



## FAST APPROXIMATION OF ALGEBRAIC AND LOGARITHMIC HYPERSINGULAR TYPE SINGULAR INTEGRALS WITH HIGHLY OSCILLATORY KERNEL

IDRISSA KAYIJUKA<sup>1,\*</sup>, ŞERIFE MÜGE EGE<sup>1</sup>, ALI KONURALP<sup>2</sup> AND FATMA SERAP TOPAL<sup>1</sup>

<sup>1</sup>*Department of Mathematics, Ege University, Izmir-Turkey*

<sup>2</sup>*Department of Mathematics, Manisa Celal Bayar University, Manisa-Turkey*

\* *Corresponding author: kayijukai@gmail.com*

ABSTRACT. Herein, highly oscillatory integrals with hypersingular type singularities are studied. After transforming the original integral into a sum of line integrals over a positive semi-infinite interval, a Gauss-related quadrature rule is constructed. The vehicle utilized is the moment's information. The comparison of two algorithms (Chebyshev and its modified one) to produce the recursion coefficients that satisfy orthogonal polynomial with respect to Gautschi logarithmic weight function, is investigated. Lastly, numerical examples are given to substantiate the effectiveness of the proposed method.

### 1. INTRODUCTION

The main aspect of our interest in this paper is the computational efficiency of hypersingular integral of the form

$$(1.1) \quad I^{[s]} [f; \mu] = \int_a^b \frac{(x-a)^\alpha (b-x)^\beta [\ln(x-a) \ln(b-x)]^s f(x) e^{i\omega x}}{(x-\mu)^k} dx, \quad (k \in \mathbb{N}),$$

---

Received July 20<sup>th</sup>, 2020; accepted August 10<sup>th</sup>, 2020; published September 23<sup>rd</sup>, 2020.

2010 *Mathematics Subject Classification.* 65D30, 65D32.

*Key words and phrases.* highly oscillatory kernel; Gauss quadrature; hypersingular integrals; Chebyshev and modified Chebyshev algorithms; algebraic and logarithm singular integrals.

©2020 Authors retain the copyrights of their papers, and all open access articles are distributed under the terms of the Creative Commons Attribution License.

where  $\alpha, \beta > -1, s \in \{0, 1\}, |\omega| \gg 1, a < \mu < b, -\infty < a < b < \infty$  and  $f$  is an analytic function of  $x$  in an appropriately large region containing the interval  $[a, b]$ . For simplicity let the above integral be written as

$$(1.2) \quad I^{[s]} [f; \mu] = \int_a^b \frac{\omega^{(\alpha, \beta, s)}(x) f(x) e^{i\omega x}}{(x - \mu)^k} dx,$$

where  $\omega^{(\alpha, \beta, s)}(x) = (x - a)^\alpha (b - x)^\beta [\ln(x - a) \ln(b - x)]^s$  is the weight function on  $[a, b]$ . While analyzing the above integral, we encounter the following cases:

1. When  $s \in \{0, 1\}$  and  $k = 0$  the integral becomes algebraic ( $s = 0$ ) or logarithmic ( $s = 1$ ) singular highly oscillatory integral,
2. When  $k = 1$  the integral is considered as the Cauchy Principal Value (CPV) integral with a highly oscillatory kernel,
3. When  $k = 2$  the integral is understood in the Hadamard Finite Part (HFP) sense integral with a highly oscillatory kernel,
4. When  $k \geq 2$  the integral is considered as Hypersingular, highly oscillatory integral.

Typically, it is somewhat difficult to evaluate CPV or Hypersingular as ordinary improper integrals due to the existence of a pole at  $x = \mu$ . If the order of singularity of the kernel function is higher than the dimension of integral, then the integral itself is called hypersingular. Having said that, hypersingular integrals play a vital role in physics and applied mechanics, where they can be utilized to formulate mixed boundary value problems. Besides these, we may find several applications of finite part integrals also in electrodynamics, aerodynamics, acoustics, and so on, for more about their applications, please see [1–5].

In an attempt to evaluate the Cauchy Hyperbolic problem for the differentiation of several variables, Hadamard [6, 7] introduced the concept of the finite-part in hypersingular integrals. Consequently, numerous approximations have been achieved, since then, to obtain the fastest algorithm for hypersingular integrals, particularly in the case where  $\omega^{(\alpha, \beta, s)} = 1$  and the frequency ( $\omega = 0$ ), for example, Gauss related types quadrature [8–13]. One may simply rewrite (1.1) in terms of the  $k$ th order derivative of the Cauchy principal value integral [11] as

$$(1.3) \quad I^{[s]} [f; \mu] = \frac{1}{(k-1)!} \frac{d^{k-1}}{d\mu^{k-1}} \int_a^b \frac{\omega^{(\alpha, \beta, s)}(x) f(x) e^{i\omega x}}{x - \mu} dx.$$

In the integral part of the above equality when  $\omega^{(\alpha, \beta, s)}(x) = 1$  and the frequency  $\omega = 0$  the integration part becomes the Hilbert transform and the sufficient condition for the existence of Hilbert transform is that  $f(x)$  has to satisfy a Lipschitz and Hölder condition in a closed interval  $[a, b]$ . In recent decades, many scientists have studied numerous computation methods and applications for numerical evaluation of hypersingular integral equations (for more details see [17–23] and the references therein). Since such types of integrals contain a broad range of applications, it is of great interest to provide a fast and accurate algorithm for the numerical computation of these types of integrals.

It is essential to mention that our primary goal here is to provide a fast algorithm for the efficient computation of the algebraic and logarithmic hypersingular type integrals with highly oscillatory kernel. Based on the assumption that  $f$  is a holomorphic function in an appropriately closed large region consisting  $[a, b]$ , the original integral is changed into several line integrals in a positive semi-infinite interval. Subsequently, these line integrals are computed using the construction of a Gauss related quadrature rule. Moreover, algorithms to produce the recursion coefficients of the three-term recurrence relation that satisfy an orthogonal polynomial with respect to Gautschi logarithmic weight function over positive half-infinite interval are compared by employing the moment's information approach.

This paper is structured as follows. In the next section, we evaluate the integral (1.1) using the proposed method. In section 3, we give the construction of the Gauss-related quadrature rule. Section 4 comprises numerical experiments to substantiate the efficiency of our proposed algorithms. Concluding remarks are provided in Section 5.

## 2. A METHOD FOR THE EVALUATION OF THE INTEGRAL (1.1)

In this section, we demonstrate a method based on contour integration in the complex plane for the integral (1.1). For  $k, s = 0$ , the application of the steepest descent or Clenshaw-Curtis methods are straightforward [24, 25]. However, for  $k \geq 0$  and  $s = 1$ , some modifications will be required inasmuch as some of the integrands will contain multi-valued functions that cannot be computed directly. Here, let  $z$  be defined as  $z = |z|e^{i(\theta+2n\pi)}$ , for  $n = 0, \pm 1, \dots$ , to get a single-valued function, we define our principal branch as  $\ln |z| + i\theta, 0 \leq \theta \leq 2\pi$ . Recall that the principal value of  $\ln z$  is the value obtained when  $n = 0$ . For instance,  $\ln(\pm i) = \ln 1 + (2n \pm \frac{1}{2})\pi i$ , which is equal to  $\pm \frac{\pi}{2}i$  for  $n = 0$ .

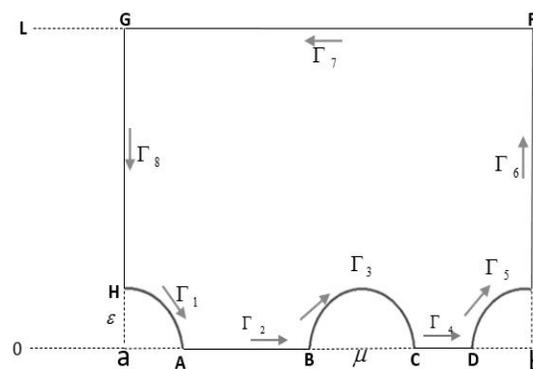


FIGURE 1. Integration path for hypersingular integral (1.1).

Let's define the region  $\mathfrak{R}$  bounded by  $\Gamma = \bigcup_{j=1}^8 \Gamma_j$ , such that

$$\begin{aligned}
 \Gamma_1 : \varphi_1(x) &= a + \varepsilon e^{ix}, 0 \leq x \leq \frac{\pi}{2}, & \Gamma_2 : \varphi_2(x) &= x, a + \varepsilon \leq x \leq \mu - \varepsilon, \\
 \Gamma_3 : \varphi_3(x) &= \mu + \varepsilon e^{ix}, 0 \leq x \leq \pi, & \Gamma_4 : \varphi_4(x) &= x, \mu + \varepsilon \leq x \leq b - \varepsilon, \\
 \Gamma_5 : \varphi_5(x) &= b + \varepsilon e^{ix}, \frac{\pi}{2} \leq x \leq \pi, & \Gamma_6 : \varphi_6(x) &= b + ix, \varepsilon \leq x \leq L, \\
 \Gamma_7 : \varphi_7(x) &= x + iL, a \leq x \leq b, & \Gamma_8 : \varphi_8(x) &= a + ix, \varepsilon \leq x \leq L.
 \end{aligned}
 \tag{2.1}$$

**Theorem 1** ([26], p. 114). Let  $f(z)$  be analytic in the region  $\mathfrak{R}$  and on its boundary  $\Gamma$ , then

$$\oint_{\Gamma} f(z) = 0.$$

The above theorem (Cauchy-Goursat), which is the focal theorem in this section, is used to prove the following theorem:

**Theorem 2.** Let  $f$  be analytic inside and on a simply-connected curve in a complex plane, briefly denoted by  $\mathfrak{R}$ , in Fig. 1. For a sufficiently large  $L$ , suppose that there exist two nonnegative constants  $M$  and  $\omega_o$  such that  $\int_{\Gamma} |f(x + iL)| dx \leq M e^{\omega_o L}$ , for  $\omega_o \in [0, \omega)$ . Then, the hypersingular oscillatory integral (1.1) can be transformed in the following form:

$$I^{[s]} [f; \mu] = I_a [f; \mu] + I_b [f; \mu] + \frac{i\pi}{(k-1)!} \frac{d^{k-1}}{d\mu^{k-1}} \left( \omega^{(\alpha, \beta, s)} (\mu) e^{i\omega\mu} f(\mu) \right), \quad k = 1, 2, \dots$$

where

$$I_a [f; \mu] = \left( \frac{i}{\omega} \right)^{1+\alpha} e^{i\omega a} \int_0^\infty \frac{(b - a - \frac{i}{\omega}y)^\beta \ln(\frac{i}{\omega}y) \ln(b - a - \frac{i}{\omega}y) f(a + \frac{i}{\omega}y) y^\alpha e^{-y}}{(a - \mu + \frac{i}{\mu}y)^k} dy$$

and

$$I_b [f] = \frac{(-i)^{1+\beta}}{\omega^{1+\beta}} e^{i\omega b} \int_0^\infty \frac{(b - a + \frac{i}{\omega}y)^\alpha \ln(-\frac{i}{\omega}y) \ln(b - a + \frac{i}{\omega}y) f(b + \frac{i}{\omega}y) y^\beta e^{-y}}{(b - \mu + \frac{i}{\omega}y)^k} dy.$$

**Proof:** Let's denote the integrand  $\phi(z)$  as

$$\phi(z) = \frac{\omega^{(\alpha, \beta, s)} (z) f(z) e^{i\omega z}}{(z - \mu)^k}, \quad k = 1, 2, \dots,$$

and let  $\mathfrak{R}$  be the region in the domain  $D = \{z \in \mathbb{C} : a \leq Re(z) \leq b, 0 \leq Im(z) \leq L\}$ , bounded by  $\Gamma$  as defined in (2.1). Since  $\phi(z)$  is analytic, by employing the Cauchy-Goursat integral theorem, we have

$$\int_{AB} \phi(x) dx + \int_{CD} \phi(x) dx = - \left( \int_{HA} + \int_{BC} + \int_{DE} + \int_{EF} + \int_{FG} + \int_{GH} \right) \phi(z) dz,$$

with the direction being taken in the positive sense (counterclockwise), as depicted in Fig. 1. We can show, with ease, that the integral over quarter circles HA and DE tend to zero as epsilon goes to zero. More

precisely, let  $\Gamma_1 : z = a + \varepsilon e^{ix}, x \in [0, \frac{\pi}{2}]$ , this yields

$$\begin{aligned} \left| \int_{\Gamma_1} \phi(z) dz \right| &= \left| - \int_0^{\frac{\pi}{2}} i \varepsilon^{1+\alpha} \frac{e^{ix(1+\alpha)} (b-a-\varepsilon e^{ix})^\beta \ln(\varepsilon e^{ix}) \ln(b-a-\varepsilon e^{ix}) f(a+\varepsilon e^{ix}) e^{i\omega(a+\varepsilon e^{ix})}}{(a-\mu+\varepsilon e^{ix})^k} dx \right| \\ &\leq \varepsilon^{1+\alpha} |\ln \varepsilon| \int_0^{\frac{\pi}{2}} |G(\varepsilon, x)| dx + \varepsilon^{1+\alpha} \int_0^{\frac{\pi}{2}} |x| |G(\varepsilon, x)| dx, \end{aligned}$$

where

$$G(\varepsilon, x) = \frac{(b-a-\varepsilon e^{ix})^\beta \ln(b-a-\varepsilon e^{ix}) f(a+\varepsilon e^{ix}) e^{i\omega(a+\varepsilon e^{ix})}}{(a-\mu+\varepsilon e^{ix})^k}.$$

It can be easily shown that the function  $G(t, \theta)$  is continuous on  $t \in [0, \varepsilon]$  and  $\theta \in [0, \frac{\pi}{2}]$ . Thus, taking the limit on both sides as  $\varepsilon \rightarrow 0$ , we get  $\left| \int_{\Gamma_1} \phi(z) dz \right| \rightarrow 0$ . The same technique can be applied to obtain  $\left| \int_{\Gamma_5} \phi(z) dz \right| \rightarrow 0$ , as  $\varepsilon \rightarrow 0$ . Moreover, by letting  $\Gamma_7 : z = x + iL, x \in [a, b]$ , it follows that

$$\begin{aligned} \left| \int_{\Gamma_7} \phi(z) dz \right| &= \left| -e^{-\omega L} \int_a^b \frac{(x-a+iL)^\alpha (b-x-iL)^\beta \ln(x-a+iL) \ln(b-x-iL) f(x+iL) e^{i\omega x}}{(x-\mu+iL)^k} dx \right| \\ &\leq e^{-\omega L} \int_a^b |\psi(x+iL)| |f(x+iL)| dx \\ &\leq MM_1 e^{-L(\omega-\omega_0-\omega_1)} (b-a) \rightarrow 0, \text{ as } L \rightarrow \infty, \end{aligned}$$

wherein the above expressions

$$\psi(x+iL) = \frac{(x-a+iL)^\alpha (b-x-iL)^\beta \ln(x-a+iL) \ln(b-x-iL)}{(x-\mu+iL)^k}.$$

Note that for any nonnegative constants  $M_1, \omega_1$  such that  $\omega > \omega_0 + \omega_1$  and for sufficiently large  $L$ , we can choose  $|\psi(x+iL)| \leq M_1 e^{\omega_1 L}$ .

Also, on  $\Gamma_6$ , letting  $\Gamma_6 : z = b + iy$  for  $x \in [\varepsilon, L]$  and applying the change of variable where  $\omega x = y$ , we get

$$\int_{\Gamma_6} \phi(z) dz = \frac{(-i)^{1+\beta}}{\omega^{1+\beta}} e^{i\omega b} \int_\varepsilon^L \frac{(b-a+\frac{i}{\omega}y)^\alpha \ln(-\frac{i}{\omega}y) \ln(b-a+\frac{i}{\omega}y) f(b+\frac{i}{\omega}y) y^\beta e^{-y}}{(b-\mu+\frac{i}{\omega}y)^k} dy,$$

similarly on  $\Gamma_8$ , we have

$$\int_{\Gamma_8} \phi(z) dz = \left(\frac{i}{k}\right)^{1+\alpha} e^{i\omega a} \int_\varepsilon^L \frac{(b-a-\frac{i}{\omega}y)^\beta \ln(\frac{i}{\omega}y) \ln(b-a-\frac{i}{\omega}y) f(a+\frac{i}{\omega}y) y^\alpha e^{-y}}{(a-\mu+\frac{i}{\omega}y)^k} dy.$$

Integral over  $\Gamma_3 : z = \mu + \varepsilon e^{ix}$  for  $0 \leq x \leq \pi$  we apply the formula (1.3) to get

$$\int_{\Gamma_3} \phi(z) dz = \frac{i\pi}{(k-1)!} \frac{d^{k-1}}{d\mu^{k-1}} \int_{\Gamma_3} \frac{\omega^{(\alpha,\beta,s)}(z) e^{i\omega z} f(z)}{z-\mu} dz.$$

The above right-hand side integral part gives

$$\int_0^\pi \frac{(\mu-a+\varepsilon e^{ix})^\alpha (b-\mu-\varepsilon e^{ix})^\beta \ln(\mu-a+\varepsilon e^{ix}) \ln(b-\mu-\varepsilon e^{ix}) f(\mu+\varepsilon e^{ix}) e^{i\omega\mu} e^{i\omega\varepsilon e^{ix}} i\varepsilon}{\varepsilon e^{ix}} dx,$$

as  $\varepsilon \rightarrow 0$ , the integral over  $\Gamma_3$  becomes

$$\int_{\Gamma_3} \phi(z) dz = \frac{i\pi}{(k-1)!} \frac{d^{k-1}}{d\mu^{k-1}} \left( \omega^{(\alpha,\beta,s)}(\mu) e^{i\omega\mu} f(\mu) \right).$$

Taking the limits as  $\varepsilon \rightarrow 0$ , and  $L \rightarrow \infty$  of the above results and substituting all in (2.5) completes the proof.

3. COMPUTATION OF (2.2) BY GAUSSIAN QUADRATURE RULE

In this section, we construct a Gaussian rule for efficiently computing the integral (2.2). The technique applied in this paper is based on the recent development of Walter Gautschi in his paper [28], followed the paper by Ball J.S. et al. [29]. For simplification let us rewrite (2.2) as

$$(3.1) \quad I[f; \mu] = G_1 \int_0^\infty F_1(y) \ln\left(\frac{i}{\omega}y\right) y^\alpha e^{-y} dy + G_2 \int_0^\infty F_2(y) \ln\left(-\frac{i}{\omega}y\right) y^\beta e^{-y} dy + H(\mu),$$

where

$$G_1 = \left(\frac{i}{\omega}\right)^{1+\alpha} e^{ika}, \quad F_1(y) = \frac{\left(b-a-\frac{i}{\omega}y\right)^\beta \ln\left(b-a-\frac{i}{\omega}y\right) f\left(a+\frac{i}{\omega}y\right)}{\left(a-\mu+\frac{i}{\omega}y\right)^k},$$

$$G_2 = \frac{(-i)^{1+\beta}}{\omega^{1+\beta}} e^{ikb}, \quad F_2(y) = \frac{\left(b-a+\frac{i}{\omega}y\right)^\alpha \ln\left(b-a+\frac{i}{\omega}y\right) f\left(b+\frac{i}{\omega}y\right)}{\left(b-\mu+\frac{i}{\omega}y\right)^k}$$

and

$$H(\mu) = \frac{i\pi}{(k-1)!} \frac{d^{k-1}}{d\mu^{k-1}} \left(\omega^{(\alpha,\beta,s)}(\mu) e^{i\omega\mu} f(\mu)\right) \text{ for } k = 1, 2, \dots$$

Drawing from the Gautschi analysis, we can transform the complex logarithmic multi-valued function that appeared in (3.1) as

$$(3.2) \quad \ln\left(\frac{i}{\omega}y\right) = \ln(i) - \ln\omega + \ln(y) = \left(\frac{\pi}{2}i - \ln\omega - 1 + y\right) - (y - 1 - \ln y)$$

and

$$(3.3) \quad \ln\left(-\frac{i}{\omega}y\right) = \ln(-i) - \ln\omega + \ln y = \left(-\frac{\pi}{2}i - \ln\omega - 1 + y\right) - (y - 1 - \ln y).$$

Substituting the above equations (3.2) and (3.3) into (3.1) leads us to the new form of (3.1) which can be denoted as

$$(3.4) \quad \begin{aligned} I[f; \mu] &= G_1 \int_0^\infty F_{11}(y) y^\alpha e^{-y} dy - G_1 \int_0^\infty F_1(y) (y - 1 - \ln y) y^\alpha e^{-y} dy \\ &+ G_2 \int_0^\infty F_{22}(y) y^\beta e^{-y} dy - G_2 \int_0^\infty F_2(y) (y - 1 - \ln y) y^\beta e^{-y} dy + H(\mu) \end{aligned}$$

where

$$F_{11}(y) = \left(\frac{\pi}{2}i - \ln\omega - 1 + y\right) F_1(y)$$

and

$$F_{22}(y) = \left(-\frac{\pi}{2}i - \ln\omega - 1 + y\right) F_2(y).$$

Moreover, analyzing (3.4) we came across the following related (nonnegative) weight functions:

$$(3.5) \quad \begin{aligned} w^{(\alpha)}(y) &= y^\alpha e^{-y}, \quad w^{(\alpha,LG)}(y) = y^\alpha e^{-y} (y - 1 - \ln y), \\ w^{(\beta)}(y) &= y^\beta e^{-y}, \quad w^{(\beta,LG)}(y) = y^\beta e^{-y} (y - 1 - \ln y). \end{aligned}$$

Here  $w^{(\alpha,LG)}$  and  $w^{(\beta,LG)}$  should be recognized as logarithmic Gautschi-Laguerre weight functions over a positive half-infinite interval whereas the rest are known to be the conventional Generalized Gauss Laguerre weight functions  $y \rightarrow w(y; m) = y^m e^{-y}$  for  $(m = \alpha, \beta > -1)$ .

By combining logarithmic Gautschi-Laguerre with the traditional Generalized Laguerre weight functions, a Gauss-related quadrature rule for the integral (3.4) can be formulated as follows

$$(3.6) \quad \begin{aligned} I_n[f; \mu] &= G_1 \sum_{p=1}^n w_p^{(\alpha)} F_{11} \left( y_p^{(\alpha)} \right) - G_1 \sum_{p=1}^n w_p^{(\alpha, LG)} F_1 \left( y_p^{(\alpha, LG)} \right) \\ &+ G_2 \sum_{p=1}^n w_p^{(\beta)} F_{22} \left( y_p^{(\beta)} \right) - G_2 \sum_{p=1}^n w_p^{(\beta, LG)} F_2 \left( y_p^{(\beta, LG)} \right) + H(\mu) \end{aligned}$$

In the above summations (3.6)  $y_p^{(m)}, y_p^{(m, LG)}$  and  $w_p^{(m)}, w_p^{(m, LG)}$  for  $(m \in \{\alpha, \beta\})$  are the nodes and weights for the rule, respectively.

The problem left is the computation of the approximation  $I_n$ ( the abscissas  $y_p^{(m)}, y_p^{(m, LG)}$  and the weights  $w_p^{(m)}, w_p^{(m, LG)}$  for  $(m \in \{\alpha, \beta\})$ ). However, before that, we first require the knowledge of the recursion coefficients  $\alpha_i, \beta_i$ . To do so, let  $\varphi$  be the space of real polynomials of degree at most  $n$  and suppose that  $w : (a, b) \rightarrow [0, \infty)$  is the nonnegative weight function with  $w(y) > 0$  and all moments  $\mu_r = \int_a^b y^r w(y) dy, \mu_0 > 0$  exist and are finite. The generalization of the dot product known as the inner product is defined as

$$(3.7) \quad (u, v) = \int_{\mathbb{R}} u(y) v(y) w(y) dy.$$

Provided that  $u$  is orthogonal to  $v$  if  $(u, v) = 0$  and the norm of  $u$  is obtained if  $u = v$  which yields

$$(3.8) \quad \|u\| = \left( \int_a^b u^2(y) w(y) dy \right)^{1/2}.$$

The monic orthogonal polynomials with respect to (3.7) are denoted by  $\pi_i(\cdot) = \pi_i(\cdot, w(y))$ ,  $i = 0, 1$ , and they satisfy the three-term recurrence relation

$$(3.9) \quad \begin{aligned} \pi_{i+1}(y) &= (y - \alpha_i) \pi_i(y) - \beta_i \pi_{i-1}(y), \quad i = 0, 1, \dots \\ \pi_0(y) &= 1, \quad \pi_{-1}(y) = 0, \end{aligned}$$

where

$$\alpha_i = \frac{(y \pi_i, \pi_i)}{(\pi_i, \pi_i)}, \quad i = 0, 1, 2, \dots$$

and

$$\beta_i = \frac{(\pi_i, \pi_i)}{(\pi_{i-1}, \pi_{i-1})}, \quad i = 1, 2, \dots$$

The coefficient  $\beta_0$  may be arbitrary but, by definition, is defined as  $\beta_0 = \mu_0 = \int_{\mathbb{R}} w(y) dy$ . Automatic computation of the above coefficients plays a significant role in the construction of the numerical quadrature rule (3.6). Moreover, their information will facilitate the roots of orthogonal polynomials to be quickly evaluated as characteristic values (eigenvalues) of the  $J_n$  (Jacobi) matrix. Among numerous algorithms that exist, that of Golub and Welsch [30] is the most popular. Their algorithm is based on the calculation of the eigenvalues, and corresponding eigenvectors of the matrix  $J$  of order  $n$  [31] whose diagonal elements are  $\{\alpha_i\}_{i=0}^{n-1}$  and sub-diagonal elements are  $\{\sqrt{\beta_i}\}_{i=1}^{n-1}$

$$(3.10) \quad J = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & 0 \\ \sqrt{\beta_1} & \alpha_1 & \ddots & \\ \vdots & \ddots & \ddots & \sqrt{\beta_{n-1}} \\ 0 & & \sqrt{\beta_{n-1}} & \alpha_{n-1} \end{bmatrix}.$$

Furthermore, Golub and Welsch demonstrated that we only need to determine eigenvalues and the first component of the corresponding normalized eigenvector  $v_{p,1}$  and their algorithm only cost  $O(n^2)$  operations. The nodes

$y_p$  ( $p = 1, 2, \dots, n$ ) are the eigenvalues of  $J$  and the weights  $w_p$  ( $p = 1, 2, \dots, n$ ) are calculated by taking  $w_p = \lambda_{n,p} = \beta_0 v_{p,1}^2$  for  $p = 1, 2, \dots, n$ , where  $\beta_0$  is defined by  $\beta_0 = \mu_0$ ,

$$J(w) v_p = y_p v_p, \quad v_p^T v_p = 1,$$

$v_{p,1}$  is the first component of the normalized eigenvector  $v_p$  ( $= [v_{p,1}, v_{p,2}, \dots, v_{p,n}]^T$ ) associated with the eigenvalue  $y_p$ . Furthermore, in the determination of the eigenvalues of the matrix  $J$ , the QR method proposed by Francis [32] is deemed the best method available since it converges very rapidly for symmetric matrices.

There exist several methods for generating the  $n$  recursion coefficients. As stated previously, we employ the method based on the moment's information.

It is well known that the first  $n$  recursion coefficients  $\alpha_i, \beta_i, i = 0, 1, \dots, n - 1$  can be evaluated by the first  $2n$  moments  $\mu_r = \int_a^b y^r w(y) dy, r = 0, 1, \dots, 2n - 1$ . Consequently, the moments associated with the weight functions in (3.5) (taking  $\alpha$  as an example) can be calculated exactly in an efficient way as

$$(3.11) \quad \begin{aligned} \mu_r^{(\alpha)} &= \int_0^\infty y^{\alpha+r} e^{-y} dy = \Gamma(\alpha + r + 1), \\ \mu_r^{(\alpha,L)} &= \int_0^\infty y^{\alpha+r} e^{-y} (y - 1 - \ln y) dy = \Gamma(\alpha + r + 1) [\alpha + r - \psi(\alpha + r + 1)], \end{aligned}$$

where  $\Gamma(z) = \int_0^\infty y^{z-1} e^{-y} dy$  is the Gamma function and  $\psi(z)$  is the logarithmic derivative of the Gamma function  $\psi(z) = \Gamma'(z) / \Gamma(z)$  known as Digamma function [33], and it can be implemented in Mathematica as PolyGamma[z]. Several formulas exist, for instance, by expressing the coefficients in terms of Hankel determinants in these moments, we obtain the  $n$  recursive coefficients. However, the sensitivity of the formula to small errors increases with  $n$  due to the condition number that grows with  $n$  if an extended precision is not employed [34]. For constructing recursive coefficients that satisfy three-term recurrence relation for orthogonal polynomials with respect to the given weight functions  $w(y; m) = y^m e^{-y}$  and  $w(y; m) = (y - 1 - \ln y) y^m e^{-y}$  for  $m = \alpha, \beta > -1$ , the moments obtained in (3.11) are sufficient.

**3.1. Algorithm 1: Computation of the coefficients  $\{\alpha_i^{LG}, \beta_i^{LG}\}_{i=0}^{n-1}$  using ordinary moments.**

**Input values:**  $n; \alpha$ ; prec (\*Precision\*); dig (\*Working Precision\*);

**Outputs:**

- 1: Compute  $\mu_r^{(\alpha,L)}$  as given in (3.11) for  $r = 0, 1, \dots$ ;
- 2: Evaluate  $\alpha_0 = \mu_1^{(\alpha,L)} / \mu_0^{(\alpha,L)}$ ;
- 3: Evaluate  $\beta_0 = \mu_0^{(\alpha,L)}$ ;
- 4: Do  $\lambda_{-1,d} = 0, d = 1, 2, \dots, 2n - 2$ ;
- 5: Do  $\lambda_{0,d} = \mu_d^{(\alpha,L)}, d = 0, 1, \dots, 2n - 1$ ;
- 6: For  $i = 1, 2, \dots, n$  Do  $\lambda_{i,d} = \lambda_{i-1,d+1} - \alpha_{i-1} \lambda_{i-1,d} - \beta_{i-1} \lambda_{i-2,d}, d = i, i + 1, \dots, 2n - i - 1$ ;
- 7: Evaluate  $\alpha_i = \lambda_{i,i+1} / \lambda_{i,i} - \lambda_{i-1,i} / \lambda_{i-1,i-1}$ ;
- 8: Evaluate  $\beta_i = \lambda_{i,i} / \lambda_{i-1,i-1}$ ;
- 9: Return  $\alpha_i; \beta_i$  for  $i = 0, 1, \dots, n - 1$ ;

Table 1: Coefficients  $\{\alpha_i^{LG}, \beta_i^{LG}\}_{i=0}^{14}$  corresponding to the weight  $w^{(\alpha, LG)}(y) = y^\alpha e^{-y} (y - 1 - \ln y)$ , for  $\alpha = -1/4$ , obtained by executing program 1, CPU=0.015625.

$i$	$\alpha_i^{LG}$	$\beta_i^{LG}$
0	0.45090716524038707962219237449708	1.0242778830275809860335715955718
1	4.4026356977629687472295132612258	1.2587291457146847219809737953640
2	6.4462609107426586575051423712921	4.3437841029461997884958893605676
3	8.0993469536298673147110463146371	10.727459601510999531760131065961
4	9.6565521442501219312044582407037	20.266321990048429475466086706460
5	11.487385620636749776309299477905	31.808060947238646026307463203268
6	13.660865631852677441998498517609	43.433947558394725076095085656902
7	15.923546713504372605163548843941	55.456655209891862267583643063180
8	18.103197479766207329491573123388	69.425911108055039576435436564038
9	20.172145438709609898281127079405	86.132945475939104712802024368667
10	22.148685896189116656533807126260	105.72362050316907059519290527165
11	24.054429060576049496535522934793	128.17144559395119262200846729860
12	25.914735451923003844755927791835	153.38231588056735603882297667714
13	27.766370023209773725154094863369	181.08839565056932241218160342789
14	29.653839520173234572797220416703	210.76020051420847765662219483336

Another alternative is to utilize the modified moments where instead of evaluating moments by the power  $y^r$  in (3.11), we substitute it with an appropriate orthogonal polynomial of degree  $i$ . That is given a set of polynomials  $\{P_i(y)\}_{i=0}^\infty$  degree  $(P_i = i)$ , the modified moments  $\hat{\mu}_i$  of the weight function  $w$  is defined by

$$\hat{\mu}_i = \int_a^b P_i(y) w(y) dy, \quad i = 0, 1, 2, 3, \dots$$

In this case, let  $\hat{L}_i^m(y)$  be the monic generalized Laguerre polynomials [35] and suppose that  $\{\hat{L}_i^m\}$  are monic polynomials satisfying a three-term recurrence relation similar to (3.9)

$$(3.12) \quad \begin{aligned} \hat{L}_{i+1}^m(y) &= (y - a_i) \hat{L}_i^m(y) - b_i \hat{L}_{i-1}^m(y), \quad i = 0, 1, \dots \\ \hat{L}_0^m(y) &= 1, \hat{L}_{-1}^m(y) = 0. \end{aligned}$$

In the above recurrence relation, the coefficients  $a_i \in \mathbb{R}, b_i \geq 0$  can be determined in the following way: We already know that

$$(3.13) \quad \begin{aligned} (i + 1) L_{i+1}^m(y) &= (2i + m + 1 - y) L_i^m(y) - (i + m) L_{i-1}^m(y), \\ L_0^m(y) &= 1, L_1^m(y) = m + 1 - y. \end{aligned}$$

Inserting  $L_i^m(y) = \frac{\hat{L}_i^m(y)}{(-1)^i i!}$  in the above recurrence relation (3.13) we get

$$(3.14) \quad \begin{aligned} \hat{L}_{i+1}^m(y) &= [y - (2i + m + 1)] \hat{L}_i^m(y) + i(i + m) \hat{L}_{i-1}^m(y), \\ \hat{L}_0^m(y) &= 1, \hat{L}_{-1}^m(y) = 0. \end{aligned}$$

Equating (3.12) and (3.14) we observe that the coefficients  $a_i \in \mathbb{R}, b_i \geq 0$  are

$$(3.15) \quad \begin{aligned} a_i &= 2i + m + 1 \text{ and } b_i = i(i + m), \\ b_0 &= \mu_0 = \Gamma(m + 1). \end{aligned}$$

Thus, the modified moments related to the weight function  $w^{(m, LG)}(y) = y^m e^{-y} (y - 1 - \ln y)$ , over  $[0, \infty)$  for  $(m = \alpha, \beta > -1)$  is expressed in the Gamma and its logarithmic derivative function as

$$(3.16) \quad \hat{\mu}_i = \int_0^\infty w^{(m, LG)}(y) \hat{L}_i^m(y) dy = \begin{cases} m\Gamma(m + 1) - \Gamma(m + 1)\psi(m + 1), & \text{if } i = 0 \\ m\Gamma(m + 1), & \text{if } i = 1 \\ (-1)^i (i - 1)! \Gamma(m + 1), & \text{if } i \geq 2. \end{cases}$$

Thanks to the modified Chebyshev algorithm [ [36], p. 76], which directly takes  $2n$  modified moments and the  $2n - 1$  coefficients  $\{a_i, b_i\}_{i=0}^{2n-2}$  in (3.12) to produce the coefficients  $\alpha_i(w), \beta_i(w), i = 0, 1, \dots, n - 1$  that we strongly desired. In order to generate coefficients using the modified moments' approach, we provide an algorithm that can be implemented in any mathematical software. Our implementation was done using Mathematica version 9.0.

**3.2. Algorithm 2: Computation of the coefficients  $\{\alpha_i^{LG}, \beta_i^{LG}\}_{i=0}^{n-1}$  using the modified moments.**

**Input values:**  $n; \alpha = m;$  prec (\*Precision\*); dig (\*Working Precision\*);

**Outputs:**

- 1: Compute  $a_i = 2i + \alpha + 1; i = 1, 2, \dots, 2n - 1;$
- 2: Evaluate  $b_i = \text{If } i = 0, \Gamma(\alpha + 1) \text{ else } i(i + \alpha);$
- 3: Evaluate  $\hat{\mu}_i = \text{If } i = 0, \hat{\mu}_0 \text{ else If } i = 1, \mu_1 \text{ else } (-1)^i (i - 1)! \Gamma(\alpha + 1);$
- 4: Approximate  $\alpha_0 = a_0 + \hat{\mu}_1 / \hat{\mu}_0;$
- 5: Approximate  $\beta_0 = \hat{\mu}_0$  as shown in step 3;
- 6: Do  $\lambda_{-1,d} = 0, d = 1, 2, \dots, 2n - 2;$
- 7: Do  $\lambda_{0,d} = \hat{\mu}_d, d = 0, 1, \dots, 2n - 1;$
- 8: For  $i = 1, 2, \dots, n - 1$  Do

$$\lambda_{i,d} = \lambda_{i-1,d+1} - (\alpha_{i-1} - a_d) \lambda_{i-1,d} - \beta_{i-1} \lambda_{i-2,d} + b_d \lambda_{i-1,d-1}, \quad d = i, i + 1, \dots, 2n - i - 1;$$

- 9: Evaluate  $\alpha_i = a_i + \lambda_{i,i+1} / \lambda_{i,i} - \lambda_{i-1,i} / \lambda_{i-1,i-1};$
- 10: Evaluate  $\beta_i = \lambda_{i,i} / \lambda_{i-1,i-1};$
- 11: Return  $\alpha_i; \beta_i$  for  $i = 0, 1, \dots, n - 1;$

Coefficients obtained from Algorithm 2 by using modified moments are in full agreement with the ones provided by Algorithm 1 using ordinary moments. Although both algorithms lead to the same results, the first algorithm requires the employment of extended precision compared to the second algorithm, resultantly, a slight computation time difference between the two algorithms may be observed. This problem is due to the ill-conditioning that grows exponentially with  $n$ .

As an example, by considering the Gautschi logarithmic weight function, for each  $n < 15$  Algorithm 1 requires a precision level of 60 in 0.015625 seconds, whereas Algorithm 2 requires precision equal to 50 in 0.015625 seconds. In order to obtain  $n < 20$  recurrence coefficients, Algorithm 1 requires a precision level of 75 in 0.06250 seconds, whereas

Algorithm 2 requires a precision level of 50 in 031250 seconds. Better illustrative examples are provided in the next section.

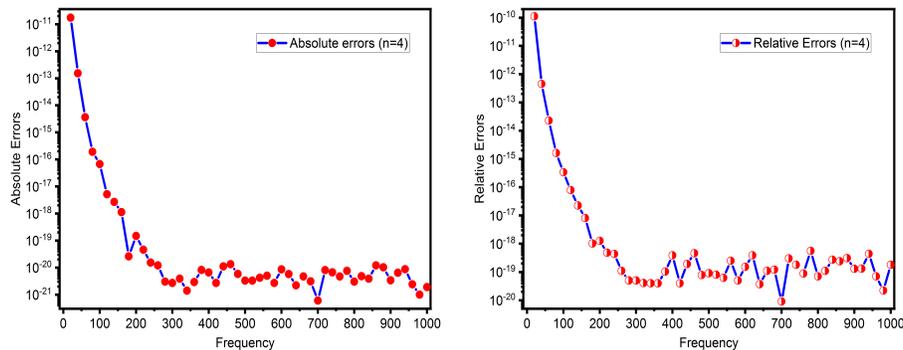


FIGURE 2. Convergence rate for hypersingular integral (1.1) with  $\alpha = -1/4$ ,  $\beta = -1/5$ ,  $\mu = 0$ ,  $k = 2$  and  $f(x) = \sin x$ .

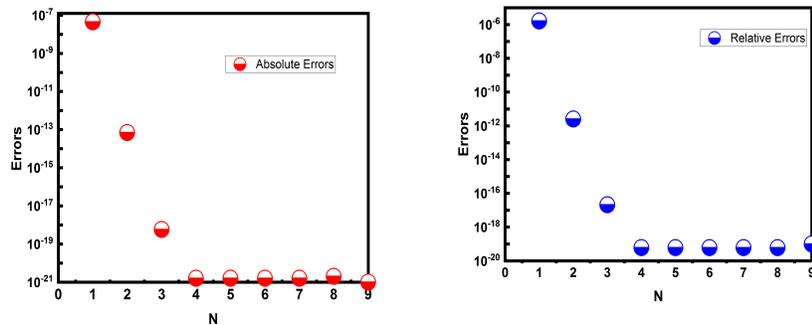


FIGURE 3. Errors for the integral  $\int_{-1}^1 \frac{(1+x)^{-0.25}(1-x)^{0.5} \ln(1+x) \ln(1-x) e^{i\omega x} e^x}{x-0.1} dx$ , for a fixed  $\omega = 10^3$ .

#### 4. NUMERICAL EXPERIMENTS

Herein, selected numerical experiments are manifested to examine the effectiveness of the proposed method to hypersingular integrals. All performed experiments were carried out using programs written in Mathematica with 32-digits arithmetic. Nevertheless, we will require exact values to calculate errors. Furthermore, the so-called exact values were computed in Mathematica version 9.0 on a personal laptop with 4GB RAM. Computation time in seconds is given in some examples.

Example 1. We compute the following hypersingular integral

$$(4.1) \quad I^{[1]} [f; \mu] = \int_{-1}^1 \frac{\omega^{(\alpha, \beta, 1)}(x) e^{i\omega x} \sin x}{(x - \mu)^2} dx, \quad |\mu| < 1,$$

for  $\alpha = -1/4$ ,  $\beta = -1/5$ ,  $\mu = 0$ . Table 2 shows errors and execution time for the integral (4.1) with  $n = 4$ .

Table 2: Absolute errors and execution time in seconds for the integral (4.1) with  $n = 4$ .

$\omega$	Absolute Errors	CPU
10	$1.85 \times 10^{-9}$	0.031
$10^2$	$6.74 \times 10^{-18}$	0.046
$10^3$	$1.90 \times 10^{-22}$	0.031
$10^4$	$7.40 \times 10^{-23}$	0.031
$10^5$	$1.10 \times 10^{-23}$	0.140

Example 2. Consider the hypersingular integral

$$(4.2) \quad I^{[1]} [f; \mu] = \int_{-1}^1 \frac{\omega^{(\alpha, \beta, 1)}(x) e^{i\omega x} \tan x}{(x - \mu)^3} dx, \quad |\mu| < 1,$$

for  $\alpha = -0.3, \beta = 0.6, \mu = 0.5$ . Table 3 shows the approximation values obtained using the proposed method with different values of  $n$  and  $\omega$  fixed. Moreover, Table 3 exhibits that as  $n$  increases, the computation accuracy also increases.

Table 3: Approximated values for the hypersingular integral (4.2) with  $\omega$  fixed.

$\omega = 1000$	
$n$	Approximated values
2	64532.688961214841864763359257867-125250.851427138988397399906359252 i
3	64532.688961214841228930837308266-125250.851427138987821012698178167 i
4	64532.688961214841228950454220065-125250.851427138987821049554599396 i
5	64532.68896121484122895045299769-125250.851427138987821049550931783 i
6	64532.68896121484122895045299781-125250.85142713898782104955093231 i
7	64532.6889612148412289504529978-125250.85142713898782104955093231 i

Example 3. We compute the following Hadamard finite part efficiently

$$(4.3) \quad I^{[1]} [f; \mu] = \int_{-1}^1 \frac{\omega^{(\alpha, \beta, 1)}(x) e^{i\omega x} (x \cos x + x^6 + 3x - 1)}{(x - \mu)^2 (1 + x^2)} dx, \quad |\mu| < 1,$$

for  $\alpha = 1/100, \beta = -1/4, \mu = 0$ . Table 4 shows errors and time in seconds obtained using the proposed method for different values of  $\omega$ .

Table 4: Errors and CPU time in seconds for the integral (4.3) with  $n = 5$ .

$\omega$	Absolute Errors	CPU
50	$1.80 \times 10^{-14}$	0.046
$10^2$	$1.00 \times 10^{-17}$	0.062
$10^3$	$9.00 \times 10^{-28}$	0.218
$10^4$	$1.30 \times 10^{-28}$	0.203
$10^5$	$6.80 \times 10^{-28}$	0.187

Example 4. We consider the computation of hypersingular integral

$$(4.4) \quad I^{[1]} [f; \mu] = \int_{-1/4}^1 \frac{\omega^{(\alpha, \beta, 1)}(x) e^{i\omega x} 1000 \ln(1+x)}{(x-\mu)^2(1+x)} dx,$$

for  $\alpha = -0.31, \beta = 0.69, \mu = 0$  using the two algorithms. Table 5 shows the errors and CPU time in seconds using the proposed algorithms with  $\omega = 350$  (frequency) fixed.

Table 5: Errors and CPU time in seconds for the integral (4.4) with  $\omega = 350$  fixed.

$n$	Absolute Errors		CPU	
	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
3	$3.15 \times 10^{-11}$	$1.41 \times 10^{-9}$	0.015	0.031
4	$7.12 \times 10^{-15}$	$3.32 \times 10^{-13}$	0.031	0.031
5	$2.60 \times 10^{-18}$	$1.21 \times 10^{-16}$	0.062	0.062
6	$1.36 \times 10^{-21}$	$6.32 \times 10^{-20}$	0.078	0.062
7	$1.00 \times 10^{-22}$	$4.47 \times 10^{-23}$	0.093	0.062

Exact value=

**64.62366062387822499728716115895 + 260.37702535680375105841469872353i** of the integral obtained in **3.70s**.

In Table 5, precision was fixed and equal to 40 for each algorithm. Table 6 exhibited that the accuracy of the two algorithms improved with  $n$ , although it can be seen that as  $n$  increases, the computation time in Algorithm 1 increased proportionally. Moreover, all the results were found to be entirely satisfactory.

Example 5. Consider the following algebraic singular Hadamard finite part:

$$(4.5) \quad I^{[0]} [f; \mu] = \int_{-1}^1 \frac{\omega^{(\alpha, \beta, 0)}(x) x e^x e^{i\omega x}}{(x-\mu)^2} dx, \quad |\mu| < 1,$$

with  $\alpha = -1/4, \beta = -1/2, \mu = \{0, 0.5\}$ . Table 6 shows Errors and CPU time in seconds using the proposed method with  $\omega = 100$  fixed and  $\mu = 0$ . Additionally, Table 7 shows the approximation values for  $\mu = 0.5$  using the method proposed.

Table 6: Absolute errors and execution time in seconds for the hypersingular integral (4.5) with  $\omega$  fixed.

$n$	$\omega = 1000$	
	Absolute Values	CPU
1	$1.32 \times 10^{-5}$	0.000
2	$3.65 \times 10^{-9}$	0.000
3	$2.77 \times 10^{-12}$	0.015
4	$3.81 \times 10^{-15}$	0.015
5	$8.00 \times 10^{-18}$	0.031

Exact value=

**0.1032823315565069232725554772316+2.7394850984212037657826989262612i** of the integral obtained in **2.93s**.

Table 7: Approximated values for the hypersingular integral (4.5) with  $\omega$  fixed and  $\mu = 0.5$ .

$\omega = 100$	
$n$	Approximated values
2	-223.68541472625086941647497358478+66.22139453813281267943681246412 i
4	-223.68541471558410137770140552331+66.22139373617587608195331122318 i
6	-223.68541471559458881378939193463+66.22139373615812273686197320247 i
8	-223.68541471559459171525944292159+66.22139373615812216947300856366 i
10	-223.68541471559459171603849043769+66.22139373615812217051001846092 i
12	-223.68541471559459171603764119079+66.22139373615812217051097259341 i
14	-223.68541471559459171603763949306+66.22139373615812217051097099077 i
16	-223.68541471559459171603763949889+66.22139373615812217051097098696 i
18	-223.6854147155945917160376394989+66.22139373615812217051097098699 i
20	-223.6854147155945917160376394989+66.22139373615812217051097098699 i

Taking  $n = 20$  as a reference, we can easily see that as  $n$  increases, the computational accuracy also increases.

Example 6. We consider the computation of

$$(4.6) \quad I^{[1]}[f; \mu] = \int_{-1}^1 \frac{\omega^{(-1/4, 1/3, 1)}(x) e^{i\omega x} f_j(x) dx}{(x - \mu)^2}, j = 1, 2$$

for

$$f_1(x) = \frac{e^x}{1+x^2}, f_2(x) = \frac{x e^x}{1+x^2}, \text{ and } \mu = 0.$$

Table 8 exhibits the accuracy of the proposed method for  $n = 4$  fixed. Furthermore, Table 8 shows the absolute errors and the executed time in seconds for both functions  $f_1$  and  $f_2$  with  $\omega = 10^2, 10^3, 10^4, 10^5, 10^6$ . We observe that the proposed method can guarantee at least  $|errors| \leq 10^{-15}$  for both functions  $f_1$  and  $f_2$ . It can be also seen through Table 8 that the proposed method is more efficient for large values of frequency.

Table 8: Absolute errors and execution time for (4.6) with different values of  $\omega$  and  $n = 4$  fixed.

$\omega$	$f_1(x) = \frac{e^x}{1+x^2}$		$f_2(x) = \frac{x e^x}{1+x^2}$	
	Absolute Errors	CPU	Absolute Errors	CPU
$10^2$	$4.82 \times 10^{-15}$	0.031	$6.83 \times 10^{-17}$	0.156
$10^3$	$9.00 \times 10^{-23}$	0.125	$8.00 \times 10^{-23}$	0.140
$10^4$	$1.90 \times 10^{-23}$	0.203	$1.90 \times 10^{-23}$	0.140
$10^5$	$6.10 \times 10^{-24}$	0.125	$6.10 \times 10^{-24}$	0.140
$10^6$	$1.40 \times 10^{-25}$	0.140	$1.30 \times 10^{-25}$	0.109

## CONCLUSIONS

Herein, the computational efficiency of algebraic and logarithmic hypersingular type singular integrals with highly oscillatory kernel was discussed. The approach employed proved that it can easily yield better approximation accuracy as  $n$  increases. Furthermore, the used approach also exhibited that taking  $n$  as fixed, increasing the frequency resultantly increases approximation accuracy. Also, two algorithms to produce the recursion coefficients that satisfy orthogonal polynomial with respect to Gautschi logarithmic weight function were given and compared. Moreover, both algorithms showed that they can give satisfactory results when one among them is used. The tables and convergence rate figures given hereinbefore provide evidence in support of our analysis.

**Conflicts of Interest:** The author(s) declare that there are no conflicts of interest regarding the publication of this paper.

## REFERENCES

- [1] N. M. A. Nik Long and Z. K. Eshkuvatov, Hypersingular integral equation for multiple curved cracks problem in plane elasticity, *Int. J. Solids Struct.* 46 (2009), 2611–2617.
- [2] I. A. Robertson, Diffraction of a plane longitudinal wave by a penny-shaped crack, *Math. Proc. Cambridge Philos. Soc.* 63 (1967), 229–238.
- [3] M. Landahl and H. Ashley, *Aerodynamics of Wings and Bodies*. New York: Dover Publication, Inc. New York, 1985.
- [4] A. Bostro, Review of hypersingular integral equation method for crack scattering and application to modeling of ultrasonic nondestructive evaluation, *Appl. Mech. Rev.* 56 (2003), 383–405.
- [5] E. G. Ladopoulos, *Singular Integral Equations: Linear and Non-linear Theory and its Applications in Science and Engineering*, Berlin, Springer-Verlag Berlin Heidelberg, New York, 2000.
- [6] J. Hadamard, *Le Probleme de Cauchy et les Equations aux Derivees Partielles Lineaires Hyperboliques*, Paris, France: Hermann, 1932.
- [7] J. Hadamard, *Lectures on Cauchy's Problem in Linear Partial Differential Equations*, Yale University Press, 1932.
- [8] G. Monegato, Numerical evaluation of hypersingular integrals, *J. Comput. Appl. Math.* 50 (1994), 9–31.
- [9] J. Li and X. Li, The Modified Trapezoidal Rule for Computing Hypersingular Integral on Interval, *J. Appl. Math.* 2013 (2013).
- [10] T. Hasegawa, T. Torii, An automatic quadrature for Cauchy principal value integrals, *Math. Comp.* 56 (1991), 741–741.
- [11] G. Criscuolo, A new algorithm for Cauchy principal value and Hadamard finite-part integrals, *J. Comp. Appl. Math.* 78 (1997), 255–275.
- [12] K. Diethelm, A method for the practical evaluation of the Hilbert transform on the real line, *J. Comp. Appl. Math.* 112 (1999), 45–53.
- [13] M. A. Golberg, The convergence of several algorithms for solving integral equations with finite part integrals. II, *Appl. Math. Comput.* 21 (1987), 283–293.
- [14] F. Erdogan, G. D. Gupta and T. S. Cook, Numerical solution of singular integral equations. in *Methods of analysis and solutions of crack problems*, Springer, The Netherlands, 1973.
- [15] R. Piessens, E. de Doncker-Kapenga, C. W. Überhuber and D. K. Kahaner, *A Subroutine Package for Automatic Integration*, Springer-Verlag, Berlin, 1983.

- [16] C. W. Clenshaw and A. R. Curtis, A method for numerical integration on an automatic computer, *Numer. Math.* 2 (1960), 197–205.
- [17] N. I. Muskhelishvili, *Singular Integral Equations*, Groningen, Noordhoff, The Netherlands, 1972.
- [18] D. Elliott, Orthogonal polynomials associated with singular integral equations having a Cauchy kernel, *SIAM J. Numer. Anal.* 13 (1982), 1041–1052.
- [19] E. Venturino, Recent developments in numerical solution of singular integral equations, *J. Math. Anal. Appl.* 115 (1986), 239–277.
- [20] G. C. Hsiao, P. Kopp and W. L. Wendland, A Galerkin collocation method for some integral equations of the first kind, *Computing.* 25 (1980), 83–130.
- [21] U. Zastrow, Numerical plane stress analysis by integral equations based on the singularity method, *Solid Mach. Arch.* 10 (1985), 113–128.
- [22] W. Feng, H. Li, L. Gao, W. Qian and K. Yang, Hypersingular flux interface integral equation for multi-medium heat transfer analysis, *Int. J. Heat Mass Transf.* 138 (2019), 852–865.
- [23] S. Alfaqeih and I. Kayijuka, Solving system of conformable fractional differential equation by conformable double Laplace decomposition method, *J. Part. Differ. Equ.* 33 (2020), 275–290.
- [24] H. Kang and X. Shao, Fast computation of singular oscillatory Fourier transforms, *Abstr. Appl. Anal.* 2014 (2014), 984834.
- [25] I. Kayijuka, J.P. Ndabakuranye and A. I. Hascelik, Computational Efficiency of Singular and Oscillatory Integrals with Algebraic Singularities, *Asian J. Math. Comp. Res.* 25 (2018), 285–302.
- [26] R.S. Murray, L. Seymour, J. S. John and S. Dennis, *Schaum's outline: Complex Variables: with an introduction to Conformal Mapping and its applications*, Second Ed. McGraw-Hill, New York, 2009.
- [27] D. Huybrechs and S. Vandewalle, On the Evaluation of Highly Oscillatory Integrals by Analytic Continuation, *SIAM J. Numer. Anal.* 44 (2006), 1026–1048.
- [28] W. Gautschi, Gauss quadrature routines for two classes of logarithmic weight functions, *Numer. Algorithms.* 55 (2010), 265–277.
- [29] J. S. Ball and N. H. F. Beebe, Efficient Gauss-related quadrature for two classes of logarithmic weight functions, *ACM Trans. Math. Softw.* 33 (2007), 1–21.
- [30] G. H. Golub and J. H. Welsch, Calculation of Gauss Quadrature Rules, *Math. Comput.* 23 (1969), 221–230.
- [31] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, 3rd ed. Springer, New York, 2002.
- [32] J. G. F. Francis, The QR transform. A unitary analogue to the LR transformation, *Comput. J.* 4 (1961), 265–271.
- [33] I. S. M. Abramowitz, *Handbook of Mathematical Functions*, National Bureau of Standards, 1964.
- [34] W. Gautschi, Algorithm 726: ORTHPOL—A Package of Routines for Generating Orthogonal Polynomials and Gauss-Type Quadrature Rules, *ACM Trans. Math. Softw.* 20 (1994), 21–62.
- [35] L. Gatteschi, On Some Orthogonal Polynomial Integrals, *Math. Comput.* 35 (1980), 1291–1298.
- [36] W. Gautschi, *Orthogonal Polynomials: Computation and Approximation*, Oxford University Press, 2004.