A CARDINAL FUNCTION ALGORITHM FOR COMPUTING MULTIVARIATE QUADRATURE POINTS

MARK A. TAYLOR*, BETH A. WINGATE†, AND LEN P. BOS ‡

Abstract. We present a new algorithm for numerically computing quadrature formulas for arbitrary domains which exactly integrate a given polynomial space. An effective method for constructing quadrature formulas has been to numerically solve a nonlinear set of equations for the quadrature points and their associated weights. Symmetry conditions are often used to reduce the number of equations and unknowns. Our algorithm instead relies on the construction of cardinal functions and thus requires that the number of quadrature points $N$ be equal to the dimension of a prescribed lower dimensional polynomial space. The cardinal functions allow us to treat the quadrature weights as dependent variables and remove them, as well as an equivalent number of equations, from the numerical optimization procedure. We give results for the triangle, where for all degree $d \leq 25$, we find quadrature formulas of this form which have positive weights and contain no points outside the triangle. Seven of these quadrature formulas improve on previously known results.

Key words. multivariate integration, quadrature, cubature, fekete points, spectral methods, triangle, polynomial approximation

AMS subject classifications. 65D32 65D30 65M60 65M70

1. Introduction. Gauss-Lobatto quadrature points are commonly used in numerical methods which rely on both accurate high-order polynomial interpolation and quadrature properties. They are heavily relied on by the diagonal-mass-matrix spectral element method, which has been very successful in geophysical applications dominated by wave propagation [17, 15, 11, 12, 25].

Gauss-Lobatto quadrature points are only known for tensor product domains such as the line, square and cube. It is unclear how to find Gauss-Lobatto like points for non tensor-product domains like triangles or tetrahedrons, which makes it difficult to extend the diagonal-mass-matrix spectral element method to these domains. There are two generalizations that have been studied in some detail. The first involves searching for points in these domains with optimal interpolation properties by minimizing the Lebesgue constant [4], [14], [24]. For high polynomial degree, the best results have been obtained for Fekete points, which can be computed in a natural way with a cardinal function (Lagrange interpolating polynomials) algorithm [24]. The second generalization involves searching for points in the domain of interest which give an optimal quadrature formula for the integral of polynomials over the domain. Here we consider a set of $N$ points $\{z_1, z_2, \ldots, z_N\}$ and weights $\{w_1, w_2, \ldots, w_N\}$ to be a quadrature formula of strength $d$ if the quadrature approximation for a domain $\Omega$,

$$\int_{\Omega} g \approx \sum_{j=1}^{N} w_j g(z_j),$$

is exact for all polynomials $g$ up to degree $d$. Among all quadrature formulas of strength $d$, the optimal formulas are those with the fewest possible points $N$. In
this work, we describe an algorithm for computing near-optimal quadrature formulas which is motivated by the Fekete point algorithm and relies heavily on the construction of cardinal functions.

The quadrature problem has been extensively studied independently of spectral element applications and has a long history of both theoretical and numerical development. For a recent review, see [5, 20, 8, 6]. An on-line database containing many of the best known quadrature formulas is described in [7]. Much of these results are also collected and distributed on CD-ROM in the book [22].

One successful approach for numerically finding quadrature formulas dates to [21]. A generalized version was used recently in [26]. Newton’s method is used to solve the nonlinear system of algebraic equations for the quadrature weights and locations of the points. Symmetry is used to reduce the complexity of the problem. If the quadrature points are invariant under the action of a group $G$, then the number of equations can be reduced to the dimension of the subspace of $P_d$ invariant under $G$.

Motivated by the cardinal function Fekete point algorithm [24], we propose a new method to reduce the complexity of the quadrature problem: we look for quadrature formula that have the same number of points as the dimension of a lower dimensional polynomial space. We can then construct a cardinal function basis for this lower dimensional space, make use of a multi-variate generalization of the Newton-Cotes quadrature weights, and derive a remarkable expression analytically relating the variation in the quadrature weights to the variation of the quadrature points. The net result is a significant reduction in the number of equations and unknowns. Symmetry can still be used to further reduce the complexity of the problem if needed. However here we have been able to find optimal quadrature sets of strength 9 through 25, subject only to the cardinal function constraint without imposing any symmetry constraints on the solutions.

2. Orthogonal Polynomials. We first define our notation and describe the basis that will be used to represent cardinal functions. Let $\Omega$ be a domain in $\mathbb{R}^n$, with $\xi$ an arbitrary point in $\Omega$. Let $P_d$ be the finite dimensional vector space of polynomials in the Cartesian components of $\xi$ of at most degree $d$, and let $N = \dim P_d$. As an example, if $\Omega$ is the right triangle, then

$$P_d = \text{span}\{\xi_1^n\xi_2^m, m + n \leq d\}$$

where $\xi = (\xi_1, \xi_2) \in \mathbb{R}^2$ and $N = \frac{1}{2}(d + 1)(d + 2)$.

Our algorithm requires working with a cardinal function basis for $P_d$. The most practical way to compute cardinal functions numerically is to work instead with their expansions in terms of an orthogonal, easily computed basis for $P_d$. We denote this basis by $\{g_i(\xi), i = 1, \ldots, N\}$. For simplicity, we require that $g_1(\xi) = 1$. Because the remaining basis functions are orthogonal to the constant function, we have

$$\int_{\Omega} g_i d\xi = \begin{cases} |\Omega|, & \text{for } i = 1 \\ 0, & \text{for } i > 1 \end{cases} \quad (2.1)$$

where $|\Omega|$ is the area of $\Omega$ and $d\xi$ represents the uniform area measure. In the right triangle, $\xi_1 \geq -1$, $\xi_2 \geq -1$ and $\xi_1 + \xi_2 \leq 0$, such a basis is given by the Kornwinder-Dubiner polynomials $\{g_{m,n}\}$ [1, 18, 9],

$$g_{m,n}(\xi) = P_m^0 \left( \frac{\xi_1}{1 - \xi_2} \right) (1 - \xi_2)^m P_n^{2m+1,0}(\xi_2), \quad m + n \leq d$$
where $P_{n}^{\alpha,\beta}$ are the Jacobi Polynomials with weight $(\alpha, \beta)$ and degree $n$. The traditional double index $(m,n)$ can be converted into the single index used here by $i = (m+n+1)(m+n+2)/2 - m$. Suitable recurrence relations for these polynomials are given in [16].

3. Cardinal functions and Quadrature points. We now describe the procedure we use to compute cardinal functions defined by a set of $N$ points

$$z = \{z_1, z_2, \ldots, z_N\}$$

where each $z_i$ is a point in the right triangle. If the points are non-degenerate, the cardinal functions can be defined uniquely as the polynomials in $\xi$ which belong to $\mathcal{P}_d$ and satisfy

$$\phi_i(\xi; z) = \begin{cases} 1, & \text{if } \xi = z_i \\ 0, & \text{if } \xi = z_j, j \neq i \end{cases} \quad (3.1)$$

The cardinal function depend implicitly on the defining points, and thus we include a second argument of $z$. To evaluate cardinal functions numerically, we first express them in terms of the orthogonal basis $\{g_m\}$,

$$\phi_i(\xi; z) = \sum_m \hat{\phi}_i^m g_m(\xi). \quad (3.2)$$

The expansion coefficients $\hat{\phi}_i^m$ are computed by evaluating Eq. 3.2 at the points $z_j$ and solving the $N \times N$ linear system

$$\phi_i(z_j; z) = \sum_m \hat{\phi}_i^m g_m(z_j).$$

If reasonable care is used, the points $\{z_j\}$ can be chosen so that the system is well conditioned and easily inverted by Gaussian elimination. The resulting cardinal functions form a basis for $\mathcal{P}_d$.

Following conventional spectral method techniques, we evaluate cardinal functions at an arbitrary point $\xi$ by by simply evaluating the $g_m(\xi)$ (via recurrence relations) and summing the series in Eq. 3.2. Derivatives of cardinal functions with respect to $\xi$ are evaluated in a similar fashion, by first evaluating the derivatives of $g_m$ after differentiating Eq. 3.2.

The Newton-Cotes quadrature formula for $\mathcal{P}_d$ can be constructed at these points by solving the system

$$\sum_{j=1}^{N} w_j \phi_i(z_j; z) = \int \phi_i d\xi \quad \forall \phi_i, i = 1, \ldots, N$$

Making use of Eq. 2.1 and Eq. 3.1, the solution of this system is given by

$$w_i = \int \phi_i d\xi = |\Omega| \hat{\phi}_i^1. \quad (3.3)$$

By construction, the Newton-Cotes weights and the points $\{z_i\}$ give a quadrature formula which exactly integrates our $N$ cardinal functions $\{\phi_i\}$. Since $\mathcal{P}_d = \text{span}\{\phi_i\}$, we have

$$\sum_{j=1}^{N} w_j g(z_j) = \int g d\xi \quad \forall g \in \mathcal{P}_d$$
Thus any set of \(N\) non-degenerate points \(\{z_j\}\) will yield a quadrature formula for \(P_d\), with uniquely determined quadrature weights. The problem now is to find the \(N\) points which integrate all of \(P_d + e\) for the largest possible \(e\).

4. Derivatives of cardinal functions with respect to \(z\). In the algorithm that follows, we will also need to compute the derivative of a cardinal function \(\phi_i\) with respect to the points \(z_j\) used to define \(\phi_i\). For this, we use:

**Proposition 4.1.** We have

\[
\frac{\partial \phi_i}{\partial z_j}(\xi; z) = -\phi_j(\xi; z) \frac{\partial \phi_i}{\partial \xi}(z_j; z).
\]

**Proof.** We actually compute the more detailed derivative

\[
\frac{\partial}{\partial (z_j)_k} \phi_i(\xi; z)
\]

where \((z_j)_k\) denotes the \(k\)th component of the \(j\)th point \(z_j\). To this end, write

\[
\frac{\partial}{\partial (z_j)_k} \phi_i(\xi; z) = \lim_{h \to 0} \frac{\phi_i(\xi; (z\setminus z_j) \cup \{z_j + he_k\}) - \phi_i(\xi; z)}{h}
\]

where \(e_k\) denotes the standard unit direction vector in the \(k\)th coordinate. But the difference

\[
\phi_i(\xi; (z\setminus z_j) \cup \{z_j + he_k\}) - \phi_i(\xi; z)
\]

is zero at the points of \(z\setminus z_j\), as is \(\phi_j(\xi; z)\), and hence by uniqueness,

\[
\phi_i(\xi; (z\setminus z_j) \cup \{z_j + he_k\}) - \phi_i(\xi; z) = C\phi_j(\xi; z)
\]

for some constant \(C\).

To evaluate \(C\), first suppose that \(j \neq i\). Then evaluate at \(\xi = z_j + he_k\) to obtain (for sufficiently small \(h\))

\[
0 - \phi_i(z_j + he_k; z) = C\phi_j(z_j + he_k; z)
\]

so that

\[
C = \frac{\phi_i(z_j + he_k; z)}{\phi_j(z_j + he_k; z)}
\]

Hence,

\[
\frac{\partial}{\partial (z_j)_k} \phi_i(\xi; z) = \lim_{h \to 0} \frac{\phi_i(z_j + he_k; z)}{h\phi_j(z_j + he_k; z)} \phi_j(\xi; z)
\]

\[
= \frac{\phi_j(\xi; z)}{\phi_j(z_j; z)} \lim_{h \to 0} \frac{0 - \phi_i(z_j + he_k; z)}{h}
\]

\[
= \frac{\phi_j(\xi; z)}{\phi_j(z_j; z)} \lim_{h \to 0} \frac{\phi_i(z_j; z) - \phi_i(z_j + he_k; z)}{h}
\]

\[
= -\frac{\phi_j(\xi; z)}{1} \frac{\partial \phi_i}{\partial \xi_k}(z_j; z)
\]

\[
= -\phi_j(\xi; z) \frac{\partial \phi_i}{\partial \xi_k}(z_j; z).
\]
Similarly, if \( j = i \) we evaluate at \( \xi = a_i + he_k \) to obtain

\[
C = \frac{1 - \phi_i(z_i + he_k; z)}{\phi_i(z_i; z)} = \frac{\phi_i(z_i; z) - \phi_i(z_i + he_k; z)}{\phi_i(z_i + he_k; z)}
\]

so that

\[
\frac{\partial}{\partial(z_i)_k} \phi_i(\xi; z) = -\phi_i(\xi; z) \frac{\partial \phi_i}{\partial \xi_k}(z_i; z).
\]

Thus the derivative of the \( i \)’th cardinal function with respect to \( j \)’th quadrature point is given by the \( j \)’th cardinal function times a term independent of \( \xi \) and involving only the conventional derivative. The later term can be easily evaluated by differentiating Eq. 3.2.

Using Eq. 3.3 and Eq. 4.1, we can also derive a similar relation showing that the derivative of the \( i \)’th weight with respect to the \( j \)’th quadrature point is given by the \( j \)’th weight times the same term that appears in Eq. 4.1:

\[
\frac{\partial w_i}{\partial z_j} = -\int \phi_j(\xi; z) \frac{\partial \phi_i}{\partial \xi}(z_j; z) d\xi = -w_j \frac{\partial \phi_i}{\partial \xi}(z_j; z) \] (4.2)

5. A cardinal function algorithm for computing quadrature points for \( \mathcal{P}_{d+e} \). We now describe an iterative method for improving an initial set of \( N \) quadrature points \( z \). In order to integrate a space larger than \( \mathcal{P}_d \), we need to find quadrature points which satisfy the nonlinear equation

\[
\sum_i w_i g_m(z_i) = \int g_m d\xi \quad \forall g_m \in \mathcal{P}_{d+e}
\] (5.1)

for some \( e > 0 \). By using the Newton-Cotes weights given by Eq. 3.3, we automatically integrate all of \( \mathcal{P}_d \), thus we need only satisfy the equations for the basis functions in \( \mathcal{P}_{d+e} \) which are not in \( \mathcal{P}_d \):

\[
\sum_i w_i g_m(z_i) = 0 \quad \forall g_m : d < \text{degree } g_m \leq d + e
\]

and we have replaced the integral on the left-hand-side of Eq. 5.1 by 0 by virtue of Eq. 2.1. Define

\[
F_m = \sum_i w_i g_m(z_i)
\]

and \( F = \{ F_m : d < \text{degree } g_m \leq d + e \} \). Note that since the weights are determined by \( z \), we can treat \( F \) as a function solely of \( z \). Then Eq. 5.1 is equivalent to \( F = 0 \), which can be solved using Newton’s method:

\[
\frac{\partial z}{\partial t} = -(\nabla F)^{-1} F.
\]

The gradient of \( F \) is not necessarily square. Following [26], we use a pseudo-inverse for \( (\nabla F)^{-1} \), and restrict ourselves to the under-determined case by choosing \( e \) so
Taylor, Wingate and Bos

that there are more equations then degrees of freedom in the problem. There is one
degree of freedom for each coordinate of each point in \( \mathbb{R}^n \), for a total of \( n \dim P_d \).
The number of equations is given by \( \dim P_{d+t} - \dim P_d \). Thus the degrees-of-freedom
constraint is given by \( \dim P_{d+t} \leq (n+1) \dim P_d = (n+1)N \).

Using Eq. 4.2, the components of the gradient of \( F \) are given by

\[
\frac{\partial F_m}{\partial z_j} = \sum_i \left( w_i \frac{\partial g_m(z_i)}{\partial z_j} + \frac{\partial w_i}{\partial z_j} g_m(z_i) \right)
\]

Comparing this approach to the traditional Newton method for quadrature, such as
in [26], we see that the use of the Newton-Cotes weights has removed the weights
from the iteration, and thus reduced the number of unknowns by \( N \). Since these
weights exactly integrate \( P_d \), we have also reduced the number of equations by \( N \).
The only increase in complexity is the addition of the term involving the derivative of
the weights with respect to the quadrature points. But this term is easy to evaluate
by virtue of Eq. 4.2.

If symmetry is imposed on the quadrature points, then additional reductions in
the number of equations and unknowns are possible, as in [21]. However, for the
results presented in this paper we typically do not impose any symmetry constraints.

6. Practical considerations. In practice, Newton’s method is only used to ac-
celerate the convergence of a slower, more robust algorithm. We first use the steepest
descent algorithm to minimize the quadrature error, \( F \cdot F \). This algorithm simply
moves the points in the direction of steepest descent given by \( \nabla (F \cdot F) \):

\[
\frac{\partial z_j}{\partial t} = -2 \sum_m F_m \frac{\partial F_m}{\partial z_j}
\]

Once this algorithm has found a possible quadrature formula, we switch to Newton’s
method and iterate until the sequence converges. If the iteration fails to converge,
then another initial condition is chosen and the procedure is repeated.

Our procedure for constructing initial conditions is the same as that used in [24].
We choose a distribution of points which approximates the extremal measure \( \mu \) for
the triangle given in [2]. For the triangle, it is conjectured that \( \mu \) is also the density
of quadrature points with positive weights in the limit \( N \) goes to infinity and it was
recently shown that this limit is bounded below by \( c \mu \) for some constant \( c \). [19].
For the right triangle \( \xi_1 \geq 0, \xi_2 \geq 0 \) and \( \xi_1 + \xi_2 \leq 1 \), the extremal measure is

\[
\mu(\xi) = \frac{1}{\sqrt{\xi_1 \xi_2 (1 - \xi_1 - \xi_2)}}
\]

To distribute a finite set of points to approximate the given density \( \mu(\xi) \), we first
assume the points lie in a nested family of triangles. We then compute a nested family
of triangular shells, each with a mass (using the measure \( \mu(\xi) d\xi \)) proportional to the
number of points we have decided to place in that shell. If there are \( k \) points to be
placed in a given shell, we break that shell into \( k \) quadrilateral pieces, all with the same
mass, and place one point in the center of each piece. For a given number of points,
there are a variety of configurations which can be generated by altering the number
of points within each shell and the number of shells. The cardinal function algorithm
is extremely sensitive to the initial condition, so many of these initial conditions must
be tried to find an optimal solution.
### Table 7.1

<table>
<thead>
<tr>
<th>Degree of cardinal functions (d)</th>
<th>Number of Points (N)</th>
<th>Degree of Exact Integration (d+e)</th>
<th>Error</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td>$4.4 \times 10^{-16}$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>5</td>
<td>$9.7 \times 10^{-16}$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>7</td>
<td>$1.7 \times 10^{-14}$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>9</td>
<td>$2.1 \times 10^{-14}$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>21</td>
<td>11</td>
<td>$2.8 \times 10^{-14}$</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>28</td>
<td>13</td>
<td>$4.7 \times 10^{-15}$</td>
<td>asym</td>
</tr>
<tr>
<td>7</td>
<td>36</td>
<td>14</td>
<td>$2.2 \times 10^{-14}$</td>
<td>asym,new</td>
</tr>
<tr>
<td>8</td>
<td>45</td>
<td>16</td>
<td>$1.8 \times 10^{-15}$</td>
<td>asym,new</td>
</tr>
<tr>
<td>9</td>
<td>55</td>
<td>18</td>
<td>$8.6 \times 10^{-15}$</td>
<td>asym,new</td>
</tr>
<tr>
<td>10</td>
<td>66</td>
<td>20</td>
<td>$3.3 \times 10^{-14}$</td>
<td>asym,new</td>
</tr>
<tr>
<td>11</td>
<td>78</td>
<td>21</td>
<td>$2.8 \times 10^{-14}$</td>
<td>asym,new</td>
</tr>
<tr>
<td>12</td>
<td>94</td>
<td>23</td>
<td>$2.9 \times 10^{-14}$</td>
<td>new</td>
</tr>
<tr>
<td>13</td>
<td>105</td>
<td>25</td>
<td>$3.3 \times 10^{-14}$</td>
<td>asym,new</td>
</tr>
<tr>
<td>14</td>
<td>120</td>
<td>27</td>
<td>$4.3 \times 10^{-14}$</td>
<td>asym,new</td>
</tr>
</tbody>
</table>

Quadrature points computed with the cardinal function algorithm. In all cases, the quadrature weights are positive and the points are not outside the triangle. Solutions which are not $D_3$ symmetric are denoted by asym. Solutions which improve upon previously published results are denoted by new.

### 7. Results

Our results for the triangle are summarized in Table 7.1. Except for quadrature formulas associated with $d = 3$ and $d = 4$, were able to obtain the optimal solution (fewest number of points) subject to the cardinal function constraint on the number of points and the degrees-of-freedom constraint:

\[
N = \dim P_d, \tag{7.1}
\]

\[
\dim P_{d+e} \leq 3N. \tag{7.2}
\]

All the quadrature points have positive weights and no points lie outside the triangle, although neither of these properties is in any way guaranteed by the cardinal function algorithm. The errors presented in the table is the max norm of the quadrature error over all the ortho-normal basis functions:

\[
\max_{g_{m,n} \in P_{d+e}} \left| \sum_i w_i g_{m,n}(z_i) - \int g_{m,n} d\xi \right|
\]

with normalization $\int g_{m,n}^2 d\xi = |\Omega|$. Many of the quadrature sets are invariant under the symmetry group of rotations and reflections of the triangle, $D_3$. The solutions which do not have this symmetry are denoted with asym in the table.

Quadrature formulas denoted by new in the table represent formulas which improve upon the best previously published results, as taken from the extensive database described in [7] and the quadrature points presented in [26] (which are not included in the database as of this writing). The new solutions for integration degree $d + e$ from 18 to 25 have fewer points then the previously published results. For $d + e = 13$ and 16, the results presented here have the fewest number of points among formulas with positive weights and no points outside the triangle. For $d + e = 13$, the previous result with the fewest number of quadrature points has $N = 36$, but but some of those points are outside the triangle and not all the weights are positive [3]. For $d + e = 16$, previous results include formulas with 52 points, some of which are outside the triangle [10], and 55 points, some of which have negative weights [13].

In Table 7.2, we summerize the resuls for the triangle from [7], [26], and the cardinal function algorithm.
The coordinates of the points for the first four new quadrature formulas are given in Appendix I. The coordinates for all the formulas in the table are available electronically from [23]. Plots for the first four of these quadrature points are shown in Fig. 7.1. In the figure, the right triangle has been mapped lineally to the equilateral triangle in order to make the asymmetry in the points more visible.

8. Summary. We have presented a cardinal function algorithm for computing multi-variate quadrature points. The key ideas involve the use of Newton-Cotes weights expressed as integrals of cardinal functions and a formula relating the derivatives of cardinal functions with respect to $z_i$ (their defining points) to conventional derivatives in $\xi$. These two ideas allow us to reduce the number of equations and number of unknowns by $N$, while still retaining analytic expressions for the gradients necessary to apply steepest decent or Newton iterations. The algorithm was applied to the triangle, where optimal (in the sense of Equations 7.1 and 7.2) formulas were constructed for integrating polynomials up to degree 25. Seven of these quadrature formulas improve on previously known results.

To apply the algorithm to other domains and more than two dimensions only requires the knowledge of an orthogonal basis of polynomials and the ability to evaluate the basis functions arbitrary points. The use of cardinal functions requires that

<table>
<thead>
<tr>
<th>Degree of Exact Integration</th>
<th>Number of points (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cools</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>10†</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>15†</td>
</tr>
<tr>
<td>9</td>
<td>19</td>
</tr>
<tr>
<td>10</td>
<td>22†</td>
</tr>
<tr>
<td>11</td>
<td>27†</td>
</tr>
<tr>
<td>12</td>
<td>33</td>
</tr>
<tr>
<td>13</td>
<td>36†</td>
</tr>
<tr>
<td>14</td>
<td>42</td>
</tr>
<tr>
<td>15</td>
<td>48†</td>
</tr>
<tr>
<td>16</td>
<td>52†</td>
</tr>
<tr>
<td>17</td>
<td>61</td>
</tr>
<tr>
<td>18</td>
<td>67†</td>
</tr>
<tr>
<td>19</td>
<td>73</td>
</tr>
<tr>
<td>20</td>
<td>79†</td>
</tr>
<tr>
<td>21</td>
<td>93</td>
</tr>
<tr>
<td>22</td>
<td>100</td>
</tr>
<tr>
<td>23</td>
<td>106</td>
</tr>
<tr>
<td>24</td>
<td>118</td>
</tr>
<tr>
<td>25</td>
<td>126</td>
</tr>
<tr>
<td>26</td>
<td>138</td>
</tr>
<tr>
<td>27</td>
<td>145</td>
</tr>
<tr>
<td>28</td>
<td>154</td>
</tr>
<tr>
<td>29</td>
<td>166</td>
</tr>
<tr>
<td>30</td>
<td>175</td>
</tr>
</tbody>
</table>

Table 7.2

Comparisons of known quadrature formula for the triangle. The column labeled Cools gives the results collected in [7]. The column labeled Wandzura and Xiao gives the results from [26]. Formulas with negative weights or points that lie outside the triangle are denoted with a †.
Fig. 7.1. Quadrature points for the triangle which, from left to right and top to bottom, exactly integrate polynomials of degree 13, 16, 18 and 20. No points are outside the triangle, and all quadrature weights are positive.

$N = \dim P_d$. This constraint can be relaxed by replacing $P_d$ with any subspace $P' \subset P_{d+c}$. The algorithm is unmodified other than that one needs to compute cardinal functions and Newton-Cotes weights for the space $P'$ instead of $P_d$. However, there are many choices for $P'$ and it is not known how the choice of $P'$ affects the results.

REFERENCES

Appendix A. Tables of quadrature points. We now list the coordinates of the first four quadrature formulas marked with a new in Table 7.1. Coordinates for all formulas are available electronically in [23]. For each line, we give the first two barycentric coordinates of each point (equivalent to the $x$ and $y$ coordinates after an equilateral triangle is linearly mapped to the unit right triangle $x \geq 0$, $y \geq 0$ and $x + y \leq 1$) followed by the associated quadrature weight. The third barycentric coordinate is defined such that the sum of all three coordinates is one.

<table>
<thead>
<tr>
<th>integration degree=13 N=36:</th>
<th>0.4757672298101</th>
<th>0.5198921829102</th>
<th>0.018447461845</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0242935351590</td>
<td>0.9493059293846</td>
<td>0.0166240998757</td>
<td></td>
</tr>
<tr>
<td>0.0265193427722</td>
<td>0.0242695130640</td>
<td>0.0166811699778</td>
<td></td>
</tr>
<tr>
<td>0.9492126023551</td>
<td>0.0265067966437</td>
<td>0.0166830569067</td>
<td></td>
</tr>
<tr>
<td>0.0033775767349</td>
<td>0.4767316412363</td>
<td>0.0175680870083</td>
<td></td>
</tr>
<tr>
<td>0.4757672298101</td>
<td>0.5198921829102</td>
<td>0.018447461845</td>
<td></td>
</tr>
<tr>
<td>0.4757672298101</td>
<td>0.5198921829102</td>
<td>0.018447461845</td>
<td></td>
</tr>
<tr>
<td>0.0242935351590</td>
<td>0.9493059293846</td>
<td>0.0166240998757</td>
<td></td>
</tr>
<tr>
<td>0.0265193427722</td>
<td>0.0242695130640</td>
<td>0.0166811699778</td>
<td></td>
</tr>
<tr>
<td>0.9492126023551</td>
<td>0.0265067966437</td>
<td>0.0166830569067</td>
<td></td>
</tr>
<tr>
<td>0.0033775767349</td>
<td>0.4767316412363</td>
<td>0.0175680870083</td>
<td></td>
</tr>
</tbody>
</table>