1024	$2.24833652401958e-02$	$2.24835314004189e-02$	$2.24835314321847e-02$	$2.24835314321747e-02$
		$T/2$		
4	$3.14970608180231e-01$	$3.77924510980014e-01$	$2.83192135403240e-01$	$3.77875438518316e-01$
8	$3.57085299557244e-01$	$3.77871383797506e-01$	$3.62034470985597e-01$	$3.77885285018836e-01$
16	$3.71684346202619e-01$	$3.77884562018459e-01$	$3.77500118404123e-01$	$3.77885345167591e-01$
32	$3.76160976031825e-01$	$3.77885268643795e-01$	$3.77885115843469e-01$	$3.77885345318519e-01$
64	$3.77426625556584e-01$	$3.77885339988010e-01$	$3.77885345242249e-01$	$3.77885345208583e-01$
128	$3.77766688396442e-01$	$3.77885345012961e-01$	$3.77885345242186e-01$	$3.77885345240730e-01$
256	$3.77855146077644e-01$	$3.77885345268310e-01$	$3.77885345242138e-01$	$3.77885345242104e-01$
512	$3.77877725251999e-01$	$3.77885345260569e-01$	$3.77885345242119e-01$	$3.77885345242207e-01$
1024	$3.77883430486508e-01$	$3.77885345261650e-01$	$3.77885345242261e-01$	$3.77885345242336e-01$
		$3T/4$		
4	$1.25772235359496e+00$	$1.48496331616485e+00$	$9.08527207627862e-01$	$1.48476741183126e+00$
8	$1.40299706702341e+00$	$1.48484683088883e+00$	$1.34537791270693e+00$	$1.48482543662104e+00$
16	$1.45870471534046e+00$	$1.48482669800773e+00$	$1.47803064967290e+00$	$1.48482697833289e+00$
32	$1.47714990113442e+00$	$1.48482701729421e+00$	$1.48481124706051e+00$	$1.48482697583293e+00$
64	$1.48269810941539e+00$	$1.48482697761121e+00$	$1.48482697578443e+00$	$1.48482697589398e+00$
128	$1.48426514002039e+00$	$1.48482697485057e+00$	$1.48482697587123e+00$	$1.48482697587233e+00$
256	$1.48468442684712e+00$	$1.48482697470479e+00$	$1.48482697587121e+00$	$1.48482697587127e+00$
512	$1.48479117838046e+00$	$1.48482697533146e+00$	$1.48482697587103e+00$	$1.48482697587131e+00$
1024	$1.48481800439398e+00$	$1.48482697586756e+00$	$1.48482697587166e+00$	$1.48482697587131e+00$
		T		
4	$2.87074247407031e+00$	$3.24645090557322e+00$	$1.41356483448959e+00$	$3.24698838038621e+00$
8	$3.12673350486256e+00$	$3.24691985568484e+00$	$2.69403421034421e+00$	$3.24693237873894e+00$
16	$3.21654754427175e+00$	$3.24694508263825e+00$	$3.20587869979588e+00$	$3.24694706268537e+00$
32	$3.24062071098484e+00$	$3.24694692646341e+00$	$3.24673342720724e+00$	$3.24694706822997e+00$
64	$3.24553294751116e+00$	$3.24694706651371e+00$	$3.24694706260230e+00$	$3.24694706856163e+00$
128	$3.24660849318175e+00$	$3.24694706847472e+00$	$3.24694706857901e+00$	$3.24694706857743e+00$
256	$3.24686412844816e+00$	$3.24694706852180e+00$	$3.24694706857848e+00$	$3.24694706857781e+00$
512	$3.24692654890202e+00$	$3.24694706854749e+00$	$3.24694706857754e+00$	$3.24694706857680e+00$
1024	$3.24694197715456e+00$	$3.24694706857369e+00$	$3.24694706857974e+00$	$3.24694706857441e+00$

Table 17: CPU time required for (**??**) in Table **??**, $t = T$.

N	BDF	RK_3	GJ_N
4	$4.06e-02$	$1.87e-01$	$3.70e-02$
8	$1.96e-02$	$3.30e-01$	$1.80e-02$
16	$2.27e-02$	$6.47e-01$	$2.30e-02$
32	$2.94e-02$	$1.29e+00$	$3.34e-02$
64	$4.10e-02$	$2.57e+00$	$5.39e-02$
128	$6.23e-02$	$5.14e+00$	$9.50e-02$
256	$1.03e-01$	$1.03e+01$	$1.82e-01$
512	$1.80e-01$	$2.06e+01$	$3.63e-01$
1024	$3.27e-01$	$4.12e+01$	$7.57e-01$

All the numerical computation has been performed on a PC with two Intel Xeon® E5420 (2GHz) processors. We remark, however, that we have not considered the special features of our PC. To perform our numerical testing we have written standard (i.e., sequential) Matlab® codes.