Dolomites Research Notes on Approximation

Volume 5 · 2012 · Pages 7–12

# Algebraic cubature on planar lenses and bubbles

Gaspare Da Fies [a] · Marco Vianello [a]

**Abstract**

By a recent result on subperiodic trigonometric Gaussian quadrature, we construct a cubature formula of algebraic degree of exactness $n$ on planar circular lenses (intersection of two overlapping disks) and "double bubbles" (union of two overlapping disks), with $n^2/2 + \mathcal{O}(n)$ nodes. An application is shown to RBF projection methods.

**2000 AMS subject classification:** 65D32.

**Keywords:** algebraic cubature, subperiodic trigonometric quadrature, product Gaussian quadrature, circular segments, planar lenses, planar double bubbles.

## 1 Introduction

In a recent paper, we have obtained a "subperiodic" trigonometric Gaussian formula, that is a quadrature formula with $n+1$ nodes (angles) and positive weights, exact on

$$\mathbb{T}_n([-\omega,\omega]) = \text{span}\{1, \cos(k\theta), \sin(k\theta),\ 1 \le k \le n\ ,\ \ \theta \in [-\omega,\omega]\}\ , \tag{1}$$

where $0 < \omega \le \pi$; cf. [5, 6]. It is related by a simple nonlinear transformation to an algebraic Gaussian formula on $(-1,1)$, and can be effectively implemented in Matlab by Gautschi's OPQ suite [11]. For the reader's convenience, we report the main result of [5]:

*Proposition* 1. Let $\{(\xi_j, \lambda_j)\}_{1\le j\le n+1}$, be the nodes and positive weights of the algebraic Gaussian quadrature formula for the weight function

$$w(x) = \frac{2\sin(\omega/2)}{\sqrt{1-\sin^2(\omega/2)\,x^2}}\ ,\ \ x \in (-1,1)\ . \tag{2}$$

Then

$$\int_{-\omega}^{\omega} f(\theta)\,d\theta = \sum_{j=1}^{n+1} \lambda_j f(\theta_j)\ ,\ \ \forall f \in \mathbb{T}_n([-\omega,\omega])\ ,\ \ 0 < \omega \le \pi \tag{3}$$

where

$$\theta_j = 2\arcsin(\sin(\omega/2)\xi_j) \in (-\omega,\omega)\ ,\ \ j = 1,2,\dots,n+1\ .$$

Exactness of (3) has been proved in [5] by symmetry based arguments, via the subperiodic trigonometric Lagrange basis studied in [3]. We give here an alternative proof using the canonical trigonometric basis (1).

*Proof of Proposition 1*. Assume that the Gaussian nodes be in increasing order, $-1 < \xi_1 < \xi_2 \cdots < \xi_{n+1} < 1$. It is well known that, the weight function (2) being even, such nodes are symmetric, namely $\xi_j = -\xi_{n+2-j}$ (cf. [10, Ch.1]), and that $\lambda_j = \lambda_{n+2-j}$ since the corresponding Lagrange polynomials satisfy $l_j(x) = l_{n+2-j}(-x)$.

Let us rename for convenience the nodes, $\eta_i = \xi_j$, $i = j - \lfloor n/2 \rfloor - 1$, so that $\eta_i = -\eta_{-i}$, and the corresponding weights, say $u_i$, satisfy $u_i = u_{-i}$, $-\lfloor n/2 \rfloor \le i \le \lfloor n/2 \rfloor$. Correspondingly, we rename the angles $\phi_i = \theta_j$, $i = j - \lfloor n/2 \rfloor - 1$, so that $\phi_i = -\phi_{-i}$, $-\lfloor n/2 \rfloor \le i \le \lfloor n/2 \rfloor$.

On one hand,

$$\int_{-\omega}^{\omega} \sin(k\theta)\,d\theta = 0 = \sum_{i=-\lfloor n/2 \rfloor}^{\lfloor n/2 \rfloor} u_i \sin(k\phi_i) = \sum_{j=1}^{n+1} \lambda_j \sin(k\theta_j)\ ,\ \ k = 1,\dots,n\ ,$$

since $\sin(k\theta)$ is an odd function of $\theta$ (and thus also $u_i \sin(k\phi_i)$ is an odd function of the index $i$).

On the other hand, setting $\alpha = \sin(\omega/2)$, by the change of variables $\theta = \theta(x) = 2\arcsin(\alpha x)$, $x \in [-1,1]$,

$$\int_{-\omega}^{\omega} \cos(k\theta)\,d\theta = \int_{-1}^{1} \cos(2k\arcsin(\alpha x))\,w(x)\,dx\ .$$

[a] Department of Mathematics, University of Padua, Italy

Observing that by standard trigonometric identities

$$\cos(2k\arcsin(\alpha x)) = \cos\left(2k\left(\frac{\pi}{2} - \arccos(\alpha x)\right)\right) = \cos(k\pi)\cos(2k\arccos(\alpha x))$$
$$+\sin(k\pi)\sin(2k\arccos(\alpha x)) = (-1)^k T_{2k}(\alpha x)\,,$$

we get

$$\int_{-\omega}^{\omega} \cos(k\theta)\,d\theta = \int_{-1}^{1} (-1)^k T_{2k}(\alpha x)w(x)\,dx = \sum_{j=1}^{n+1} \lambda_j (-1)^k T_{2k}(\alpha\xi_j) =$$
$$= \sum_{j=1}^{n+1} \lambda_j \cos(2k\arcsin(\alpha\xi_j)) = \sum_{j=1}^{n+1} \lambda_j \cos(k\theta_j)\,,\ \ k = 0,\dots,n\,.$$

Since the formula is exact on the canonical trigonometric basis (1), it is exact for every $f \in \mathbb{T}_n([-\omega,\omega])$. □

The trigonometric Gaussian formula (3) has then been applied in [5] to the construction of product Gaussian formulas with polynomial exactness on relevant sections of the disk, such as circular (annular) sectors, zones and symmetric lenses. All the formulas have $n^2/2 + \mathcal{O}(n)$ nodes at exactness degree $n$. Indeed, while several cubature formulas are known for the disk (cf. [4]) and for special disk sections (such as e.g. complete annuli, cf. [12]), they seem to be missing in the literature in general cases.

In particular, algebraic cubature formulas on arbitrary asymmetric *circular lenses*, i.e., intersections of two disks of possibly different radii, are not available. In this note, using (3) we construct a product Gaussian formula on *circular segments* (one of the two portions of a disk cut by a chord), with $n^2/4 + \mathcal{O}(n)$ nodes at exactness degree $n$, from which a cubature formula on general lenses, which are union of two circular segments, is immediately derived. Similarly, we give a cubature formula for planar *"double bubbles"* (union of two overlapping disks), which are also union of two circular segments. We provide two Matlab functions, named `gqlens` and `gqdbubble`, that compute nodes and weights for such formulas given the disks centers and radii, cf. [7].

Finally, we show an application to the numerical integration of products of compactly supported functions, a problem which arises for example within RBF projection methods, such as continuous least squares approximation or Galerkin-like meshfree methods; cf., e.g., [9] and references therein.

## 2 Circular segments

Consider a circular segment, i.e. one of the two portions of a disk of radius $R$ cut by a chord. Up to translation and rotation, it can be written as

$$S = \{(x,y) = (R\cos(\theta), Rt\sin(\theta),\ \theta \in [0,\omega],\ t \in [-1,1]\}\,, \tag{4}$$

where $0 < \omega \le \pi$. Our main result is the following

*Proposition* 2. For any circular segment $S$ like (4) the following product Gaussian formula holds

$$\iint_S f(x,y)\,dx\,dy = \sum_{j=1}^{\lceil\frac{n+2}{2}\rceil} \sum_{i=1}^{\lceil\frac{n+1}{2}\rceil} W_{ij} f(x_{ij}, y_{ij})\,,\ \ \forall f \in \mathbb{P}_n^2\,, \tag{5}$$

where $\mathbb{P}_n^2$ denotes the space of bivariate polynomials of total degree not greater than $n$, and

$$W_{ij} = R^2 \sin^2(\theta_j) w_i^{GL} \lambda_j\,,\ \ (x_{ij}, y_{ij}) = (R\cos(\theta_j), Rt_i^{GL}\sin(\theta_j))\,, \tag{6}$$

$\{(\theta_j, \lambda_j)\}$ being the angles and weights of the trigonometric Gaussian formula (3) of degree of exactness $n+2$ on $[-\omega,\omega]$, and $\{(t_i^{GL}, w_i^{GL})\}$ the nodes and weights of the Gauss-Legendre formula of degree of exactness $n$ on $[-1,1]$.

*Proof*. By the transformation (4), whose Jacobian is $|J| = R^2\sin^2(\theta)$, we can write

$$\iint_S f(x,y)\,dx\,dy = \int_{-1}^{1}\int_{0}^{\omega} f(R\cos(\theta), Rt\sin(\theta))R^2\sin^2(\theta)\,d\theta\,dt\,.$$

On the other hand, by symmetry the circular segment can be also represented as $S = \{(x,y) = (R\cos(\theta), Rt\sin(\theta)),\ \theta \in [-\omega, 0],\ t \in [-1,1]\}$, where the transformation has the same Jacobian $|J| = R^2\sin^2(\theta)$, thus we have also

$$\iint_S f(x,y)\,dx\,dy = \int_{-1}^{1}\int_{-\omega}^{0} f(R\cos(\theta), Rt\sin(\theta))R^2\sin^2(\theta)\,d\theta\,dt\,,$$

and hence

$$\iint_S f(x,y)\,dx\,dy = \frac{1}{2}\int_{-1}^{1}\int_{-\omega}^{\omega} f(R\cos(\theta), Rt\sin(\theta))R^2\sin^2(\theta)\,d\theta\,dt\,.$$

Now, the integrand is a mixed algebraic-trigonometric polynomial belonging to the tensor-product space $\mathbb{P}_n([-1,1]) \bigotimes \mathbb{T}_{n+2}([-\omega,\omega])$, so that we get exactness using the corresponding trigonometric Gaussian formula (3) in the variable $\theta$ and the classical Gauss-Legendre formula in the variable $t$. Observing that each node is repeated twice, apart from the node $(R,0)$ for $n$ even, which corresponds to $j = 1 + (n+2)/2$ and is repeated $2(n+1)$ times but has null weight, we obtain (5)-(6). □

Observe that formula (5)-(6) has approximately $(n+1)(n+2)/4 = n^2/4 + \mathcal{O}(n)$ nodes, and thus is an improvement with respect to the formula provided in [5], which is obtained considering a circular segment as a special case of circular zone and has $n^2/2 + \mathcal{O}(n)$ nodes.

Notice also that for $\omega = \pi$, we obtain an apparently new algebraic cubature formula for the whole disk, as a special instance of circular segment. The corresponding angles $\{\theta_j\}$ become equally spaced in $(0,\pi)$ and the weights $\{\lambda_j\}$ are the Gauss-Chebyshev weights (multiplied by 2). Such a formula, though obtained in a completely different way, is reminiscent of the non-standard Gaussian formula for the disk studied in [1, 2], which has a similar structure being based on vertical lines corresponding to a different family of equally spaced angles in $(0,\pi)$ (related to the zeros of Chebyshev polynomials of the second kind); see also the discussion after Theorem 2.2 in [1] for a standard pointwise use of the formula. That formula is sligthly more efficient, since it has $n^2$ nodes at exactness degree $2n-1$, whereas (5)-(6) uses $n(n+1)$ nodes for the same degree. We stress, however, that our formula is not tailored to the disk, but to general circular segments.

In [7], we provide a Matlab function, `gqcircsegm`, that computes the nodes and weights (6), via (3) implemented by the Matlab function `trigauss` [6]. In order to test numerically the polynomial exactness of the cubature formula, we have computed the integral of the positive polynomial $(x + y + 2)^n$ on circular segments (4) with $R = 1$, for several values of $n$ and $\omega$, that is

$$I(\omega,n) = \iint_S (x+y+2)^n \, dx\, dy$$

$$= \int_0^\omega \frac{(\cos(\theta)+\sin(\theta)+2)^{n+1} - (\cos(\theta)-\sin(\theta)+2)^{n+1}}{n+1} \sin(\theta)\, d\theta \,. \tag{7}$$

In Table 1 we report the maximum and average of the relative error sequence

$$E(\omega,n) = \frac{\left| I(\omega,n) - \sum_{j=1}^{\lceil\frac{n+2}{2}\rceil} \sum_{i=1}^{\lceil\frac{n+1}{2}\rceil} W_{ij}\,(x_{ij}+y_{ij}+2)^n \right|}{I(\omega,n)} \,, \tag{8}$$

made by `gqcircsegm` for $n = 5,10,\dots,95,100$. The reference values of $I(\omega,n)$ have been obtained by integrating the trigonometric polynomial in (7) by the Matlab function `trigquad` in [6] (that implements the Fejer-like trigonometric quadrature rule studied in [3]).

**Table 1:** Maximum and average of the sequence of relative errors (8), made by the Matlab function `gqcircsegm` for $n = 5,10,15,\dots,95,100$.

| $\omega$ | $\pi/16$ | $\pi/8$ | $\pi/4$ | $\pi/2$ | $3\pi/4$ | $7\pi/8$ | $15\pi/16$ |
|---|---|---|---|---|---|---|---|
| $E_{max}$ | 3.2e-15 | 7.0e-15 | 7.2e-15 | 9.8e-15 | 1.1e-14 | 1.1e-14 | 9.1e-15 |
| $E_{av}$ | 1.1e-15 | 1.7e-15 | 2.0e-15 | 2.5e-15 | 2.8e-15 | 3.2e-15 | 2.5e-15 |

## 2.1 Planar lenses and double bubbles

A *planar lens*, that is the *intersection of two overlapping disks* with possibly different radii and distance of the centers smaller than the sum of the radii, is the union of two circular segments with the chord in common (see Figure 1 left). The lens become symmetric when the radii coincide. We can obtain a cubature formula for such a (in general *asymmetric*) lens, simply by union of the nodes (and corresponding weights) for the two circular segments. The overall number of nodes at exactness degree $n$ is then approximately $2(n+1)(n+2)/4 = n^2/2 + \mathcal{O}(n)$. In [5, 6], an algebraic cubature formula with approximately the same number of nodes has been implemented, which uses a different transformation and is restricted to the special case of symmetric lenses.

In [7] we provide a Matlab function, `gqlens`, that computes the nodes and weights of the cubature formula for the intersection of two arbitrary disks, by (5)-(6), making the appropriate translations and rotations. In Figure 1 left, we show an example of algebraic cubature nodes for an asymmetric lens.

The implementation does not use directly `gqcircsegm`, in order to compute only once the relevant Gauss-Legendre nodes and weights (whereas the trigonometric Gaussian angles and weights have to be computed separately for each circular arc by `trigauss` [6]). In the singular case when the intersection has measure zero (non overlapping disks), the function returns one single node (one of the two centers) with zero weight, so that any computing process calling `gqlens` can go on correctly. The computing time of nodes and weights at a given degree of exactness $n$, varies experimentally from some $10^{-3}$ seconds for small $n$, to some $10^{-2}$ seconds around $n = 100$, independently of $\omega$. All numerical tests have been made in Matlab 7.7.0 with an Athlon 64 +3800 2.40GHz processor.

Similar considerations hold for a planar *double bubble* [14], i.e. the *union of two overlapping disks* with possibly different radii, which is also the union of two circular segments with the chord in common, at least one having angle greater than $\pi$. Indeed, these two circular segments are the “complementary” ones with respect to those forming the corresponding lens; see

Figure 1 right. The cubature formula for double bubbles has again $n^2/2 + \mathcal{O}(n)$ nodes at exactness degree $n$, and is implemented by the Matlab function `gqdbubble` in [7].

*Remark* 1. *Concerning the convergence rate of our algebraic cubature formulas, due to the positivity of the weights it is simple to show by standard arguments of quadrature theory that, for any fixed $k > 0$ and sufficiently regular integrand $f$, we have the error estimate*

$$\iint_Q f(x,y)\,dx\,dy = \sum_{j=1}^{N_n} \sum_{i=1}^{\lceil \frac{n+1}{2} \rceil} \sigma_{ij} f(a_{ij}, b_{ij}) + \mathcal{O}(n^{-k}) \,, \tag{9}$$

*where $Q$ is either a circular segment ($N_n = \lceil (n+2)/2 \rceil$), or a lens, or a double bubble (the latter two with $N_n = 2\lceil (n+2)/2 \rceil$), and $\{(a_{ij}, b_{ij})\}$, $\{\sigma_{ij}\}$ denote the nodes and weights of the corresponding cubature formulas. Indeed, such compacts are all* Jackson compacts.

*We recall that a fat compact set $Q \subset \mathbb{R}^d$ (i.e., $Q = \overline{intQ}$) is termed a* Jackson compact *if it admits a* Jackson inequality*, namely for each $k \in \mathbb{N}$ there exist a positive integer $m_k$ and a positive constant $c_k$ such that*

$$n^k \, dist_Q(f, \mathbb{P}_n^d) \le c_k \sum_{|i| \le m_k} \|D^i f\|_Q \,, \; n > k \,, \;\; \forall f \in C^{m_k}(Q) \tag{10}$$

*where $dist_Q(f, \mathbb{P}_n^d) = \inf\{\|f - p\|_Q \,, \, p \in \mathbb{P}_n^d\}$. Examples of Jackson compacts are $d$-dimensional cubes (with $m_k = k+1$) and Euclidean balls (with $m_k = k$). Any circular segment (and any lens) is a Jackson compact, being fat and convex (and thus a Whitney-regular Markov compact [13, Thm. 7]); on the other hand, any planar double bubble is a Jackson compact being the union of two disks [13, Thm. 1].*

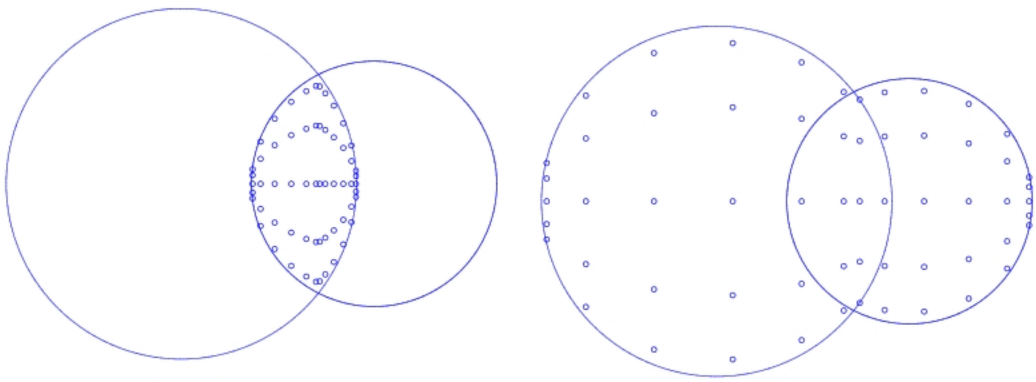

**Figure 1:** $60 = 2 \times (6 \times 5)$ cubature nodes of algebraic exactness degree 9 on a planar lens (left) and a double bubble (right).

### 2.1.1 Example: integrating the product of Wendland RBFs

A general cubature formula on planar lenses is useful, for example, whenever one has to integrate numerically the product of two functions supported on disks (with possibly different radii). This is exactly the situation that occurs within projection methods by compactly supported basis functions with radial support. In continuous RBF least squares approximation, one has to integrate a large number of such products to build the relevant *Gram* matrix. On the other hand, in meshfree Galerkin-like methods for elliptic PDEs, for example in RBF Galerkin schemes, one has to integrate a large number of such products to construct the relevant *stiffness* matrix; cf., e.g., [8, 9].

For the purpose of illustration, we compute the integral

$$I(a) = \iint_\Omega \varphi\left(\|P\|_2\right) \, \varphi\left(\|P - A\|_2\right) \, dx\,dy \,, \;\; P = (x,y) \,, \, A = (0,a) \,, \tag{11}$$

where $\varphi$ is the $C^2$ Wendland Radial Basis Function $\varphi(r) = (1-r)_+^4(4r+1)$ (cf., e.g., [9]), and $\Omega$ the lens corresponding to the intersection of the unit disks centered at $(0,0)$ and $(0,a)$.

In Table 2, we compare the errors and the computing times for different values of $a \in [0,2)$, using either the standard Matlab bivariate integrator `dblquad`, or our cubature Matlab function `gqlens` [6]. We stress that in its present version `gqlens` is not able to guarantee a prescribed error tolerance, but simply provides a cubature formula exact for bivariate polynomials of a certain total-degree. Indeed, we have taken experimentally the smallest $n$ such that the error goes below the same tolerance given to `dblquad`, that is $10^{-6}$. On the other hand, integration by `dblquad` has been made on the least rectangle enclosing the lens, that is $[a-1,1] \times [-\sin(\arccos(a/2)), \sin(\arccos(a/2))]$; for $a = 0$ the disks coincide, whereas $a = 1.9$ corresponds to a small intersection (the integral in this case is extremely small, around $10^{-12}$). The numerical tests have been made in Matlab 7.7.0 with an Athlon 64 +3800 2.40GHz processor. The reference values of the integral have been computed by `dblquad` with a relative tolerance of $10^{-10}$.

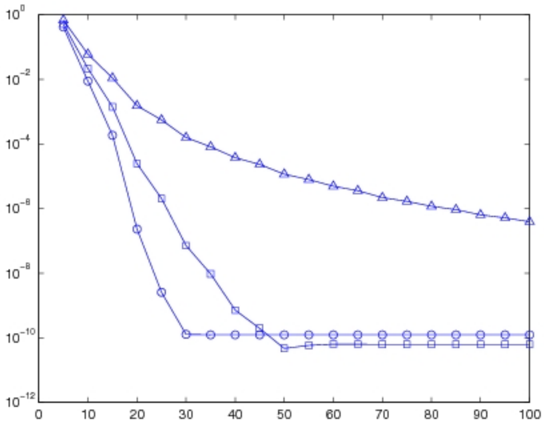


**Figure 2:** Relative errors of `gqlens` in the computation of the integral (11): $a = 0.5$ (triangles), $a = 1$ (squares), $a = 1.5$ (circles).

Observe that for $a > 1$ the centers are outside the lens (for $a = 1$ they are on the boundary), and the integrand becomes $C^\infty$ on the lens. See Figure 2 for a comparison of the error decay with different values of $a$.

We see from Table 2 that `gqlens` is quite efficient on this problem, with a speed-up with respect to `dblquad` varying from about 10 (intersections containing the centers) up to 40 (smallest intersection). The speed-up can be even more impressive at lower tolerances. For example, with $a = 1$ and a required relative error below $10^{-9}$, `gqlens` needs a degree of exactness $n = 40$ and about 0.01 seconds, versus 4.8 seconds of `dblquad`, that is a speed-up around 500.

Though we cannot simply say that these speed-ups would be inherited in the construction of Gram or stiffness matrices on some domain (cf., e.g., [9]), since the situation is more complicated for those integrals interacting with the domain boundary, still the example suggests that our cubature formula on lenses could be useful in the framework of projection methods by compactly supported basis functions with radial support.

**Table 2:** Relative errors and CPU times (seconds) in the computation of the integral (11), by `dblquad` with a relative tolerance of $10^{-6}$, and by `gqlens` where $n$ is the least degree such that the error becomes smaller than $10^{-6}$ (this fails for $a = 1.9$ where we report the smallest error for $n = 1, \dots, 100$).

| | | $a = 0$ | $a = 0.1$ | $a = 0.5$ | $a = 1$ | $a = 1.5$ | $a = 1.9$ |
|---|---|---|---|---|---|---|---|
| `dblquad` | err | 1.8e-6 | 2.2e-6 | 1.9e-6 | 4.9e-6 | 1.4e-6 | 5.5e-5 |
| | cpu | 0.33s | 0.32s | 0.29s | 0.28s | 0.30s | 0.40s |
| `gqlens` | $n$ | 85 | 112 | 84 | 26 | 20 | 14 |
| | err | 9.6e-7 | 9.9e-7 | 9.2e-7 | 6.4e-7 | 2.3e-7 | 4.7e-6 |
| | cpu | 0.02s | 0.04s | 0.02s | 0.01s | 0.01s | 0.01s |

**Acknowledgments.** This work has been supported by the “ex-60%” funds of the University of Padova, and by the GNCS-INdAM.

## References

[1] B. Bojanov, Quadrature formulae of non-standard type, Rend. Mat. Appl. 25 (2005), 195–221.

[2] B. Bojanov and G. Petrova, Numerical integration over a disk. A new Gaussian quadrature formula, Numer. Math. 80 (1998), 39–59.

[3] L. Bos and M. Vianello, Subperiodic trigonometric interpolation and quadrature, submitted, draft online at: http://www.math.unipd.it/~marcov/CAApubl.html.

[4] R. Cools and K.J. Kim, A survey of known and new cubature formulas for the unit disk, Korean J. Comput. Appl. Math. 7 (2000), 477–485.

[5] G. Da Fies and M. Vianello, Trigonometric Gaussian quadrature on subintervals of the period, submitted, draft online at: http://www.math.unipd.it/~marcov/CAApubl.html.

[6] G. Da Fies and M. Vianello, `trigquad.m`, `trigauss.m`, `gqcircsect.m`, `gqcirczone.m`, `gqsymmlens.m`: Matlab functions for subperiodic trigonometric quadrature and for product Gaussian quadrature on circular (annular) sectors, zones and symmetric lenses, online at:
http://www.math.unipd.it/~marcov/CAAsoft.html.

[7] G. Da Fies and M. Vianello, `gqcircsegm.m`, `gqlens.m`, `gqdbubble.m`: Matlab functions for product Gaussian quadrature on circular segments, general planar lenses and double bubbles, online at:
http://www.math.unipd.it/~marcov/CAAsoft.html.

[8] S. De and K.-J. Bathe, The method of finite spheres with improved numerical integration, Comput. & Structures 79 (2001), 218–231.

[9] G. Fasshauer, Meshfree Approximation Methods with MATLAB, World Scientific, Singapore, 2007.

[10] W. Gautschi, Orthogonal Polynomials: Computation and Approximation, Oxford University Press, New York, 2004.

[11] W. Gautschi, Orthogonal polynomials (in Matlab), J. Comput. Appl. Math. 178 (2005), 215–234, software online at: http://www.cs.purdue.edu/archives/2002/wxg/codes.

[12] W.H. Pierce, Numerical integration over the planar annulus, J. Soc. Indust. Appl. Math. 5 (1957), 66–73.

[13] W. Pleśniak, Multivariate Jackson Inequality, J. Comput. Appl. Math. 233 (2009), 815–820.

[14] WolframMathWorld, Double Bubble, online at: http://mathworld.wolfram.com/DoubleBubble.html.