J. Approx. Softw., Vol. 2, No. 1 (2025) 2

Journal of Approximation Software

doi: 10.13135/3103-1935/11939 – ISSN: 3103–1935

AGquad: a Matlab package for 1D and 2D anti-Gauss type rules

P. Díaz de Alba [1], L. Fermo [1⋆], and G. Rodriguez [1]

[1] *Department of Mathematics and Computer Science, University of Cagliari*

Received: 28/04/2025 – Published: 15/09/2025
Communicated by: A. Sommariva (from the SA2025 workshop)

Abstract

Anti-Gauss formulae have attracted a great deal of interest in recent years. There are two main reasons for this. On the one hand, they make it possible to construct new rules, namely averaged or stratified formulae, which are characterized by a higher accuracy and a lower computational cost, compared to formulae with the same number of nodes. On the other hand, they provide numerical estimates of the error of the Gaussian rule for a fixed number of points. This allows one to determine the number of nodes required to achieve a prescribed accuracy in the integral approximation. In this paper, we provide a Matlab toolbox, including an interactive graphical user interface (GUI), to assist the interested reader in performing the computation of 1D integrals using anti-Gauss formulae and one of their generalization. When coupled to Gauss rules, they lead to averaged and generalized averaged scheme, respectively. We also consider the computation of 2D integrals by Gauss/anti-Gauss rules. With the exception of the GUI, the software also runs on Octave.

Keywords: Orthogonal polynomials, Gauss rules, anti-Gauss formulae, averaged schemes (MSC2020: 65D30, 42C05)

1 Introduction

One of the most important and widely used integration formula is the Gauss quadrature rule G_n [8]

$$I(f) = \int_{-1}^{1} f(x)w(x)\,dx = \sum_{j=1}^{n} \lambda_j f(x_j) + e_n(f) =: G_n(f) + e_n(f), \tag{1}$$

where f is a known integrable function and w is the Jacobi weight given by

$$w(x) = (1-x)^{\alpha}(1+x)^{\beta}, \qquad \alpha, \beta > -1. \tag{2}$$

⋆ Corresponding author: fermo@unica.it

In (1), λ_j is the j-th Christoffel number, x_j is the j-th zero of $p_n(w)$, i.e., the polynomial of degree n orthonormal with respect to the weight w, and $e_n(f)$ is the quadrature error. It is an interpolatory formula that exactly integrates polynomials $p \in \mathbb{P}_{2n-1}$, where $\mathbb{P}_{2n-1}$ denotes the set of algebraic polynomials of degree at most $2n-1$. Its main important feature is that it is the optimal integration formula: there are no other formulae based on n points with a better degree of accuracy. In a nutshell, if a n-point formula is such that $e_n(p) = 0$ for any $p \in \mathbb{P}_{2n-1}$, then it is a Gauss rule.

Over the years, several researchers have been interested in estimating the error

$$e_n(f) = I(f) - G_n(f), \tag{3}$$

to determine the number of nodes needed to achieve a given accuracy: too few nodes will produce an error greater than the desired one, while too many nodes will require the integrand to be evaluated at an unnecessarily large number of points, leading to an increase in both computing time and error propagation. A classical approach is to replace the exact value of the integral in (3) by a new formula Q_m, with $m > n$, obtaining

$$e_n(f) \simeq Q_m(f) - G_n(f). \tag{4}$$

A simple choice is to set $Q_m(f) = G_{n+1}(f)$, but estimating the difference $G_{n+1}(f) - G_n(f)$ may be unreliable [1], and the degree of exactness would only slightly improve, from $2n-1$ to $2n+1$. Motivated by this, Kronrod introduced the so-called Gauss-Kronrod formula for the Legendre weight [11, 12], later developed for other kind of measures [9, 15]

$$G^K_{2n+1} = G_n(f) + \sum_{j=1}^{n+1} \hat{\lambda}_j f(\hat{x}_j),$$

where the coefficients $\hat{\lambda}_j$ and $\hat{x}_j$ are chosen so that the maximum degree of exactness is (at least) $3n+1$. The obvious advantage of such a formula is that the degree of exactness improves considerably, from $2n-1$ to $3n+1$, at the price of $n+1$ further evaluations of the integrand function f. The main disadvantage is that the existence of a Gauss-Kronrod rule with positive coefficients, which is an essential stability condition, depends on the measure and on the number of nodes. Several cases of non-existence of such a formula are known. For example, in the case of the weight (2) with $\min\{\alpha,\beta\} \geq 0$ and $\max\{\alpha,\beta\} > 5/2$, there is no Gauss-Kronrod formula with positive coefficients, for n sufficiently large; see [19, 20].

In 1996, D. P. Laurie constructed a new formula $\widetilde{G}_{n+1}$ having the same degree of accuracy as the corresponding G_n formula and an error equal in magnitude to e_n but opposite in sign, when applied to polynomials of degree up to $2n+1$ [14]. Because of this property, he called it the *anti-Gauss rule*.

As Laurie noted in his paper, the idea of constructing a numerical method that produces errors of the same magnitude but of opposite sign was not new. It had already been applied to the numerical solution of initial value differential problems [2, 22]. The application of this idea in the field of numerical integration has allowed the construction of more accurate and computationally advantageous formulae. These are known as *stratified nested rules*, a term coined by Laurie in an earlier paper [13]; see also [18]. A pair of stratified nested formulae have the property that the nodes of the first rule appear also in the second one, and the corresponding weights are multiplied times a constant. The simplest stratified formula is the *averaged formula*, i.e. a simple average of G_n and $\widetilde{G}_{n+1}$; see relation (8). However, several stratified schemes have been developed during the years, for several kind of weight functions [6, 23, 24, 25] and explored in the context of the numerical treatment of second kind Fredholm integral equations [3, 4, 5, 7].

In 2007, M. M. Spalević developed an optimal stratified rule Q^*_{2n+1} based on $2n+1$ nodes that he called *generalized averaged Gauss rule* [24]. Its main feature is the degree of exactness: it is exact for polynomials of degree at least $2n+2$, but if the weight function w is even, i.e., $w(x) = w(-x)$, the degree is $2n+3$. Moreover, for specific measures the degree of exactness is much higher; see [6, 16, 26]. In his paper, Spalević proved, in particular, the positivity of the coefficients and the sufficient conditions under which the quadrature points belong to the interval $[-1,1]$. He also observed that the Gauss nodes x_k are a subset of the quadrature points of the formula, and showed that nodes and weights can be computed by solving the eigenvalue problem associated to a matrix of order $2n+1$. It is in 2021, in collaboration with L. Reichel [23], that he reduced the computational cost by decomposing the formula as a convex combination of a Gauss rule G_n and a new formula G^*_{n+1}, further studied in [7] in terms of stability and convergence. We refer to G^*_{n+1} as the *generalized anti-Gauss rule*.

The aim of this paper is to provide a Matlab package for the numerical evaluation of integrals using the mentioned above formulae. In Section 2, we consider the univariate case and present the anti-Gauss rule, the generalized formula, and the corresponding averaged rules. In this context, we refer to [14] and [3] for the anti-Gauss rule, and to [23] and [7] for the generalized version. In Section 3, we approximate bivariate integrals by the anti-Gauss cubature formula, developed for the first time in [4], and discuss the corresponding averaging formula. Finally, in Section 4, we report a selection of numerical examples, some of which appeared in the aforementioned papers, to both demonstrate the performance of the methods considered, and explain to the reader how to use our Matlab software.

The software is available from GitHub at https://github.com/CaNAgroup/AGquad and at the web page https://bugs.unica.it/cana/software.html.

2 The 1D case

In this section, we deal with the numerical treatment of integrals defined on $[-1,1]$ whose integrand function may have algebraic singularities at the endpoints ± 1. Such a problem can be reformulated as

$$I(f) = \int_{-1}^{1} f(x)w(x)\,dx, \tag{5}$$

where f is a known integrable function and w is the Jacobi weight given in (2).

We point out that the choice of the interval $[-1,1]$ is not restrictive, since any integral defined on a generic interval $[a,b]$ can be traced back to (5) by the change of variable

$$\int_a^b F(y)(b-y)^\alpha (y-a)^\beta\,dy = \left(\frac{b-a}{2}\right)^{\alpha+\beta+1} \int_{-1}^{1} f(x)w(x)\,dx,$$

where $f(x) := F\left(\frac{(b-a)}{2}(x+1)+a\right)$.

2.1 The anti-Gauss rule and the averaged formula

The anti-Gauss formula $\widetilde{G}_{n+1}$, given by Laurie in [14] to approximate (5), reads

$$\widetilde{G}_{n+1}(f) = \sum_{j=1}^{n+1} \tilde{\lambda}_j f(\tilde{x}_j), \tag{6}$$

where the coefficients $\{\tilde{\lambda}_j\}_{j=1}^{n+1}$ and the nodes $\{\tilde{x}_j\}_{j=1}^{n+1}$ are determined so that for the corresponding quadrature error $\tilde{e}_{n+1}(f)$ it holds

$$\tilde{e}_{n+1}(f) := I(f) - \widetilde{G}_{n+1}(f) = -e_n(f), \quad \text{for all } f \in \mathbb{P}_{2n+1}, \tag{7}$$

with e_n given in (3).

From (7), one gets

$$I(f) = \frac{1}{2}[\widetilde{G}_{n+1}(f) + G_n(f)] =: Q_{2n+1}(f), \quad \text{for all } f \in \mathbb{P}_{2n+1}, \tag{8}$$

which is a new formula involving $2n+1$ nodes and is more accurate than the two single formulae involved, having degree of accuracy $2n+1$. Such a formula is named *averaged quadrature rule* and can be used in (4) to estimate the Gauss quadrature error

$$e_n(f) \simeq Q_{2n+1}(f) - G_n(f) = \frac{1}{2}[\widetilde{G}_{n+1}(f) - G_n(f)] =: r_n(f). \tag{9}$$

In this way, one may choose the number of nodes n so that the Gauss rule G_n reaches a prescribed accuracy. Let us also further emphasize that from (7) one can deduce that if $f \in \mathbb{P}_{2n+1}$, the Gauss and anti-Gauss rules provide an interval which contains the value of the integral, i.e., either

$$I(f) \in [G_n(f), \widetilde{G}_{n+1}(f)] \qquad \text{or} \qquad I(f) \in [\widetilde{G}_{n+1}(f), G_n(f)]. \tag{10}$$

This property is often denoted as *bracketing* of the exact integral. Notice that the size of the interval decreases as n increases.

It is possible to prove that relation (7), and hence (10), is still valid for a generic $f \notin \mathbb{P}_{2n+1}$ only under suitable conditions; see, for instance, [21, Theorem 3.1] and [3]. Such assumptions are only sufficient conditions for the *bracketing* property, and are rather restrictive. However, numerical experiments show that the change of sign in the error is observed in a wide class of integrand functions, at the point that it is rather difficult to find a function for which it does not happen. It is still an open problem to prove that bracketing occur under wider assumptions than those proposed in [21, Theorem 3.1]. A result in this direction is given in [3, Corollary 1] for the Chebyshev weight.

Laurie proved in [14] that the coefficients $\{\tilde{\lambda}_j\}_{j=1}^{n+1}$ of the anti-Gauss rule (6) are all positive, and the nodes $\{\tilde{x}_j\}_{j=1}^{n+1}$ are zeros of the polynomial defined by

$$\tilde{p}_{n+1}(x) = \begin{cases} p_{n+1}(x) - \beta_n, & n = 1, \\ p_{n+1}(x) - \beta_n p_{n-1}(x), & n > 1, \end{cases}$$

where $\{p_j\}_{j=0}^{\infty}$ is the sequence of monic orthogonal polynomials on $[-1, 1]$ with respect to the Jacobi weight (2) and

$$\beta_n = \frac{4n(n+\alpha)(n+\beta)(n+\alpha+\beta)}{(2n+\alpha+\beta)^2((2n+\alpha+\beta)^2-1)}, \qquad n \geq 1.$$

Such zeros are real, distinct, and interlace the zeros $\{x_j\}_{j=1}^{n}$ of p_n, i.e.,

$$\tilde{x}_1 < x_1 < \tilde{x}_2 < \cdots < \tilde{x}_n < x_n < \tilde{x}_{n+1}.$$

This means that $\tilde{x}_j \in (-1, 1)$ for $j = 2, \ldots, n$, while the first and last nodes may not belong to $(-1, 1)$ and, in particular, they may coincide with the endpoints.

Next theorem is a reformulation of [14, Theorem 3.3] and provides sufficient and necessary conditions on the parameters α and β so that the anti-Gauss formula is internal, i.e., $\tilde{x}_j \in [-1, 1]$, $j = 1, \ldots, n+1$.

Theorem 2.1. *The smallest node* $\tilde{x}_1 \in [-1,1]$ *if and only if*

$$(2\beta+1)n^2+(2\beta+1)(\alpha+\beta+1)n+\frac{1}{2}(\beta+1)(\alpha+\beta)(\alpha+\beta+1)\geq 0;$$

the largest node $\tilde{x}_{n+1} \in [-1,1]$ *if and only if*

$$(2\alpha+1)n^2+(2\alpha+1)(\alpha+\beta+1)n+\frac{1}{2}(\alpha+1)(\alpha+\beta)(\alpha+\beta+1)\geq 0.$$

Moreover, the endpoint $\tilde{x}_1=-1$ *is a zero if and only if* $\beta=-1/2$ *and* $\alpha=\pm 1/2$*; the endpoint* $\tilde{x}_{n+1}=1$ *is a zero if and only if* $\alpha=-1/2$ *and* $\beta=\pm 1/2$.

Finally, in the Gegenbauer or ultraspherical case ($\alpha=\beta$)*, the anti-Gauss rule is internal if and only if* $\alpha=\beta\geq -1/2$.

To our knowledge, a closed form for the zeros and coefficients of an anti-Gauss rule is only known if $\alpha=\beta=-1/2$ [3, Theorem 2]

$$\tilde{x}_j=\cos\left((n-j+1)\frac{\pi}{n}\right),\qquad j=1,\ldots,n+1,$$
$$\tilde{\lambda}_j=\begin{cases}\dfrac{\pi}{2n}, & j=1,n+1,\\[2mm] \dfrac{\pi}{n}, & j=2,\ldots,n.\end{cases}$$

In all the other cases, one needs to compute them. Laurie [14] observed that, being $\widetilde{G}_{n+1}$ a Gauss formula, nodes and weights can be found by the algorithm introduced for the first time by H. Wilf in 1962, known as the Golub–Welsh algorithm [10]. More precisely, one has to consider the symmetric tridiagonal matrix $\widetilde{J}_{n+1}$

$$\widetilde{J}_{n+1}=\begin{bmatrix} J_n & \sqrt{2\beta_n}\mathbf{e}_n\\ \sqrt{2\beta_n}\mathbf{e}_n^T & \alpha_n\end{bmatrix},\quad J_n=\begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & & \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & & \\ & \sqrt{\beta_2} & \alpha_2 & \ddots & \\ & & \ddots & \ddots & \sqrt{\beta_{n-1}}\\ & & & \sqrt{\beta_{n-1}} & \alpha_{n-1}\end{bmatrix},$$

with $\mathbf{e}_n=(0,0,\ldots,1)^T\in\mathbb{R}^n$ and

$$\alpha_j=\frac{\beta^2-\alpha^2}{(2j+\alpha+\beta)(2j+\alpha+\beta+2)},\qquad j\geq 0, \tag{11}$$
$$\beta_j=\frac{4j(j+\alpha)(j+\beta)(j+\alpha+\beta)}{(2j+\alpha+\beta)^2((2j+\alpha+\beta)^2-1)},\qquad j\geq 1, \tag{12}$$

and solve the corresponding eigenvalue problem. In fact, the jth zero $\tilde{x}_j$ is the jth eigenvalue of $\widetilde{J}_{n+1}$ and the jth coefficient $\tilde{\lambda}_j$ is proportional to the square of the first component $v_{j,1}$ of the jth eigenvector, i.e.,

$$\lambda_j=\beta_0 v_{j,1}^2,\qquad \beta_0=\frac{2^{\alpha+\beta+1}\Gamma(\alpha+1)\Gamma(\beta+1)}{\Gamma(\alpha+\beta+2)},$$

being Γ the Gamma function. We recall that the computational cost for the construction of each quadrature formula is $O(n^2)$. Therefore, the averaged rule (8) needs $O(2n^2)$ flops, which

is cheaper than the cost of the Gauss rule G_{2n} based on the same number of points, which is $O(4n^2)$.

We conclude this section by noting that the anti-Gauss rule is an interpolatory formula. In fact, let us introduce a polynomial of degree n interpolating the function f at the $n+1$ points $\tilde{x}_j$, with $j = 1,\dots,n+1$,

$$L_{n+1}(f,x) = \sum_{j=1}^{n+1} \ell_j(x) f(\tilde{x}_j), \qquad \ell_j(x) = \prod_{\substack{k=1 \\ k\neq j}}^{n+1} \frac{x-\tilde{x}_k}{\tilde{x}_j-\tilde{x}_k}.$$

Then, since the anti-Gauss rule is a Gauss rule with respect to the linear functional $2I(f) - G_n(f)$ [14, relation (6)], we immediately deduce

$$\sum_{j=1}^{n+1} \tilde{\lambda}_j L_{n+1}(f,\tilde{x}_j) = 2\int_{-1}^{1} L_{n+1}(f,x)w(x)\,dx - \sum_{j=1}^{n} \lambda_j L_{n+1}(f,x_j),$$

that is, by the degree of exactness of Gauss rule,

$$\sum_{j=1}^{n+1} \tilde{\lambda}_j f(\tilde{x}_j) = \sum_{j=1}^{n+1} f(\tilde{x}_j) \int_{-1}^{1} \ell_j(x)w(x)\,dx.$$

This implies the interpolatory property

$$\tilde{\lambda}_j = \int_{-1}^{1} \ell_j(x)w(x)\,dx.$$

2.2 A generalized anti-Gauss rule and a generalized averaged formula

The generalized anti-Gauss formula introduced by Reichel and Spalević [23] has the form

$$G^*_{n+1}(f) = \sum_{k=1}^{n+1} \lambda^*_k f(x^*_k), \tag{13}$$

where $\{\lambda^*_k\}_{k=1}^{n+1}$ are the corresponding coefficients and $\{x^*_k\}_{k=1}^{n+1}$ the quadrature nodes. As mentioned in the Introduction, such a formula is strictly related to the optimal generalized averaged rule Q^*_{2n+1} developed by Spalević in 2007 [24]. The latter involves $2n+1$ nodes, as the averaged rule Q_{2n+1} in (8), but has degree of exactness at least $2n+2$. Therefore, it is more accurate than Q_{2n+1}.

Despite these positive features, the initial formulation of Q^*_{2n+1} was much more expensive than the averaged rule Q_{2n+1}. Indeed, the computation of nodes and weights by the Golub and Welsh algorithm would involve the resolution of an eigenvalue problem of order $2n+1$, requiring $O(4n^2)$ operations. To avoid this drawback, Reichel and Spalević proved in [23] that, similarly to Q_{2n+1}, the optimal formula Q^*_{2n+1} can be expressed as a convex combination of the Gauss rule G_n and the generalized formula G^*_{n+1}, as follows

$$Q^*_{2n+1}(f) = \frac{\beta_{n+1}}{\beta_n+\beta_{n+1}} G_n(f) + \frac{\beta_n}{\beta_n+\beta_{n+1}} G^*_{n+1}(f), \tag{14}$$

where the β_n coefficients are given by (12). By this approach, the computational complexity is reduced and becomes comparable with that of the average formula, i.e., $O(2n^2)$. Moreover, from (14) one obtains an estimate for the Gauss quadrature error as

$$e_n(f) \simeq Q^*_{2n+1}(f) - G_n(f) = \frac{\beta_n}{\beta_n+\beta_{n+1}} [G^*_{n+1}(f) - G_n(f)] =: r^*_n(f). \tag{15}$$

We note that the formula $Q^*_{2n+1}(f)$ may give better results compared to Q_{2n+1} only for small values of n. Indeed, we recall that for any $\alpha,\beta>-1$ equation (12) implies that $\lim_{n\to\infty}\beta_n=\frac{1}{4}$. This means that the coefficients in (14) quickly approach $\frac{1}{2}$ as n increases, making the formula Q^*_{2n+1} substantially equivalent to Q_{2n+1} in applications where the integrand function is not smooth, as the number of quadrature points must be increased to obtain good results; see also the numerical experiments in [7].

Formula (13) has similar properties to the anti-Gauss rule. Firstly, the nodes are real and distinct and coincide with the eigenvalues of the matrix

$$J^*_{n+1}=\begin{bmatrix} J_n & \sqrt{\beta_n+\beta_{n+1}}\mathbf{e}_n \\ \sqrt{\beta_n+\beta_{n+1}}\mathbf{e}_n^T & \alpha_n \end{bmatrix},$$

where β_n and α_n are given in (11)–(12), and the coefficients $\{\lambda^*_k\}_{k=1}^{n+1}$ are proportional to the squared first components of the normalized eigenvectors of J^*_{n+1}. Secondly, the quadrature Gauss points interlace the nodes x^*_k (see also [7, Theorem 2.1]), i.e.,

$$x^*_1<x_1<x^*_2<\cdots<x^*_n<x_n<x^*_{n+1}.$$

Finally, the quadrature error $e^*_{n+1}(f)$ is opposite in sign with respect to the Gauss error $e_n(f)$, according to the following relation

$$e^*_{n+1}(f)=-\frac{\beta_{n+1}}{\beta_n}e_n(f),\quad \text{for all } f\in\mathbb{P}_{2n+2}.$$

3 The 2D case

Here, we consider the approximation of integrals of the type

$$\mathscr{I}(f)=\int_S f(\mathbf{x})w(\mathbf{x})\,d\mathbf{x}=\int_{-1}^1\int_{-1}^1 f(x_1,x_2)w_1(x_1)w_2(x_2)\,dx_1\,dx_2, \tag{16}$$

where $S:=[-1,1]\times[-1,1]$, $\mathbf{x}=(x_1,x_2)$, f is a given function defined on the square and $w(\mathbf{x})=w_1(x_1)w_2(x_2)$, with $w_i(x_i)=(1-x_i)^{\alpha_i}(1+x_i)^{\beta_i}$, $\alpha_i,\beta_i>-1$, for $i=1,2$.

As in the univariate case, it is not restricting to consider the unit square S, since any other rectangular domain $D=[a,b]\times[c,d]$ can be backtracked to S as follows

$$\begin{aligned}&\int_c^d(d-y_2)^{\alpha_2}(y_2-c)^{\beta_2}\int_a^b F(y_1,y_2)(b-y_1)^{\alpha_1}(y_1-a)^{\beta_1}\,dy_1\,dy_2\\ &=\left(\frac{b-a}{2}\right)^{\alpha_1+\beta_1+1}\left(\frac{d-c}{2}\right)^{\alpha_2+\beta_2+1}\int_{-1}^1\int_{-1}^1 f(x_1,x_2)w_1(x_1)w_2(x_2)\,dx_1\,dx_2,\end{aligned}$$

where $f(x_1,x_2)=:F\left(\frac{(b-a)}{2}(x_1+1)+a,\frac{(d-c)}{2}(x_2+1)+c\right)$.

The integral (16) may be approximated by the optimal Gauss cubature rule constructed as a tensor product of two univariate Gauss formulae; see, for instance, [17]. We obtain the $(n_1\times n_2)$-point Gauss cubature rule by using n_1 points in the integral with the differential dx_1 and n_2 nodes in that with dx_2, as follows

$$G_{n_1,n_2}(f)=\sum_{j_1=1}^{n_1}\sum_{j_2=1}^{n_2}\lambda^{(1)}_{j_1}\lambda^{(2)}_{j_2}f(x^{(1)}_{j_1},x^{(2)}_{j_2}). \tag{17}$$

Letting $\ell = 1,2$, $\lambda_j^{(\ell)}$ is the jth Christoffel number with respect to the weight $w_\ell(x)$ appearing in the integral, and $x_j^{(\ell)}$ is the jth zero of the monic polynomial $p_{n_\ell}^{(\ell)}(x)$ orthogonal with respect to the same weight. The corresponding cubature error

$$e_{n_1,n_2}(f) = \mathscr{I}(f) - G_{n_1,n_2}(f) \tag{18}$$

is such that

$$e_{n_1,n_2}(p) = 0, \qquad \forall p \in \mathbb{P}_{2n_1-1,2n_2-1},$$

where $\mathbb{P}_{k,\ell}$ represents the set of all bivariate polynomials with degree at most k in the variable x_1 and at most ℓ in x_2.

In the next subsection, we discuss a stratified formula that allow us to estimate the error (18) by replacing the exact value of the integral by a cubature formula

$$e_{n_1,n_2}(f) \simeq Q_{m_1,m_2}(f) - G_{n_1,n_2}(f). \tag{19}$$

This enables us to determine the number of nodes required to achieve a given precision, avoiding unnecessary evaluations that, especially in the multivariate case, significantly increase computational cost and error propagation.

Before going on with the discussion, for ease of exposition, we introduce a multi-index notation allowing an index to take integer vectorial values. Such indices will be denoted by bold letters, for example, $\mathbf{n} = (n_1, n_2)$. Let us consider the set of bi-indices

$$\mathfrak{I}_{\mathbf{n}} = \{\mathbf{j} = (j_1, j_2) : 1 \le j_1 \le n_1, 1 \le j_2 \le n_2\}.$$

For $\mathbf{j} \in \mathfrak{I}_{\mathbf{n}}$, consistently with the notation $\mathbf{x} = (x_1, x_2)$, we define $\mathbf{x_j} = (x_{j_1}^{(1)}, x_{j_2}^{(2)})$, where $x_{j_\ell}^{(\ell)}$, $\ell = 1,2$, are the previously introduced Gaussian nodes, and $\lambda_{\mathbf{j}} = \lambda_{j_1}^{(1)} \lambda_{j_2}^{(2)}$. With this new formalism, formula (17) and the error (18) can be written, respectively, as

$$G_{\mathbf{n}}(f) = \sum_{\mathbf{j} \in \mathfrak{I}_{\mathbf{n}}} \lambda_{\mathbf{j}} f(\mathbf{x_j}), \qquad e_{\mathbf{n}}(f) = \mathscr{I}(f) - G_{\mathbf{n}}(f). \tag{20}$$

3.1 The anti-Gauss and averaged cubature rule

Differently from the univariate case, not so many cubature rules have been developed to estimate the error for bivariate integrals, as in (19). However, the need to determine the number of nodes necessary to achieve a given accuracy is even more important than in the univariate case, as the number of evaluations of the integrand function is quadratic with respect to the number of nodes.

To fill this gap, in [4] the authors developed an anti-Gauss cubature rule constructed as a tensor product of two anti-Gauss univariate formulae (6). It has the form

$$\widetilde{G}_{\mathbf{n+1}}(f) = \sum_{\mathbf{j} \in \mathfrak{I}_{\mathbf{n+1}}} \tilde{\lambda}_{\mathbf{j}} f(\tilde{\mathbf{x}}_{\mathbf{j}}),$$

where $\mathbf{1} = (1,1)$, $\tilde{\lambda}_{\mathbf{j}} = \tilde{\lambda}_{j_1}^{(1)} \tilde{\lambda}_{j_2}^{(2)}$, and $\tilde{\mathbf{x}}_{\mathbf{j}} = (\tilde{x}_{j_1}^{(1)}, \tilde{x}_{j_2}^{(2)})$, with $\tilde{\lambda}_j^{(\ell)}$ and $\tilde{x}_j^{(\ell)}$ the jth anti-Gauss quadrature weight and node, respectively, for $\ell = 1,2$.

The fundamental property of the above formula is the following [4, Proposition 2.1]

$$\tilde{e}_{\mathbf{n+1}}(f) = -e_{\mathbf{n}}(f), \quad \forall f \in \mathbb{P}_{2n_1+1,2n_2-1} \cup \mathbb{P}_{2n_1-1,2n_2+1}. \tag{21}$$

Indeed, formula (21) has three important consequences. First, the anti-Gauss cubature rule is exact for polynomials $f \in \mathbb{P}_{2n_1-1,2n_2-1}$, like the Gauss cubature formula. Second, Gauss and anti-Gauss rules provide upper and lower bounds for the integral when the integrand function f belongs to the set of polynomials $\mathbb{P}_{2n_1+1,2n_2-1} \cup \mathbb{P}_{2n_1-1,2n_2+1}$, i.e.,

$$\widetilde{G}_{\mathbf{n}+\mathbf{1}}(f) \leq \mathscr{I}(f) \leq G_{\mathbf{n}}(f) \quad \text{or} \quad G_{\mathbf{n}}(f) \leq \mathscr{I}(f) \leq \widetilde{G}_{\mathbf{n}+\mathbf{1}}(f). \tag{22}$$

Third, from (21) it follows

$$\mathscr{I}(f) = \frac{1}{2}[G_{\mathbf{n}}(f) + \widetilde{G}_{\mathbf{n}+\mathbf{1}}(f)] =: Q_{2\mathbf{n}+\mathbf{1}}(f), \qquad \forall f \in \mathbb{P}_{2n_1+1,2n_2-1} \cup \mathbb{P}_{2n_1-1,2n_2+1},$$

that is, we obtain the *averaged cubature formula*, $Q_{2\mathbf{n}+\mathbf{1}}(f)$.

The advantages of such new rule, classified as a stratified nested formula, are multiple. It is more accurate than the two single cubature formulae and its nodes and weights can be computed in $O(2n_\ell^2)$ flops, which is cheaper than the $O(4n_\ell^2)$ flops required by the Gauss cubature formula $G_{2\mathbf{n}}$. Similarly to the univariate case, numerical experiments show that the *bracketing* (22) is generally verified also when the integrand function is not a polynomial. A first result to identify the class of functions for which *bracketing* occurs is given in [4, Theorem 2.2, Corollary 2.3, and Corollary 2.4], but a theoretical analysis that agrees with the numerical experiments has not yet been developed; see the numerical experiments in [4, Example 1 and Example 2].

The Gauss cubature error can be estimated by the averaged rule as follows

$$e_{\mathbf{n}}(f) \simeq Q_{2\mathbf{n}+\mathbf{1}}(f) - G_{\mathbf{n}}(f) = \frac{1}{2}\left[\widetilde{G}_{\mathbf{n}+\mathbf{1}}(f) - G_{\mathbf{n}}(f)\right] =: r_{\mathbf{n}}(f). \tag{23}$$

4 The AGquad Matlab package

To ascertain the effectiveness of the different approximation formulae introduced throughout the paper, here we report the results of numerical examples in both the 1D and 2D cases. At the same time, we explain how to use the Matlab package *AGquad* which implements the algorithms discussed in the previous sections.

The package includes 12 Matlab functions, a graphical user interface (GUI), two test scripts, and some auxiliary files, including the file README.txt that provides all the necessary information for the user regarding the installation procedure. The two test scripts illustrate how to use the computational routines, obtain results, and draw conclusions. All the functions, including the test scripts, run both in Matlab and in Octave. The graphical user interface, on the contrary, can only be used in Matlab.

The routines are organized by groups in Table 1, where a brief overview provides basic information about each of them. The first category, "Computational routines," includes functions for calculating nodes and weights of the various formulae, and compute quadrature and cubature rules. This category also includes the graphical user interface AGquadGUI. The "Test scripts" are intended for testing both 1D and 2D integral approximations and illustrate the calling syntax for the various functions. The sets, "Auxiliary routines" and "Auxiliary files," list additional routines that are necessary to complete the entire process, but are unlikely to be called directly by the user. For further details, we refer to the Contents.m file.

The graphical interface AGquadGUI, created with Matlab AppDesigner (see Figure 1) allows to run interactively some numerical examples, and may be useful to familiarize the user with the package. To open it, the user simply has to issue the command `AGquadGUI` in Matlab command window. For tasks more complicated than the computation of a single integral, the

Computational routines	
`nodesweights`	compute nodes and weights for Gauss, anti-Gauss and G^* rules
`quadG1D`	Gauss quadrature for an integral on $[-1,1]$
`quadGS1D`	G^* quadrature for an integral on $[-1,1]$
`quadG2D`	Gauss cubature for an integral on $[-1,1]\times[-1,1]$
`quadantiG1D`	anti-Gauss quadrature for an integral on $[-1,1]$
`quadantiG2D`	anti-Gauss cubature for an integral on $[-1,1]\times[-1,1]$
`AGquadGUI`	GUI for 1D and 2D integral approximation
Auxiliary routines	
`agintnodes`	check if anti-Gauss nodes are internal to $[-1,1]$
`fun1D`	evaluate the test function f at points x
`fun2D`	evaluate the test function f at points x and y
`guiplot1D`	plot the 1D integrand function
`guiplot2D`	plot the 2D integrand function
`guiprint`	auxiliary output routine
Test scripts	
`test1D`	test script for 1D integral approximation
`test2D`	test script for 2D integral approximation
Auxiliary files	
README.txt	general information about the package
Contents.m	documentation file typical of Matlab
information.pdf	file displayed by AGquadGUI
`approxint1D`	auxiliary function for 1D integral approximation
`approxint2D`	auxiliary function for 2D integral approximation

Table 1: Routines of AGquad package.

user is encouraged to read the scripts `test1D` and `test2D`, which show how to use the functions included in the package.

The software is distributed as a compressed archive file and can be downloaded from GitHub at https://github.com/CaNAgroup/AGquad, or from the web page https://bugs.unica.it/cana/software.html. By uncompressing it, a new directory containing the toolbox functions will be created. Such directory should be added to the Matlab search path to use the package from any other directory.

4.1 Computing 1D integrals

The following two examples provide a practical illustration of the theoretical properties of the quadrature rules discussed in Section 2. In both cases, we take the value $G_{512}(f)$, obtained from the Gauss quadrature rule (1) with $n = 512$, as the reference solution, since the exact value $I(f)$ of the integral defined in (5) is not known.

The tables display the quadrature errors $e_n(f)$, $\tilde{e}_{n+1}(f)$, $e^*_{n+1}(f)$ computed as

$$e_n(f) = I(f) - G_n(f), \quad \tilde{e}_{n+1}(f) = I(f) - \widetilde{G}_{n+1}(f), \quad e^*_{n+1}(f) = I(f) - G^*_{n+1}(f),$$

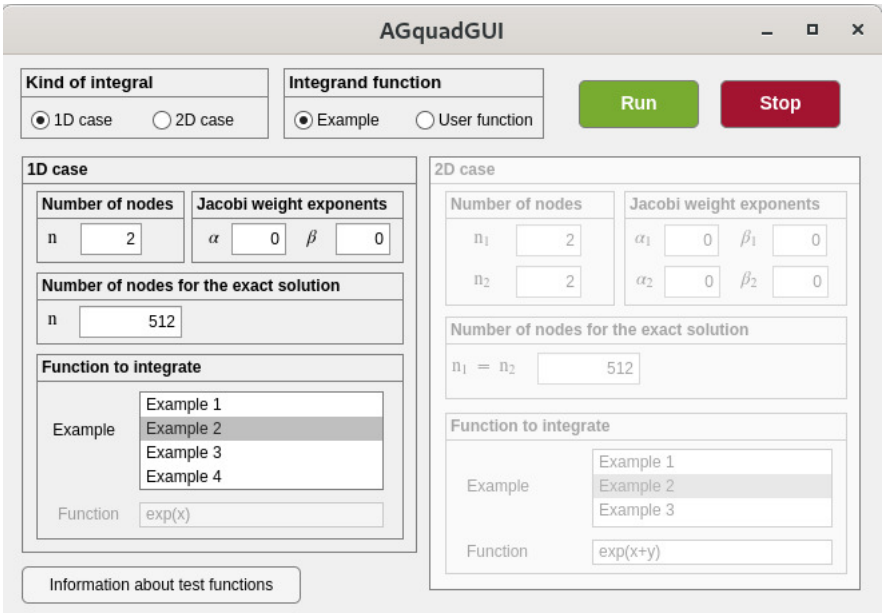

Figure 1: AGquadGUI.

as well as the errors

$$\tilde{\xi}_n(f) = I(f) - Q_{2n+1}(f), \quad \xi_n^*(f) = I(f) - Q_{2n+1}^*(f),$$

corresponding to the averaged (8) and the generalized averaged (14) formulae, respectively.

Moreover, the last two columns show the estimates for the Gauss error $r_n(f)$ and $r_n^*(f)$, defined in (9) and (15), respectively.

Example 4.1. Let us approximate the value of the integral

$$\int_{-1}^{1} |\sin(1-x)|^{\frac{9}{2}} w(x)\,dx,$$

where the weight function $w(x)$ is defined in (2) with $\alpha = \beta = 0$. In this case, the integrand function has continuous derivatives up to the fourth order, which means that it is not necessary to significantly increase the number of quadrature nodes to accurately approximate the integral.

The results are reported in Table 2. In columns 2–4 we can observe the quadrature errors produced by G_n, $\widetilde{G}_{n+1}$, and G_{n+1}^*, for increasing values of n. As expected, the $\widetilde{G}_{n+1}$ and G_{n+1}^* rules give errors of the same magnitude but opposite in sign, compared to the one of the Gauss formula G_n. This implies that Q_{2n+1} and Q_{2n+1}^* can achieve a better approximation of the integral; see the last two columns. The fifth and sixth columns, on the other hand, demonstrate that the values r_n and r_n^* provided by $\widetilde{G}_{n+1}$ and G_{n+1}^*, respectively, are good estimates for the Gauss error.

To reproduce these results, the user can run the Matlab script `test1D.m`, included in the AGquad package, by setting inside the script:

`examp=2;`	at line 9
`alpha=0;`	at line 17
`beta=0;`	at line 18

By doing this, two tables will be displayed in the command window: the first one, contains the values computed by the different approximation formulae G_n, $\widetilde{G}_{n+1}$, G_{n+1}^*, Q_{2n+1}, and Q_{2n+1}^*, for an increasing number of nodes n. The second one corresponds to Table 2.

n	$e_n(f)$	$\tilde{e}_{n+1}(f)$	$e^*_{n+1}(f)$	$r_n(f)$	$r^*_n(f)$	$\tilde{\xi}_n(f)$	$\xi^*_n(f)$
2	-8.31e-02	+8.35e-02	+8.17e-02	-8.33e-02	-8.39e-02	+1.59e-04	+7.87e-04
4	-2.14e-03	+2.14e-03	+2.13e-03	-2.14e-03	-2.14e-03	-6.05e-07	+1.16e-07
8	-1.42e-08	+1.42e-08	+1.42e-08	-1.42e-08	-1.42e-08	+1.67e-11	+1.28e-11
16	-6.41e-12	+6.43e-12	+6.42e-12	-6.42e-12	-6.42e-12	+9.66e-15	+8.66e-15
32	-4.00e-15	+2.33e-15	+3.00e-15	-3.16e-15	-3.50e-15	-7.77e-16	-5.55e-16

Table 2: Errors and estimates for Example 4.1

The same can be done by using AGquadGUI (see Figure 1), by setting the parameters corresponding to Example 4.1, i.e.:

- set the values of n, α, and β;
- select '1D case' in the 'Kind of integral' panel;
- select 'Example' in the 'Integrand function' panel;
- choose 'Example 2' in the 'Function to integrate' panel.

Then, by clicking 'Run', the numerical results and the plot of the integrand function will be displayed in separated windows.

Example 4.2. Let us now consider the following integral

$$\int_{-1}^{1} |x-1|^{\frac{3}{2}} \sin(x) w(x)\, dx,$$

where $\alpha = \frac{1}{2}$ and $\beta = -\frac{1}{2}$ in the Jacobi weight $w(x)$. In this case, the integrand is less smooth than the previous one, possessing just one continuous derivative. Therefore, we need to increase the number of quadrature nodes to obtain an accurate approximation.

n	$e_n(f)$	$\tilde{e}_{n+1}(f)$	$e^*_{n+1}(f)$	$r_n(f)$	$r^*_n(f)$	$\tilde{\xi}_n(f)$	$\xi^*_n(f)$
2	+6.26e-02	-6.25e-02	-6.25e-02	+6.26e-02	+6.26e-02	+4.32e-05	+4.32e-05
4	+8.96e-05	-8.74e-05	-8.74e-05	+8.85e-05	+8.85e-05	+1.14e-06	+1.14e-06
8	+1.62e-06	-1.57e-06	-1.57e-06	+1.59e-06	+1.59e-06	+2.40e-08	+2.40e-08
16	+2.87e-08	-2.78e-08	-2.78e-08	+2.82e-08	+2.82e-08	+4.42e-10	+4.42e-10
32	+4.84e-10	-4.69e-10	-4.69e-10	+4.77e-10	+4.77e-10	+7.54e-12	+7.54e-12
64	+7.90e-12	-7.64e-12	-7.64e-12	+7.77e-12	+7.77e-12	+1.28e-13	+1.28e-13
128	+1.30e-13	-1.17e-13	-1.17e-13	+1.23e-13	+1.23e-13	+6.66e-15	+6.66e-15
256	+5.33e-15	+2.22e-15	+2.22e-15	+1.55e-15	+1.55e-15	+4.00e-15	+4.00e-15

Table 3: Errors and estimates for Example 4.2

In Table 3, we observe again that the errors $\tilde{e}_{n+1}$ and e^*_{n+1} are equal in magnitude but opposite in sign with respect to e_n. Thus, the two averaged rules provide more accurate approximations of the integral; see $\tilde{\xi}_n$ and ξ^*_n. From the fifth and sixth columns, we confirm that the alternative estimates for the Gauss error are indeed accurate.

We point out that, as mentioned in Section 2.1, the computational cost for the construction of the averaged rule (8) is smaller than the Gauss rule G_{2n}. For example, for $n = 8$ the error produced by Q_{2n+1} is of the same order of magnitude than the one given by G_{16}. The first one, requires only about 128 flops, while the second needs approximately 256. The same happens with the generalized averaged formula Q^*_{2n+1}.

To reproduce the results displayed in Table 3 by the script `test1D.m`, one should set `examp=4` and the correct values of α and β in the script, before running it. The same can be done using AGquadGUI.

Remark 4.1. The number of nodes for the exact solution can be changed when necessary, either in the script `test1D.m` (line 21) or in the corresponding box of AGquadGUI.

Remark 4.2. For more details about the test integrand functions included in the package, it is also possible to click on the 'Information about test functions' button in AGquadGUI.

4.2 Computing 2D integrals

Here, we present two numerical tests concerning the cubature rules introduced in Section 3. Also in this case, the exact value of the integral $\mathscr{I}(f)$ in (16), is obtained by $G_{512,512}(f)$, i.e., the Gauss cubature rule (17) with $n_1 = n_2 = 512$.

The tables report the cubature errors $e_{\mathbf{n}}(f)$, defined in (20), and

$$\tilde{e}_{\mathbf{n}+1}(f) = \mathscr{I}(f) - \widetilde{G}_{\mathbf{n}+1}(f),$$

along with the error produced by the averaged formula from Section 3.1, defined as

$$\tilde{\xi}_{\mathbf{n}}(f) = \mathscr{I}(f) - Q_{2\mathbf{n}+1}(f).$$

Additionally, the final column reports the estimate $r_{\mathbf{n}}(f)$ for the Gauss error, defined in (23).

Example 4.3. In this example, we approximate the integral

$$\int_{-1}^{1}\int_{-1}^{1}\left[x_1\left|\cos\left(\frac{1}{2}-x_1\right)\right|^{\frac{3}{2}}+x_2|\sin(1+x_2)|^{\frac{3}{2}}\right]w_1(x_1)w_2(x_2)\,dx_1\,dx_2,$$

where $\alpha_i = \beta_i = 0$, for $i = 1,2$, in the weight function $w(\mathbf{x})$ appearing in (16).

In this case, the integrand function exhibits low smoothness with respect to both variables. Therefore, to achieve an accurate approximation it is necessary to increase both n_1 and n_2.

In Table 4, the advantage of the averaged rule in terms of accuracy compared to the Gauss scheme is clear; see third, fourth and sixth columns. Regarding the computational cost, we note that if we set $n_1 = n_2 = 8$, the averaged rule produces an error of order 10^{-7} with only $n_1n_2 + (n_1+1)(n_2+1) = 97$ function evaluations. In contrast, the same error order is achieved by the Gaussian formula with $n_1 = n_2 = 16$, requiring $n_1n_2 = 256$ function evaluations, which is more than two times more. Finally, the fifth column confirms the good performance of the estimate $r_{\mathbf{n}}(f)$ for Gauss error.

The results in Table 4 can be reproduced by running the script `test2D.m` by setting in the code

`examp=2;`	at line 9
`alpha1=0;`	at line 21
`beta1=0;`	at line 22
`alpha2=0;`	at line 23
`beta2=0;`	at line 24

n_1	n_2	$e_{\mathbf{n}}(f)$	$\tilde{e}_{\mathbf{n}+1}(f)$	$r_{\mathbf{n}}(f)$	$\tilde{\xi}_{\mathbf{n}}(f)$
2	2	-2.03e-01	+2.03e-01	-2.03e-01	+3.70e-05
4	4	-1.02e-03	+1.03e-03	-1.02e-03	+5.66e-06
8	8	-2.43e-05	+2.46e-05	-2.45e-05	+1.49e-07
16	16	-7.79e-07	+7.89e-07	-7.84e-07	+5.03e-09
32	32	-2.58e-08	+2.62e-08	-2.60e-08	+1.67e-10
64	64	-8.36e-10	+8.47e-10	-8.41e-10	+5.43e-12
128	128	-2.66e-11	+2.70e-11	-2.68e-11	+1.97e-13
256	256	-8.13e-13	+8.79e-13	-8.46e-13	+3.31e-14

Table 4: Errors and estimates for Example 4.3

The script will print a table with the approximation values obtained by the different cubature formulae and Table 4.

Using AGquadGUI is similar to the previous examples. The user has just to select the '2D case' in the 'Kind of integral' panel and proceed as before.

Example 4.4. In the last example, we approximate the integral

$$\int_{-1}^{1}\int_{-1}^{1} e^{(1+x_1+x_2)}|x_1-1|^{\frac{7}{2}}\, w_1(x_1)w_2(x_2)\,dx_1\,dx_2,$$

where $\alpha_1=\beta_1=\frac{1}{2}$ and $\alpha_2=\beta_2=-\frac{1}{2}$. The integrand function has a low degree of smoothness with respect to the first variable, while it is smooth with respect to the second one. Thus, we let the number of cubature nodes n_1 increase and set $n_2=8$; see Table 5.

n_1	n_2	$e_{\mathbf{n}}(f)$	$\tilde{e}_{\mathbf{n}+1}(f)$	$r_{\mathbf{n}}(f)$	$\tilde{\xi}_{\mathbf{n}}(f)$
2	8	-3.16e-01	+3.16e-01	-3.16e-01	+1.02e-04
4	8	+1.18e-03	-1.17e-03	+1.17e-03	+2.58e-07
8	8	+8.08e-07	-8.07e-07	+8.08e-07	+5.69e-10
16	8	+1.02e-09	-1.02e-09	+1.02e-09	+9.02e-13
32	8	+1.22e-12	-1.25e-12	+1.23e-12	-1.42e-14
64	8	-1.42e-14	-7.11e-15	-3.55e-15	-1.07e-14

Table 5: Errors and estimates for Example 4.4

Also in this example, the averaged rule is significantly more accurate than the Gauss scheme, the errors $e_{\mathbf{n}}(f)$ and $\tilde{e}_{\mathbf{n}+1}(f)$ change sign and have roughly the same absolute value, and the Gauss error is well approximated by $r_{\mathbf{n}}(f)$.

The construction of Table 5 can be repeated either by the script `test2D.m` or using AGquadGUI by selecting `examp=3` and setting the other relevant parameters.

We finally point out that Remarks 4.1 and 4.2 are valid also in the 2D case. Regarding Remark 4.1, note that the number of nodes for the exact solution has to be changed at lines 27 and 28 of `test2D.m`.

5 Conclusions

In this paper, we present a new Matlab package, *AGquad*, for approximating univariate and bivariate integrals by pairs of Gauss and anti-Gauss rules and the corresponding averaged formulae. We demonstrate the performance of the quadrature and cubature rules through various numerical tests, and describe how to use the package to replicate the results, either by a Matlab script or using a graphical user interface included in the software package.

Acknowledgments

The authors are members of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM). P. Díaz de Alba is partially supported by the INdAM-GNCS project 2025 "Tecniche numeriche per problemi di grandi dimensioni" (CUP E53C24001950001) and by the PRIN 2022 PNRR project no. P2022PMEN2 financed by the European Union-NextGenerationEU and by the Italian Ministry of University and Research (MUR). L. Fermo and G. Rodriguez are partially supported by the INdAM-GNCS project 2025 "Metodi di approssimazione globale per operatori integrali e applicazioni alle equazioni funzionali" (CUP E53C24001950001), the PRIN 2022 project no. 2022ANC8HL financed by the European Union - NextGeneration EU, Missione 4 Componente 1 CUP F53D23002700006, and by the PRIN 2022 PNRR project no. P20229RMLB financed by the European Union - NextGeneration EU and by the Italian Ministry of University and Research (MUR).

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C. W. Clenshaw and A. R. Curtis. *A method for numerical integration on an automatic computer*. Numer. Math., vol. 2, (1960), 197–205. URL http://dx.doi.org/10.1007/BF01386223

[2] V. Devyatko. *On a two-sided approximation for the numerical integration of ordinary differential equations*. USSR Computational Mathematics and Mathematical Physics, vol. 3, 2, (1963), 336–350. URL http://dx.doi.org/10.1016/0041-5553(63)90024-7

[3] P. Díaz de Alba, L. Fermo, and G. Rodriguez. *Solution of second kind Fredholm integral equations by means of Gauss and anti-Gauss quadrature rules*. Numer. Math., vol. 146, (2020), 699–728. URL http://dx.doi.org/10.1007/s00211-020-01163-7

[4] P. Díaz de Alba, L. Fermo, and G. Rodriguez. *Anti-Gauss cubature rules with applications to Fredholm integral equations on the square*. SIAM J. Sci. Comput., vol. 47, 2, (2025), A689–A712. URL http://dx.doi.org/10.1137/24M1631432

[5] D. L. Djukic, L. Fermo, and R. Mutavdzic Djukic. *Averaged Nyström interpolants for bivariate Fredholm integral equations on the real positive semi-axes*. Electron. Trans. Numer. Anal., vol. 61, (2024), 51–65. URL http://dx.doi.org/10.1553/etna_vol61s51

[6] D. L. Djukić, L. Reichel, M. M. Spalević, and J. D. Tomanović. *Internality of generalized averaged Gauss rules and their truncations for Bernstein-Szegö weights*. Electron. Trans. Numer. Anal., vol. 45, (2016), 405–419. URL http://dx.doi.org/10.1007/s11075-022-01385-w

[7] L. Fermo, L. Reichel, G. Rodriguez, and M. M. Spalević. *Averaged Nyström interpolants for the solution of Fredholm integral equations of the second kind*. Appl. Math. Comput., vol. 467, (2024), 128482. URL http://dx.doi.org/10.1016/j.amc.2023.128482

[8] C. F. Gauss. *Methodus nova integralium valores per approximationem inveniendi*. Comm. Soc. R. Sci. Göttingen Recens., vol. 3, (1814), 39–76. Werke **3**, 163–196, (1866), URL http://dx.doi.org/10.1017/CBO9781139058247.008

[9] W. Gautschi. *Orthogonal polynomials. Computation and Approximation*. Numerical Mathematics and Scientific Computation. Oxford: Oxford University Press. URL http://dx.doi.org/10.1093/oso/9780198506720.001.0001

[10] G. Golub and J. H. Welsch. *Calculation of Gauss quadrature rules*. Math. Comp., vol. 23, (1969), 221–230. URL http://dx.doi.org/10.2307/2004418

[11] A. S. Kronrod. *Integration with control of accuracy*. *Soviet Physics Doklady*, vol. 9. vol. 9, 17. URL https://www.mathnet.ru/eng/dan29027

[12] A. S. Kronrod. *Nodes and weights for quadrature formulae. Sixteen-place tables (Russian), Izdat* (1964)

[13] D. P. Laurie. *Stratified sequences of nested quadrature formulas*. Quaestiones Mathematicae, vol. 15, 3, (1992), 365–384. URL http://dx.doi.org/10.1080/16073606.1992.9631697

[14] D. P. Laurie. *Anti-Gaussian quadrature formulas*. Math. Comp., vol. 65, (1996), 739–747. URL http://dx.doi.org/10.1090/S0025-5718-96-00713-2

[15] S. E. Notaris. *Gauss–Kronrod quadrature formulae–a survey of fifty years of research*. Electron. Trans. Numer. Anal, vol. 45, (2016), 371–404. URL https://elibm.org/article/10006399

[16] S. E. Notaris. *Stieltjes polynomials and related quadrature formulae for a class of weight functions, II*. Numer. Math., vol. 142, (2019), 129–147. URL http://dx.doi.org/10.1007/s00211-018-1009-8

[17] D. Occorsio and M. G. Russo. *Numerical methods for Fredholm integral equations on the square*. Appl. Math. Comput., vol. 218, 5, (2011), 2318–2333. URL http://dx.doi.org/10.1016/j.amc.2011.07.053

[18] T. Patterson. *Stratified nested and related quadrature rules*. J. Comput. Appl. Math., vol. 112, 1, (1999), 243–251. URL http://dx.doi.org/10.1016/S0377-0427(99)00224-1

[19] F. Peherstorfer and K. Petras. *Ultraspherical Gauss–Kronrod Quadrature Is Not Possible for $\lambda > 3$*. SIAM J. Numer. Anal., vol. 37, 3, (2000), 927–948. URL http://dx.doi.org/10.1137/S0036142998327744

[20] F. Peherstorfer and K. Petras. *Stieltjes polynomials and Gauss-Kronrod quadrature for Jacobi weight functions*. Numer. Math., vol. 95, 4, (2003), 689–706. URL http://dx.doi.org/10.1007/s00211-002-0412-2

[21] M. S. Pranić and L. Reichel. *Generalized anti-Gauss quadrature rules*. J. Comput. Appl. Math., vol. 284, (2015), 235–243. URL http://dx.doi.org/10.1016/j.cam.2014.11.016

[22] J. V. Rakitskii. *Some properties of the solutions of systems of ordinary differential equations by one-step methods of numerical integration*. USSR Computational Mathematics and Mathematical Physics, vol. 1, 4, (1962), 1113–1128. URL http://dx.doi.org/10.1016/0041-5553(62)90035-6

[23] L. Reichel and M. M. Spalević. *A new representation of generalized averaged Gauss quadrature rules*. Appl. Numer. Math., vol. 165, (2021), 614–619. URL http://dx.doi.org/10.1016/j.apnum.2020.11.016

[24] M. M. Spalević. *On generalized averaged Gaussian formulas*. Math. Comp., vol. 76, (2007), 1483–1492. URL http://dx.doi.org/10.1090/S0025-5718-07-01975-8

[25] M. M. Spalević. *On generalized averaged Gaussian formulas. II*. Math. Comp., vol. 86, (2017), 1877–1885. URL http://dx.doi.org/10.1090/mcom/3225

[26] M. M. Spalević. *A note on generalized averaged Gaussian formulas for a class of weight functions*. Numer. Algorithms, vol. 85, 3, (2020), 977–993. URL http://dx.doi.org/10.1007/s11075-019-00848-x