Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

High-order symmetric cubature rules for tetrahedra and pyramids

Jan Jaśkowiec[1] | N. Sukumar[*,2]

[1]Faculty of Civil Engineering, Cracow University of Technology, Warszawska 24, 31-155 Cracow, Poland
[2]Department of Civil and Environmental Engineering, University of California, Davis, CA 95616, USA

Correspondence
*N. Sukumar, Department of Civil and Environmental Engineering, University of California, One Shields Avenue, Davis, CA 95616, USA
Email: nsukumar@ucdavis.edu

In this article, we present an algorithm to construct high-order fully symmetric cubature rules for tetrahedral and pyramidal elements, with positive weights and integration points that are in the interior of the domain. Cubature rules are fully symmetric if they are invariant to affine transformations of the domain. We divide the integration points into symmetry orbits where each orbit contains all the points generated by the permutation stars. These relations are represented by equality constraints. The construction of symmetric cubature rules require the solution of nonlinear polynomial equations with both inequality and equality constraints. For higher orders, we use an algorithm that consists of five sequential phases to produce the cubature rules. In the literature, symmetric numerical integration rules are available for the tetrahedron for orders $p = 1$–10, 14, and for the pyramid up to $p = 10$. We have obtained fully symmetric cubature rules for both of these elements up to order $p = 20$. Numerical tests are presented that verify the polynomial-precision of the cubature rules. Convergence studies are performed for the integration of exponential, weakly singular and trigonometric test functions over both elements with flat and curved faces. With increase in p, improvements in accuracy is realized, though nonmonotonic convergence is observed.

KEYWORDS:
numerical integration, positive weights, pyramidal element, strictly interior integration points, symmetric cubature rules, tetrahedral element

1 | INTRODUCTION

In this article, we construct high-order fully symmetric cubature rules for tetrahedra and pyramids. This contribution draws on our recent work,[1] in which high-order non-symmetric cubature rules over tetrahedra were proposed. We include full symmetries via additional equality constraints in the algorithm to realize symmetric cubature rules. This article is self-contained; however, to avoid needless repetitions, where appropriate we point the reader to Jaśkowiec and Sukumar[1] for some of the details.

Finite element (FE) methods require numerical integration to be performed over two- and three-dimensional element shapes. In three-dimensional higher order FE calculations,[2,3] efficient and accurate numerical integration (cubature) schemes are required to deliver reliable and fast simulations. Any cubature scheme is represented as $\{\boldsymbol{x}_i, w_i\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the i-th integration point (node) and w_i is the corresponding weight. It is desirable that all weights are positive and all nodes are in the interior of the domain, which are collectively referred to as the 'PI' criteria. In addition, in many applications involving finite elements, it is also desirable that the cubature scheme is symmetric, i.e., it is invariant to affine transformation into itself.[4,5,6,7] Symmetric rules are preferred since symmetric distribution of interpolation nodes (e.g., shape functions on a tetrahedron) can be leveraged to deliver fast assembly of stiffness and mass matrices in finite element analysis. More importantly, use of symmetric cubature rules in nonlinear finite element analysis ensures that the simulation results do not depend on the nodal ordering in the element connectivity.

In three dimensions, commonly used finite elements shapes are: tetrahedron, hexahedron, prism and pyramid. For all of these elements, a mapping is available to a reference element on which numerical integration is carried out. For hexahedral and prismatic elements, symmetric cubature schemes can be readily obtained using tensor-product of univariate and 2D Gaussian rules, even though the number of nodes is far from optimal. For the tetrahedron and pyramid, such standard tensor-product constructions are not possible. Use of the Duffy transformation[8] to generate symmetric integration schemes for pyramids has been proposed,[9] but this approach brings additional complications in the numerical integration. The tetrahedral, pyramidal, hexahedral, and prismatic elements are well-studied in the literature.[10] Bedrosian[11] was the first to construct shape functions for pyramidal elements, which was improved upon and extended for higher order by Bergot et al.[12] For meshing complex geometries, pyramidal elements serve as a glue to connect tetrahedral and hexahedral elements, which ensures C^0-continuity over the finite element mesh.[13] Various contributions pertaining to pyramidal finite element have appeared in the literature, among which we mention the construction of H-div and H-curl conforming higher order pyramidal elements[14,15,16,17,18] and a five-node pyramidal element for explicit dynamics simulations.[19]

In this article, we present fully symmetric cubature rules up to order (synonymous with polynomial degree) $p = 20$ for tetrahedral and pyramidal finite elements. Over the past decade, a few contributions have appeared on the construction of symmetric cubature rules over tetrahedra and pyramids,[20,4,21,7,6,9,22] with rules that are available for $p = 1$–10, 14 and $p = 1$–10 for the tetrahedron and the pyramid, respectively. As noted in Jaśkowiec and Sukumar,[1] constructing high-order 3D cubature rules over finite elements is a challenging problem, which is further compounded herein due to the presence of additional symmetry constraints. Symmetric cubature rules for simplexes (including the tetrahedron) of arbitrary order can be generated using open or closed Newton-Cotes formulas proposed by Silvester.[23] However, this approach provides far from optimal number of integration points and also includes some negative weights. In Zhang et al,[21] barycentric coordinates are applied to generate two symmetric cubature rules ($p = 8$ and $p = 14$) for the tetrahedron. Shunn and Ham[7] describe a family of symmetric integration rules for tetrahedra. The distribution of points in each cubature rule is based on an underlying cubic close-packed grid, and the precise point locations and weights are optimized to reduce the truncation error in the cubature approximation. This approach results in a family of symmetric rules up to $p = 7$ for a tetrahedron. A different approach is utilized in Kubatko et al,[9] where symmetric cubature rules for the pyramid up to $p = 8$ (PI criteria is met) are obtained by using the Duffy transformation from the hexahedron. In William et al,[6] a sphere close-packed lattice arrangement of points is utilized in the formulation of symmetric cubature rules on the tetrahedron. An analytical approach is used in Chen et al[22] to find symmetric cubature rules for the pyramid up to order $p = 3$. Witherden and Vincent[4] use the refinement approach of Zhang et al[21] to generate symmetric schemes up to order $p = 10$ for the tetrahedron and the pyramid.

In this article, the algorithm we use to generate symmetric cubature rules for tetrahedral and pyramidal elements is an extension of our recent work,[1] where non-symmetric cubature rules up to $p = 20$ over the tetrahedron are proposed. To construct symmetric rules, we adopt combinations of symmetry orbits as first introduced by Felippa,[24] and later also used by Zhang et al,[21] Witherden and Vincent,[4] and Geevers at al.[25] Each orbit consists of a set of symmetric nodes that are generated from a permutation star. The position of the nodes is represented by free variables that are present in the permutation star. These free variables are determined through an iterative search procedure. The algorithm has five sequential phases, where each phase 'improves' the position of the nodes in the orbits, so that the final cubature scheme so devised has an accuracy of $\mathcal{O}(10^{-160})$. For $p < 8$, all five phases are not necessary, but in other cases due to ill-conditioning in the solution of the nonlinear system of equations and round-off errors that accumulate, all five phases are required. As an improvement over the current state-of-the-art, we construct cubature rules up to order $p = 20$ for both the tetrahedron and the pyramid. To assess their accuracy, the polynomial-precision of all cubature rules is verified using double-precision arithmetic. Cubature rules (128 decimal digits of precision) for the tetrahedron and the pyramid from $p = 2$ up to $p = 20$ are provided in the supplementary materials.

The structure of the reminder of this article follows. The formulation is described in Section 2, wherein first we present the reference elements (tetrahedron and pyramid) in Section 2.1, then discuss symmetry relations on the reference elements in Section 2.2, and finally present the constrained nonlinear problem in Section 2.3. The main elements of the five-phase cubature algorithm are described in Section 3. The cubature rules obtained in this paper are presented in Section 4, and the presentation and discussion of verification and convergence tests are conducted in Section 5. Finally, the main findings from this work are provided in Section 6.

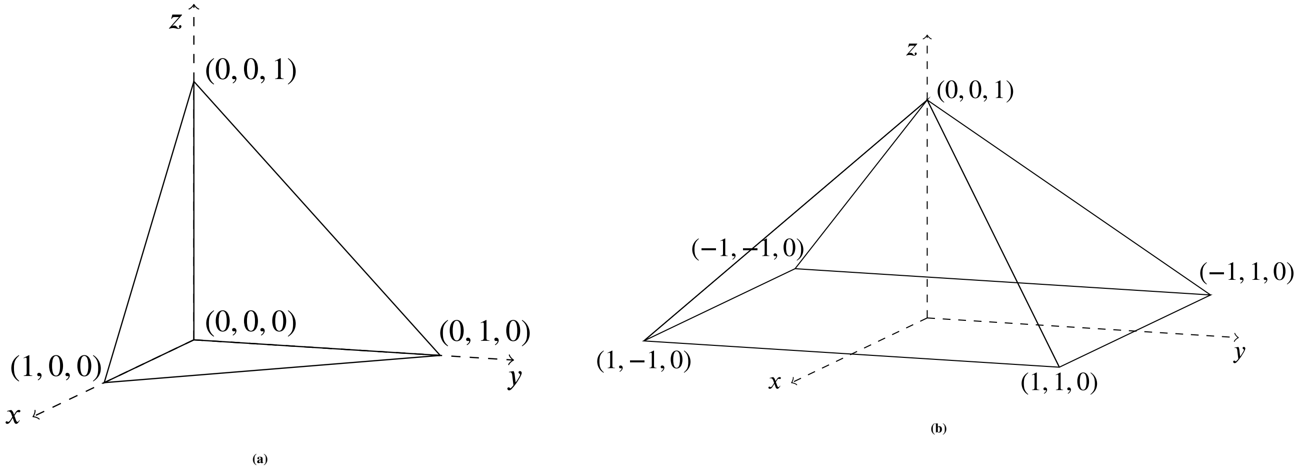


FIGURE 1 Reference elements. (a) Tetrahedron and (b) Pyramid.

2 | FORMULATION

First, the domains used for the reference tetrahedron and pyramid are described. Then, details on the symmetry relations over each reference element are provided, and finally we present the problem of solving the nonlinear polynomial (moment) equations that are subjected to linear equality and linear inequality constraints to generate the cubature schemes.

2.1 | Reference elements

Our objective is to obtain symmetric cubature schemes over a tetrahedron and a pyramid. As problem domains, we consider the reference tetrahedron $\boldsymbol{T}$ and the reference pyramid $\boldsymbol{P}$ shown in Fig. 1. We use monomials as basis functions over these reference domains. The integral of a monomial of order $p = r + s + t$ over $\boldsymbol{T}$ and $\boldsymbol{P}$ is given by[21,22]

$$\int_{T} x^r y^s z^t \, d\boldsymbol{x} = \frac{r!\, s!\, t!}{(r+s+t+3)!}, \qquad \int_{P} x^r y^s z^t \, d\boldsymbol{x} = \begin{cases} \dfrac{4\,(r+s+2)!\, t!}{(r+1)(s+1)(r+s+t+3)!} & \text{if } r \text{ and } s \text{ are both even} \\ 0 & \text{otherwise} \end{cases} . \tag{1}$$

2.2 | Symmetry relations

In order to obtain the symmetric arrangement of the integration points in the tetrahedron, it is convenient to use barycentric coordinates.[26,21,4] If we assign $\boldsymbol{v}_i$ as the coordinate of the i-th vertex of the reference tetrahedron (see Fig. 1a), then a point $\boldsymbol{x} \in T$ can be expressed using the barycentric coordinates, $\lambda = (\lambda_1,\, \lambda_2,\, \lambda_3,\, \lambda_4)^T$, as

$$\boldsymbol{x} = \sum_{i=1}^{4} \lambda_i \boldsymbol{v}_1, \quad \sum_{i=1}^{4} \lambda_i = 1, \quad (0 \le \lambda_i \le 1).$$

For the reference tetrahedron T, the barycentric coordinates and Cartesian coordinates are related by

$$\boldsymbol{x} = \tilde{\boldsymbol{D}}\boldsymbol{\lambda}, \qquad \tilde{\boldsymbol{D}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2a}$$

$$\boldsymbol{\lambda} = \tilde{\boldsymbol{B}}\boldsymbol{x} + \tilde{\boldsymbol{r}}, \qquad \tilde{\boldsymbol{B}} = \begin{bmatrix} -1 & -1 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \tilde{\boldsymbol{r}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2b}$$

We note the following identities:

$$\tilde{\boldsymbol{B}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{B}} = \tilde{\boldsymbol{B}}, \qquad \tilde{\boldsymbol{D}}\tilde{\boldsymbol{B}}\tilde{\boldsymbol{D}} = \tilde{\boldsymbol{D}}, \qquad \tilde{\boldsymbol{D}}\tilde{\boldsymbol{r}} = \boldsymbol{0}. \tag{3}$$

The cubature scheme in T is symmetric if it is invariant under permutations of the barycentric coordinates of the cubature points. This implies that if there is a cubature point expressed in barycentric coordinates that is associated with the weight w then other points generated by all possible permutations of the barycentric coordinates must also be associated with the same weight w. Such permutations naturally divide the integration points into symmetry orbits S_i^T, where each orbit contains all the points generated by its permutation star. The permutation star of an orbit is an m-tuple of positive real numbers (with possible repetitions), which when permuted yield symmetric points that belong to the orbit. If a point in a reference element is represented in Cartesian coordinates then $m = 3$, and if it is given in barycentric (homogeneous) coordinates then $m = 4$ with only three of them being independent since they must sum to unity. To further clarify, we present the construction of the permutation star for the tetrahedron. Let the barycentric coordinates of a point in the reference tetrahedron be represented by the 4-tuple (N_1, N_2, N_3, N_4), where $N_1 + N_2 + N_3 + N_4 = 1$. If all entries are the same, then the permutation star is unique, $(1/4, 1/4, 1/4, 1/4)$ (centroid of the element), and is the sole member of S_1^T. Now, consider the barycentric coordinate of a point with one free variable a (repeated three times). The corresponding permutation star is $([a, a, a, 1 - 3a])$, and on permuting the entries within the square brackets it yields the orbit $S_2^T = \{(a, a, a, 1 - 3a), (a, a, 1 - 3a, a), (a, 1 - 3a, a, a), (1 - 3a, a, a, a)\}$. If the number of points that belong to S_i^T is denoted by $|S_i^T|$, then we have $|S_1^T| = 1$ and $|S_2^T| = 4!/3! = 4$. The remaining permutation stars are: $([a, a, 1/2 - a, 1/2 - a])$, $([a, a, b, 1 - 2a - b])$ and $([a, b, c, 1 - a - b - c])$ for a total of five permutation stars in a tetrahedron. These five permutation stars in the tetrahedron (see Table 1) were proposed by Felippa,[24] and later also adopted by others.[21,4,25] In this article, we use these five permutation stars for generating orbits in the cubature schemes.The search for the cubature schemes is an iterative process in which the parameters in the permutation stars have to be found for each orbit. Note that $S_1^T \subset S_2^T \subset S_3^T \subset S_4^T \subset S_5^T$. Hence, during the iterative procedure, it might occur that parameters a and b in S_5^T

TABLE 1 Tetrahedron symmetry orbits with barycentric coordinates.The entries within the square brackets are permuted.

symmetry orbits	permutation star in barycentric coordinates	$\lvert S_i^T \rvert$
S_1^T	$\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)$	1
S_2^T	$\left([a,\ a,\ a,\ 1-3a]\right)$	4
S_3^T	$\left([a,\ a,\ \frac{1}{2}-a,\ \frac{1}{2}-a]\right)$	6
S_4^T	$\left([a,\ a,\ b,\ 1-2a-b]\right)$	12
S_5^T	$\left([a,\ b,\ c,\ 1-a-b-c]\right)$	24

are almost the same. If so, then this orbit is replaced by S_4^T. This same scenario is applicable to other orbits for the tetrahedron and the pyramid.

The number of integration points in the tetrahedral symmetric cubature rule is given as a linear combination of the symmetry orbits:

$$n = \sum_{i=1}^{5} n_i \lvert S_i^T \rvert, \tag{4}$$

where n_i is the number of i-th symmetry orbits. It should be noted that the position of S_1^T is fixed, so $n_1 = \{0, 1\}$ can only assume two values. Even though the position of the point in the orbit S_1^T is fixed, the associated weight has to be determined. This leaves this orbit with *one free variable*. For orbits S_2^T and S_3^T, besides their weights an additional variable needs to be determined, so they have two free variables. Proceeding likewise, the orbits S_4^T and S_5^T have three and four free variables, respectively. The symmetric cubature rules consist of the set of symmetry orbits, so for unique description of the cubature all these free variables need to be known. All the free variables are represented by the vector $\boldsymbol{s}$ and the total number of free variables is given by

$$n_s = n_1 + 2n_2 + 2n_3 + 3n_4 + 4n_5.$$

For the pyramid shown in Fig. 1b, any point in the element is expressed using Cartesian coordinates. For the pyramidal element, the relevant relations in (3) are trivial since $\tilde{\boldsymbol{D}} = \tilde{\boldsymbol{B}} = \boldsymbol{I}$ and $\tilde{\boldsymbol{r}} = \boldsymbol{0}$. For this element, the symmetry orbits $\{S_i^P\}_{i=1}^4$ are listed in Table 2. The relation for the number of integration points is analogous to (4). The number of free variables for the pyramid is given by

$$n_s = 2n_1 + 3n_2 + 3n_3 + 4n_4.$$

The cubature scheme, $\{\boldsymbol{x}_i, w_i\}_{i=1}^n$, consists of set of nodes and associated weights. All the unknowns are represented in the extended vector $\boldsymbol{z} = [w_1\ x_1\ y_1\ z_1\ w_2\ x_2\ y_2\ z_2\ \ldots w_n\ x_n\ y_n\ z_n]^T$. On considering (2) as well as the permutations in Tables 1

TABLE 2 Pyramid symmetry orbits.

symmetry orbits	permutation star in global coordinates	$\lvert S_i^P \rvert$
S_1^P	$\big(0,\, 0,\, c\big)$	1
S_2^P	$\big([\pm a,\, 0],\, c\big)$	4
S_3^P	$\big(\pm a,\, \pm a,\, c\big)$	4
S_4^P	$\big([\pm a,\, \pm b],\, c\big)$	8

and 2, the following relations between $\boldsymbol{z}$ and $\boldsymbol{s}$ are established:

$$\boldsymbol{z} = \boldsymbol{B}\boldsymbol{s} + \boldsymbol{r}, \tag{5a}$$

$$\boldsymbol{s} = \boldsymbol{D}\boldsymbol{z}. \tag{5b}$$

For two cases, we now provide explicit forms of the matrices $\boldsymbol{B}$ and $\boldsymbol{D}$, and the vector $\boldsymbol{r}$. For the tetrahedron, let us consider a cubature scheme that has a single S_2^T orbit (see $p = 2$ in Table 3). Then, $n = 4$ (four cubature nodes) and $n_s = |\boldsymbol{s}| = 2$ (two free variables). Letting $\boldsymbol{s} := \{a\ w\}^T$ denote the unknown vector with the free variables, we can write

$$\boldsymbol{B} = \begin{bmatrix} 1 & 1 & -3 & 0 & 1 & -3 & 1 & 0 & -3 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}^T, \quad \boldsymbol{D} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and the vector $\boldsymbol{r} = \big\{0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\big\}^T$. For the pyramid, we consider a cubature scheme with the two orbits S_1^P and S_3^P (see $p = 2$ in Table 5). Here, $n = 5$ and $n_s = |\boldsymbol{s}| = 5$. With $\boldsymbol{s} := \{c_{1P}\ w_{1P}\ a_{3P}\ c_{3P}\ w_{3P}\}^T$, we can write the relevant matrices as

$$\boldsymbol{B} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}^T, \quad \boldsymbol{D} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and the vector $\boldsymbol{r} = \boldsymbol{0}$ (20×1 column vector).

The identities presented in (3) are inherited for the matrices $\boldsymbol{B}$ and $\boldsymbol{D}$, and hence

$$\boldsymbol{B}\boldsymbol{D}\boldsymbol{B} = \boldsymbol{B}, \qquad \boldsymbol{D}\boldsymbol{B}\boldsymbol{D} = \boldsymbol{D}, \qquad \boldsymbol{D}\boldsymbol{r} = \boldsymbol{0}. \tag{6}$$

The two relations in (5) are combined to provide another relation for the global vector $\boldsymbol{z}$:

$$\boldsymbol{G}\boldsymbol{z} = \boldsymbol{r}, \qquad \boldsymbol{G} = \boldsymbol{I} - \boldsymbol{B}\boldsymbol{D}. \tag{7}$$

The matrices $\boldsymbol{B}$, $\boldsymbol{D}$ and $\boldsymbol{G}$ consist mostly of zeros, and since the nonzero values are integers, they are easy to construct and operate on. Equation (7) defines the constraints for the vector $\boldsymbol{z}$ so that the points in the tetrahedron or pyramid are arranged in a fully symmetric manner. The matrix $\boldsymbol{G}$ is singular and the pair $(\boldsymbol{G}, \boldsymbol{r})$ represents the consistent set of equations. This means that some of the equations in (7) are linearly dependent. These dependent equations are removed from the system of equations via a modified Gauss-Jordan (MGJ) elimination procedure,[1] which results in the modified system of equations

$$\bar{\boldsymbol{G}}\boldsymbol{z} = \bar{\boldsymbol{r}}. \tag{8}$$

Since (8) has fewer equations than unknowns, its solution is nonunique. The vectors that satisfy (7) and (8) are grouped in the following set:

$$S_c = \left\{ \boldsymbol{y} : ||\boldsymbol{G}\boldsymbol{y} - \boldsymbol{r}|| = 0 \right\} = \left\{ \boldsymbol{y} : ||\bar{\boldsymbol{G}}\boldsymbol{y} - \bar{\boldsymbol{r}}|| = 0 \right\}. \tag{9}$$

To prepare us for what lies ahead, we define another set $S_c^0 = \{\boldsymbol{y} : \boldsymbol{y} \in \ker(\boldsymbol{G})\} = \{\boldsymbol{y} : \boldsymbol{y} \in \ker(\bar{\boldsymbol{G}})\}$, which defines the null space of $\boldsymbol{G}$ and also $\bar{\boldsymbol{G}}$. The matrix $\bar{\boldsymbol{G}}$ is used to define the projection matrix[27]

$$\boldsymbol{P} = \boldsymbol{I} - \bar{\boldsymbol{G}}^{\mathrm{T}} \left(\bar{\boldsymbol{G}}\bar{\boldsymbol{G}}^{\mathrm{T}} \right)^{-1} \bar{\boldsymbol{G}}, \tag{10}$$

which is used to project the incremental vector onto the null space of $\bar{\boldsymbol{G}}$ (constrained symmetry space). We now present three lemmas that are used later on.

Lemma 1. If a vector $\boldsymbol{u}$ is defined as $\boldsymbol{u} = \boldsymbol{B}\boldsymbol{v} + \boldsymbol{r}$, then

$$\boldsymbol{u} \in S_c \quad \forall \boldsymbol{v}.$$

Proof. If the vector $\boldsymbol{u}$ belongs to the set S_c, it suffices to show that $\boldsymbol{G}\boldsymbol{u} - \boldsymbol{r} = \boldsymbol{0}$. To this end, on using the definition of the matrix $\boldsymbol{G}$ given in (7) and the relations in (6), we can write

$$\boldsymbol{G}\boldsymbol{u} - \boldsymbol{r} = \boldsymbol{G}\boldsymbol{B}\boldsymbol{v} + \boldsymbol{G}\boldsymbol{r} - \boldsymbol{r} = \boldsymbol{B}\boldsymbol{v} - \boldsymbol{B}\boldsymbol{D}\boldsymbol{B}\boldsymbol{v} + \boldsymbol{r} - \boldsymbol{B}\boldsymbol{D}\boldsymbol{r} - \boldsymbol{r} = \boldsymbol{B}\boldsymbol{v} - \boldsymbol{B}\boldsymbol{v} = \boldsymbol{0}. \qquad \square$$

Lemma 2. If a vector $\boldsymbol{u}$ is defined as $\boldsymbol{u} = \boldsymbol{B}\boldsymbol{v}$, then

$$\boldsymbol{u} \in S_c^0 \quad \forall \boldsymbol{v}.$$

Proof. If the vector $\boldsymbol{u}$ belongs to the set S_c^0, then it suffices to show that $\boldsymbol{G}\boldsymbol{u} = \boldsymbol{0}$. To this end, on using the definition of the matrix $\boldsymbol{G}$ given in (7) and the relations in (6), we can write

$$\boldsymbol{G}\boldsymbol{u} = \boldsymbol{G}\boldsymbol{B}\boldsymbol{v} = \boldsymbol{B}\boldsymbol{v} - \boldsymbol{B}\boldsymbol{D}\boldsymbol{B}\boldsymbol{v} = \boldsymbol{B}\boldsymbol{v} - \boldsymbol{B}\boldsymbol{v} = \boldsymbol{0}. \qquad \square$$

Lemma 3. If a vector $\boldsymbol{u}$ is defined as $\boldsymbol{u} = \boldsymbol{P}\boldsymbol{v}$, then

$$\boldsymbol{u} \in S_c^0 \quad \forall \boldsymbol{v}.$$

Proof. If the vector $\boldsymbol{u}$ belongs to the set S_c^0, it suffices to show that $\bar{\boldsymbol{G}}\boldsymbol{u} = \boldsymbol{0}$. To this end, on using the definition of the projection matrix given in (10), we can write

$$\bar{\boldsymbol{G}}\boldsymbol{u} = \bar{\boldsymbol{G}}\boldsymbol{P}\boldsymbol{v} = \left[\bar{\boldsymbol{G}} - \bar{\boldsymbol{G}}\bar{\boldsymbol{G}}^{\mathrm{T}}\left(\bar{\boldsymbol{G}}\bar{\boldsymbol{G}}^{\mathrm{T}}\right)^{-1}\bar{\boldsymbol{G}}\right]\boldsymbol{v} = \left[\bar{\boldsymbol{G}} - \bar{\boldsymbol{G}}\right]\boldsymbol{v} = \boldsymbol{0}. \qquad \square$$

2.3 | Constrained moment problem

The integration of an arbitrary function $f(\boldsymbol{x})$ over the domain Ω is given by a cubature formula of the form

$$\int_{\Omega} f(\boldsymbol{x})\, d\boldsymbol{x} \approx |\Omega| \sum_{i=1}^{n} f(\boldsymbol{x}_i) w_i, \tag{11}$$

where the pair $\{\boldsymbol{x}_i, w_i\}_{i=1}^n$ represents the cubature scheme with $\boldsymbol{x}_i$ the i-th integration point (node), w_i is the i-th weight, n is the number of integration points, and $|\Omega|$ represents the volume of the domain. We apply (11) to the tetrahedral and pyramidal reference elements shown in Fig. 1. The volumes $|\Omega|$ of the tetrahedral and pyramidal reference elements are $1/6$ and $4/3$, respectively. For a given p, the goal is to find the cubature scheme with the minimal number of nodes, along with the position of each node and its associated weight that gives the exact integral for any p-order polynomial. In this article, we require that the p-order cubature scheme meets the following three requirements: (i) weights are positive, (ii) nodes reside in the interior of the domain, and (iii) cubature scheme is fully symmetric. The two first requirements are clear. A cubature scheme being fully symmetric means that it is invariant to affine transformation of the domain Ω. The solution of this problem is nonunique, because there can exist distinct p-order cubature rules that satisfy the constrained problem. In our previous work,[1] high-order non-symmetric cubature schemes were proposed with minimal number of nodes that were close to the lower bound estimate. The symmetry constraints lead to more number of integration points in the symmetric cubature rules in comparison to the non-symmetric case.

In order to find p-order cubatures, we have to find the solution of the following constrained moment problem:

$$\boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{0}, \tag{12a}$$

subject to the linear inequality and linear equality constraints

$$\boldsymbol{A}\boldsymbol{z} < \boldsymbol{b}, \tag{12b}$$

$$\bar{\boldsymbol{G}}\boldsymbol{z} = \bar{\boldsymbol{r}}, \tag{12c}$$

where (12a) are polynomial (moment) equations, the pair $(\boldsymbol{A}, \boldsymbol{b})$ represents algebraic constraints for the inside nodes and positive weights, and the pair $(\bar{\boldsymbol{G}}, \bar{\boldsymbol{r}})$ represents the symmetric constraints. We define the vector $\boldsymbol{f}$ to represent the error of the moment equations:

$$\boldsymbol{f} := \boldsymbol{M}\boldsymbol{w} - \boldsymbol{p}, \tag{13a}$$

where

$$\boldsymbol{M} = |\Omega| \begin{bmatrix} \Phi_1(\boldsymbol{x}_1) & \Phi_1(\boldsymbol{x}_2) & \dots & \Phi_1(\boldsymbol{x}_n) \\ \Phi_2(\boldsymbol{x}_1) & \Phi_2(\boldsymbol{x}_2) & \dots & \Phi_2(\boldsymbol{x}_n) \\ \dots & \dots & \dots & \dots \\ \Phi_\ell(\boldsymbol{x}_1) & \Phi_\ell(\boldsymbol{x}_2) & \dots & \Phi_\ell(\boldsymbol{x}_n) \end{bmatrix}, \quad \boldsymbol{w} = \begin{Bmatrix} w_1 \\ w_2 \\ \dots \\ w_m \end{Bmatrix}, \quad \boldsymbol{p} = \begin{Bmatrix} \int_\Omega \Phi_1(\boldsymbol{x})\, d\boldsymbol{x} \\ \int_\Omega \Phi_2(\boldsymbol{x})\, d\boldsymbol{x} \\ \dots \\ \int_\Omega \Phi_\ell(\boldsymbol{x})\, d\boldsymbol{x} \end{Bmatrix}. \tag{13b}$$

In (13b), Φ_i $(i = 1, 2, \dots, \ell)$ are the basis functions, which are chosen as monomials and ℓ represents the number of basis functions. The number of basis functions, $\ell = (p+1)(p+2)(p+3)/6$, which consists of all monomials up to order p in 3D. The problem in (12) is strongly nonlinear, and an iterative procedure based on Newton's method is applied. Equation (1) is used to precompute the integrals of the monomials in multi-precision arithmetic (160 digits) to form $\boldsymbol{p}$ in (13), and the results are stored in a data file.

For the cubature problem, some authors have used the symmetry requirements to reduce the size of the problem domain and the number of unknowns.[20,21,4,28] In these contributions, symmetry is applied to the nodal positions and the basis functions, so that in the reduced-scale problem only the unknowns in $\boldsymbol{s}$ (fewer than in $\boldsymbol{z}$) need to be determined. In this article, we follow the approach in Jaśkowiec and Sukumar,[1] and apply preconditioning to the Jacobian matrix. This operation destroys the symmetry of the basis functions. Therefore, we consider the full domain (T and P) on which the symmetry constraints (12c) are enforced on the unknown vector $\boldsymbol{z}$ at each iteration step of the algorithm. For example, in a $p = 15$ cubature rule, the reduced-scale problem has 56 unknowns whereas the full-scale problem has 1056 unknowns. We obtain robustness for higher p but at greater computational costs, but this is not a significant limitation since it is a one-time cost. In Section 3, we present the algorithm for solving (12), where the symmetry constraints play an important role.

3 | ALGORITHM FOR SYMMETRIC CUBATURES

To determine the cubature rules, we search for the vector $\boldsymbol{z}$ that satisfies the nonlinear equations (12a) subject to the linear inequality constraints (12b) and the linear equality constraints (12c). The algorithm has five phases, and in each phase an iterative procedure is performed to calculate the increment $\Delta \boldsymbol{z}$ such that

$$\boldsymbol{z}_k = \boldsymbol{z}_{k-1} + \alpha \Delta \boldsymbol{z}, \quad \boldsymbol{z}_{k-1},\, \boldsymbol{z}_k \in S_c, \tag{14}$$

where $k-1$ and k are iteration step counters, and α is the step size. On combining (7) with (14) it can be shown that the vector increment $\Delta \boldsymbol{z}$ belongs to S_c^0.

At every phase of the algorithm, the increment $\Delta \boldsymbol{z}$ in (14) is computed using a modified Newton procedure. The step size α is set to 0.1 (damped or guarded Newton) for phases one to three, and $\alpha = 1$ (full Newton) for phases four and five. However, to improve the convergence rate in the third phase, the optimal α can be determined as presented in Jaśkowiec and Sukumar.[1] The Jacobian matrix $\boldsymbol{J}_k = \partial \boldsymbol{f}_k / \partial \boldsymbol{z}$ needs to be calculated at every iteration step. The pair $(\boldsymbol{J}_k, \boldsymbol{f}_k)$ is calculated using the monomial basis, which if used as is would lead to ill-conditioning of the Jacobian matrix as p is increased. To resolve this issue, we follow Jaśkowiec and Sukumar[1] and precondition the Newton equations using the MGJ procedure, which results in the modified well-conditioned pair $(\bar{\boldsymbol{J}}_k, \bar{\boldsymbol{f}}_k)$. To solve (12), we seek the increment $\Delta \boldsymbol{z}$ that minimizes the Newton residual and satisfies the linear inequality and linear equality constraints. For the so-called extended version of the problem, the full vector $\Delta \boldsymbol{z}$ is found, whereas in the reduced version of the problem, Δs (increment of the free variables vector) is firstly found, and then $\Delta \boldsymbol{z}$ is set. The extended version is used in phases two to four, and the reduced version is used in phase one and phase five. In all phases, various types of constrained minimization problems arise. For constrained linear least-squares problems, the `Matlab` function `lsqlin` is utilized.

The flowchart of the five-phase algorithm is shown in Fig. 2. In all phases symmetry is enforced on the vector $\boldsymbol{z}_k$ at every iteration step. To initiate the algorithm, we first choose the set of orbits and then randomly generate points in the orbits. This set of points is sent to phase one of the algorithm. The role of phase one is to reposition the nodes to be suitable for the next phase. For phases one to five, if the convergence criteria is met the computations proceed to the subsequent phase; if not, one more orbit is added and the algorithm restarts from phase one. In phases one to three the inequality constraint (12b) is enforced (also checked at each iteration step), but in the fourth and fifth phases it is not. If phase four terminates successfully, then we proceed to phase five. The fifth phase uses high-precision calculations, and is time-consuming. Hence, it is instructive to search for a more efficient scheme that has fewer nodes by removing one of the symmetry orbits and repeating the algorithm from phase two. When the cubature with minimal possible number of integration points is found, then the final phase five is invoked to calculate the cubature rules with high precision. We assign a prescribed error tolerance ϵ_i ($i = 1, 2, \dots, 5$) to each phase of the algorithm, and within each iteration step of the i-th phase, convergence is checked via $||\boldsymbol{f}|| < \epsilon_i$. The values of ϵ_i are set individually for each cubature rule. We know they must be monotonically decreasing with each phase, but there is no simple means to choose a specific value. In phase five, which is done using variable-precision floating-point arithmetic (VPA), ϵ_5 is set to 10^{-160}.

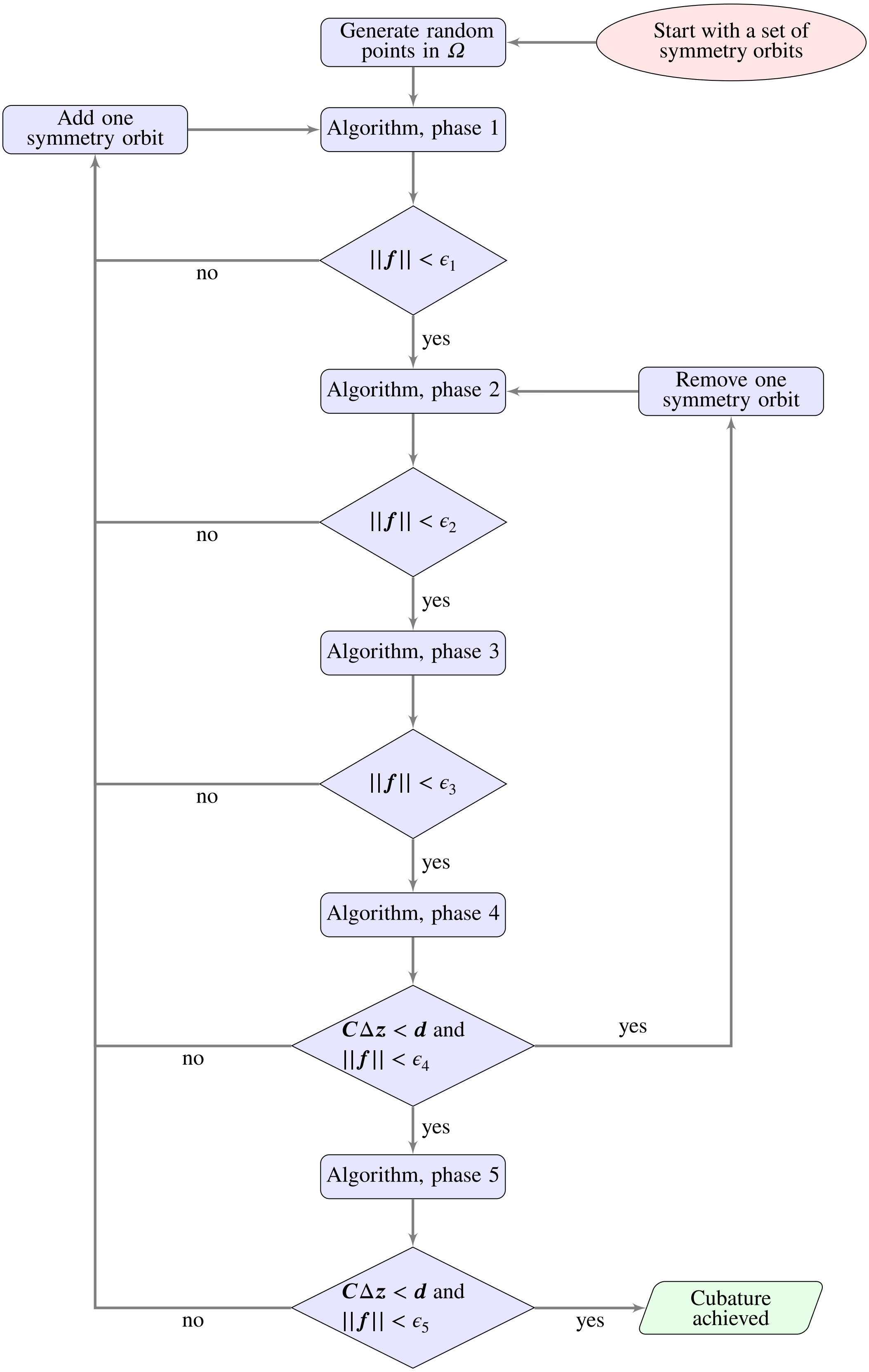

Generate random points in Ω
Start with a set of symmetry orbits
Add one symmetry orbit
Algorithm, phase 1
$||\boldsymbol{f}|| < \epsilon_1$
no
yes
Algorithm, phase 2
Remove one symmetry orbit
$||\boldsymbol{f}|| < \epsilon_2$
no
yes
Algorithm, phase 3
$||\boldsymbol{f}|| < \epsilon_3$
no
yes
Algorithm, phase 4
$\boldsymbol{C}\Delta\boldsymbol{z} < \boldsymbol{d}$ and $||\boldsymbol{f}|| < \epsilon_4$
no
yes
yes
Algorithm, phase 5
$\boldsymbol{C}\Delta\boldsymbol{z} < \boldsymbol{d}$ and $||\boldsymbol{f}|| < \epsilon_5$
no
yes
Cubature achieved

FIGURE 2 Flowchart of the symmetric cubature algorithm.

3.1 | Phase one

In this phase, we solve the constrained minimization problem

$$\min_{\Delta \boldsymbol{z}} \frac{1}{2} || \boldsymbol{J}_k \Delta \boldsymbol{z} + \boldsymbol{f}_k ||^2, \tag{15a}$$

subject to the following linear inequality and linear equality constraints

$$\boldsymbol{C} \Delta \boldsymbol{z} < \boldsymbol{d}, \tag{15b}$$

$$\bar{\boldsymbol{G}} \Delta \boldsymbol{z} = \boldsymbol{0}, \tag{15c}$$

where the pair $(\boldsymbol{C},\ \boldsymbol{d})$ is constructed to apply the inequality constraints (12b) for the increment vector $\Delta \boldsymbol{z}$. In (15), the vector $\Delta \boldsymbol{z}$ can be expressed as

$$\Delta \boldsymbol{z} = \boldsymbol{B} \Delta \boldsymbol{s}. \tag{16}$$

On using Lemma 2, we have $\bar{\boldsymbol{G}} \Delta \boldsymbol{z} = \boldsymbol{0}$ for every vector $\Delta \boldsymbol{s}$. Let $\boldsymbol{J}_k^r = \boldsymbol{J}_k \boldsymbol{B}$ and $\boldsymbol{C}^r = \boldsymbol{C}\boldsymbol{B}$. The matrix $\boldsymbol{J}_k^r$ is rectangular with greater number of rows than columns and the pair $(\boldsymbol{J}_k^r, \boldsymbol{f}_k)$ represents the singular, consistent equation. This pair is preprocessed using the MGJ procedure,[1] so that all equations that are linearly dependent are removed from the pair, which results in the modified form $(\bar{\boldsymbol{J}}_k^r, \bar{\boldsymbol{f}}_k)$. Now, the constrained minimization problem is:

$$\min_{\Delta \boldsymbol{s}} \frac{1}{2} || \bar{\boldsymbol{J}}_k^r \Delta \boldsymbol{s} + \bar{\boldsymbol{f}}_k ||^2, \tag{17a}$$

subject to the linear inequality constraints

$$\boldsymbol{C}^r \Delta \boldsymbol{s} < \boldsymbol{d}. \tag{17b}$$

3.2 | Phase two

In this phase, the vector $\Delta \boldsymbol{z}$ is calculated as

$$\Delta \boldsymbol{z} = \boldsymbol{P} \Delta \bar{\boldsymbol{z}}, \tag{18}$$

where $\Delta \bar{\boldsymbol{z}}$ is the solution of the following minimization problem:

$$\min_{\Delta \bar{\boldsymbol{z}}} \frac{1}{2} || \bar{\boldsymbol{J}}_k \Delta \bar{\boldsymbol{z}} + \bar{\boldsymbol{f}}_k ||^2, \tag{19a}$$

subject to the linear inequality constraints

$$\boldsymbol{C} \Delta \bar{\boldsymbol{z}} < \boldsymbol{d}, \tag{19b}$$

where the pair $(\bar{\boldsymbol{J}}_k,\ \bar{\boldsymbol{f}}_k)$ is the preprocessed pair $(\boldsymbol{J}_k,\ \boldsymbol{f}_k)$ using MGJ.[1] The vector $\Delta\boldsymbol{z}$ calculated in (18) belongs to S_c^0, which is established using Lemma 3.

In this phase the inequality constraints are enforced on the $\Delta\bar{\boldsymbol{z}}$, and then this vector is projected onto the symmetry subspace to obtain the vector $\Delta\boldsymbol{z}$, see (18). It may happen that during the iteration procedure some points in the vector $\boldsymbol{z}_k$ from (14) go outside the tetrahedron, but after a few iterations they return to be within the element. According to Lemma 3, we know that the vector $\Delta\boldsymbol{z} \in S_c^0$, and so the vector $\boldsymbol{z}_k$ keeps the symmetry of the points in the domain in each iteration step. The projection operation in (18) is performed with machine-precision (i.e., double-precision herein) accuracy. As the iterations proceed, truncation errors may accumulate and the symmetry in the vector $\boldsymbol{z}$ may be violated, especially for high-order problems. In order to circumvent this issue, using Lemma 1, we perform the following simple procedure after each iteration:

$$\boldsymbol{s}_k = \boldsymbol{D}\boldsymbol{z}_k, \quad \boldsymbol{z}_k = \boldsymbol{B}\boldsymbol{s}_k + \boldsymbol{r}. \tag{20}$$

3.3 | Phase three

In this phase, the vector $\Delta\boldsymbol{z}$ is obtained using the same projection as in the previous phase, see (18). Also in this case to get rid of the influence of the truncation error during the projection, the procedure in (20) is performed. In this phase, $\Delta\bar{\boldsymbol{z}}$ is the solution of the following constrained quadratic minimization problem:

$$\min_{\Delta\bar{\boldsymbol{z}}} \frac{1}{2}||\Delta\bar{\boldsymbol{z}}||^2, \tag{21a}$$

subject to the following linear inequality and linear equality constraints:

$$\boldsymbol{C}\Delta\bar{\boldsymbol{z}} < \boldsymbol{d}, \quad \bar{\boldsymbol{J}}_k\Delta\bar{\boldsymbol{z}} = -\bar{\boldsymbol{f}}_k. \tag{21b}$$

3.4 | Phase four

The increment $\Delta\boldsymbol{z}$ is obtained in a similar manner to that in the previous two phases, and the symmetry corrections are also implemented. For the vector $\Delta\bar{\boldsymbol{z}}$, the following quadratic minimization problem is posed:

$$\min_{\Delta\bar{\boldsymbol{z}}} \frac{1}{2}||\Delta\bar{\boldsymbol{z}}||^2, \tag{22a}$$

subject to the linear equality constraints

$$\bar{\boldsymbol{J}}_k\Delta\bar{\boldsymbol{z}} = -\bar{\boldsymbol{f}}_k. \tag{22b}$$

The problem posed in (22) has an exact (algebraic) solution:

$$\Delta \bar{\boldsymbol{z}} = -\bar{\boldsymbol{J}}_k^{\mathrm{T}} \left(\bar{\boldsymbol{J}}_k \bar{\boldsymbol{J}}_k^{\mathrm{T}} \right)^{-1} \bar{\boldsymbol{f}}_k. \tag{23}$$

In this phase, condition (12b) is not enforced, but is checked after each iteration. Up to this phase, all calculations are done in double precision. Though a rare occurrence, if it so happens that during the iterations any point goes outside the reference element (tetrahedron or pyramid), or any weight is negative, then the whole algorithm is considered to have failed, and the algorithm is restarted.

3.5 | Phase five

The previous four phases are performed using double-precision arithmetic. After the fourth phase, we often have a cubature scheme with an accuracy of $\mathcal{O}(10^{-16})$ for small p and about $\mathcal{O}(10^{-6})$ for $p > 15$. The goal of this phase is to improve the accuracy of the integration scheme. To this end, VPA is applied in this phase with 160 digits of precision. The calculations using VPA arithmetic are very time-consuming and so the number of iterations should be limited to only a few. Thus, it is important that the algorithm converges fast to achieve an accuracy of $\mathcal{O}(10^{-160})$ for each p.

In this phase, the vector $\Delta \boldsymbol{z}$ is calculated using (16), where $\Delta \boldsymbol{s}$ is the solution of the following constrained quadratic minimization problem:

$$\min_{\Delta \boldsymbol{s}} \frac{1}{2} ||\Delta \boldsymbol{s}||^2, \tag{24a}$$

subject to the linear equality constraints

$$\bar{\boldsymbol{J}}_k^r \Delta \boldsymbol{s} = -\bar{\boldsymbol{f}}_k. \tag{24b}$$

The solution of (24) is of the same form as (23), and is given by

$$\Delta \boldsymbol{s} = -(\bar{\boldsymbol{J}}_k^r)^{\mathrm{T}} \left(\bar{\boldsymbol{J}}_k^r (\bar{\boldsymbol{J}}_k^r)^{\mathrm{T}} \right)^{-1} \bar{\boldsymbol{f}}_k. \tag{25}$$

4 | SYMMETRIC CUBATURE SCHEMES

The proposed cubature algorithm is implemented to generate p-order cubature schemes ($p = 2$ to $p = 20$) on a tetrahedron and pyramid. During phases 1–4, the computations are done using double-precision arithmetic, whereas high-precision (160 digits) arithmetic is applied in the fifth phase.

The algorithm is started by choosing the combination of symmetry orbits S_i, and then randomly scattered points in the orbits are generated so that they all are in the interior of the domain. For moderate p, convergence is fast, but for $p > 15$, the algorithm becomes sensitive to the location of the starting points. Thus, for higher orders the distribution of initial points are selected using the integration points from two previously generated lower order cubature schemes. It takes about 40 min to obtain the cubature rule for $p = 10$ using double-precision calculations, whereas it takes about 2 days to generate the cubature rule for $p = 20$. For the high-precision calculations using VPA (160 decimal digits) in `Matlab`, it is sufficient to perform about 7 iterations. However, these calculations are time-consuming. For example, an iteration for $p = 10$ takes 40 min, whereas for $p = 20$, one iteration takes over 30 h on a multiprocessor server. All cubature weights are positive, and the value of a weight defines the influence of the corresponding node. Thus, it is desirable that their values are not very small. The quality of the cubature scheme can be assessed by the value of the following parameter:

$$r_w = \frac{\min(w_i)}{\max(w_i)}.$$

It is desirable that r_w is as close to unity as possible, but this is difficult to achieve in practice and typically worsens with increase in p. A very small value of r_w indicates that the influence of some nodes in the cubature scheme is diminished.

In Table 3, the number of integration points in a tetrahedron for different p is listed using our scheme and those presented in Witherden and Vincent[4] and Zhang et al.[21] The table also shows the combinations of symmetry orbits as well as the value of r_w for all schemes. Up to $p = 10$, the number of integration points for our schemes are the same as those of Witherden and Vincent,[4] though the value of r_w for our schemes is the same or higher. For $p = 9$, our results also match for the combination of symmetry orbits; however, for $p = 10$ we obtain different combination of orbits but with the same number of integration points. Our cubature schemes for $p = 2$, $p = 5$ and $p = 6$ exactly matches those of Witherden and Vincent.[4] Furthermore, we have generated rules for $p = 11$ to $p = 20$, and besides $p = 14$, all other rules are new additions to the literature. Note that for $p = 14$, we obtain fewer number of integration points than Zhang et al.[21] In Table 4, we compare our results for the number of integration points in the symmetric tetrahedral cubature scheme to the non-symmetric rules presented in Jaśkowiec and Sukumar.[1] The lower bound estimate for the number of integration points is also indicated. It is seen that placing the symmetry requirements results in greater number of nodes. On average, the symmetric cubature rule has about 20 percent more nodes than the non-symmetric cubature rule.

The analogous listing of Table 3 for the pyramid (see Fig. 1b) is presented in Table 5. For $p = 2$ to $p = 10$, we have obtained schemes that have the same or fewer number of nodes than those in Witherden and Vincent;[4] for $p = 6$ and $p = 10$, our rules have fewer number of nodes. For most p, we obtain values of r_w that are greater than those of Witherden and Vincent.[4] Our symmetric pyramidal cubature rules from $p = 11$ to $p = 20$ are new additions to the literature.

For the tetrahedral and pyramidal cubature schemes, the location of the cubature nodes for three values of p ($p = 8,\ 16,\ 20$) is depicted in Fig. 3. In Fig. 4, the cubature nodes for $p = 7,\ 13,\ 17$ are presented on the xy-plane, which reveal the symmetry in the position of the nodes. Each node is shown as a ball with the radius of the ball being proportional to the value of the cubature

TABLE 3 Number of integration points and combination of symmetry orbits for various symmetric tetrahedral cubature rules. Our results are compared to the rules presented in Witherden and Vincent[4] and Zhang et al.[21]

	This work							Witherden and Vincent[4] and Zhang et al[21] (in braces)						
p	n	$\lvert S_1^T \rvert$	$\lvert S_2^T \rvert$	$\lvert S_3^T \rvert$	$\lvert S_4^T \rvert$	$\lvert S_5^T \rvert$	r_w	n	$\lvert S_1^T \rvert$	$\lvert S_2^T \rvert$	$\lvert S_3^T \rvert$	$\lvert S_4^T \rvert$	$\lvert S_5^T \rvert$	r_w
2	4	0	1	0	0	0	1.000	4	0	1	0	0	0	1.000
3	8	0	2	0	0	0	0.961	8	0	2	0	0	0	0.835
4	14	0	2	1	0	0	0.711	14	0	2	1	0	0	0.378
5	14	0	2	1	0	0	0.378	14	0	2	1	0	0	0.378
6	24	0	3	0	1	0	0.182	24	0	3	0	1	0	0.182
7	35	1	1	1	2	0	0.085	35	1	1	1	2	0	0.085
8	46	0	4	1	2	0	0.150	46	0	4	1	2	0	0.138
								(46)	(0)	(4)	(1)	(2)	(0)	(0.117)
9	59	1	4	1	3	0	0.040	59	1	4	1	3	0	0.001
10	81	1	2	2	5	0	0.026	81	1	2	0	6	0	0.008
11	110	0	2	3	5	1	0.086	–	–	–	–	–	–	–
12	168	0	3	2	4	4	0.049	–	–	–	–	–	–	–
13	172	0	4	2	6	3	0.023	–	–	–	–	–	–	–
14	204	0	6	4	5	4	0.008	(236)	(0)	(5)	(0)	(4)	(7)	(0.020)
15	264	0	3	2	6	7	0.032	–	–	–	–	–	–	–
16	304	0	4	2	7	8	0.014	–	–	–	–	–	–	
17	364	0	4	4	9	9	0.026	–	–	–	–	–	–	–
18	436	0	7	8	10	10	0.007	–	–	–	–	–	–	–
19	487	1	3	1	13	13	0.013	–	–	–	–	–	–	–
20	552	0	6	2	13	15	0.012	–	–	–	–	–	–	–

TABLE 4 Number of integration points as a function of p for tetrahedral cubature rules. Results are compared to the lower bound estimate, $n_{\text{lb}} = \lceil (p+1)(p+2)(p+3)/24 \rceil$, and to the rules given in Jaśkowiec and Sukumar.[1]

p	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
n_{lb}	3	5	9	14	21	30	41	55	72	91	114	140	170	204	242	285	333	385	443
n^1	4	6	11	14	23	31	44	57	74	94	117	144	175	207	247	288	338	390	448
n	4	8	14	14	24	35	46	59	81	110	168	172	204	264	304	364	436	487	552

weight. It can be observed that nodes with larger weights are located in the vicinity of the center of the element. With increasing distance from the center of the element, the nodal weights decrease. Smallest weights are for nodes located close to the faces of the element. This shows that the nodes located well inside the element have the greatest contribution to the value of the integral, whereas those close to the outer boundaries have much less influence.

The reader can find the listing of the cubature points and weights in the supplementary materials, where data with 128 digits of precision are provided in two forms: compact where only the unique values in the symmetry orbits (free variables) are given (see Tables 1 and 2), and extended with all points and weights given in column-format $x\ y\ z\ w$. Additionally, tetrahedral cubature rules are also provided in barycentric coordinates. The results are given in a text file as well as in MATLAB® binary format.

TABLE 5 Number of integration points and combination of the symmetry orbits for various order of symmetric pyramidal cubature rules. Results are compared to the rules presented in Witherden and Vincent.[4]

	This work						Witherden and Vincent[4]					
p	n	$\lvert S_1^P\rvert$	$\lvert S_2^P\rvert$	$\lvert S_3^P\rvert$	$\lvert S_4^P\rvert$	r_w	n	$\lvert S_1^P\rvert$	$\lvert S_2^P\rvert$	$\lvert S_3^P\rvert$	$\lvert S_4^P\rvert$	r_w
2	5	1	0	1	0	0.643	5	1	1	0	0	0.856
3	6	2	0	1	0	0.560	6	2	0	1	0	0.446
4	10	2	1	1	0	0.307	10	2	1	1	0	0.307
5	15	3	1	2	0	0.271	15	3	1	2	0	0.234
6	23	3	2	3	0	0.081	24	4	2	3	0	0.036
7	31	3	2	5	0	0.133	31	3	2	5	0	0.132
8	47	3	4	5	1	0.061	47	3	3	6	1	0.030
9	62	2	6	5	2	0.083	62	2	4	9	1	0.055
10	80	4	6	7	3	0.014	83	3	3	9	4	0.051
11	103	3	8	11	3	0.110	–	–	–	–	–	–
12	127	3	9	12	5	0.016	–	–	–	–	–	–
13	152	4	8	15	7	0.040	–	–	–	–	–	–
14	184	4	12	17	8	0.023	–	–	–	–	–	–
15	234	2	13	23	11	0.028	–	–	–	–	–	–
16	285	1	13	22	18	0.027	–	–	–	–	–	–
17	319	3	13	28	19	0.009	–	–	–	–	–	–
18	357	1	14	31	22	0.011	–	–	–	–	–	–
19	418	2	17	29	29	0.014	–	–	–	–	–	–
20	489	1	19	31	36	0.009	–	–	–	–	–	–

5 | NUMERICAL TESTS

In this section, we present verification and convergence tests for the cubature rules. The cubature schemes are provided with 128 digits of precision in the supplementary material; however, numerical tests are performed using double-precision arithmetic. The aim of the verification tests is to ensure that the cubature rules deliver close to machine-precision accuracy for the integration of polynomial functions. In the convergence tests, nonpolynomial functions are considered to assess the performance of the cubature rules. For comparison purposes, the same tests are carried out for the cubature rules proposed in Witherden and Vincent[4] (taken from the 'quadpy' project[29]), and the non-symmetric tetrahedral cubature rules from Jaśkowiec and Sukumar.[1] A single verification test is presented in Section 5.1, and in Sections 5.2.1–5.2.4 we consider convergence tests for exponential, weakly singular and trigonometric functions. The weakly singular function test is performed on the reference tetrahedron; all other tests are performed on the biunit cube or unit cube. All cubes are meshed using 12 tetrahedra and 6 pyramids as shown in Fig. 5. In the final convergence test (see Section 5.2.4), integration is performed on the semi-cylindrical domain, which is mathematically generated by the transformation from the biunit cube.[1] For all tests, the relative error is defined as

$$R = \frac{|I - I_q|}{|I|},$$

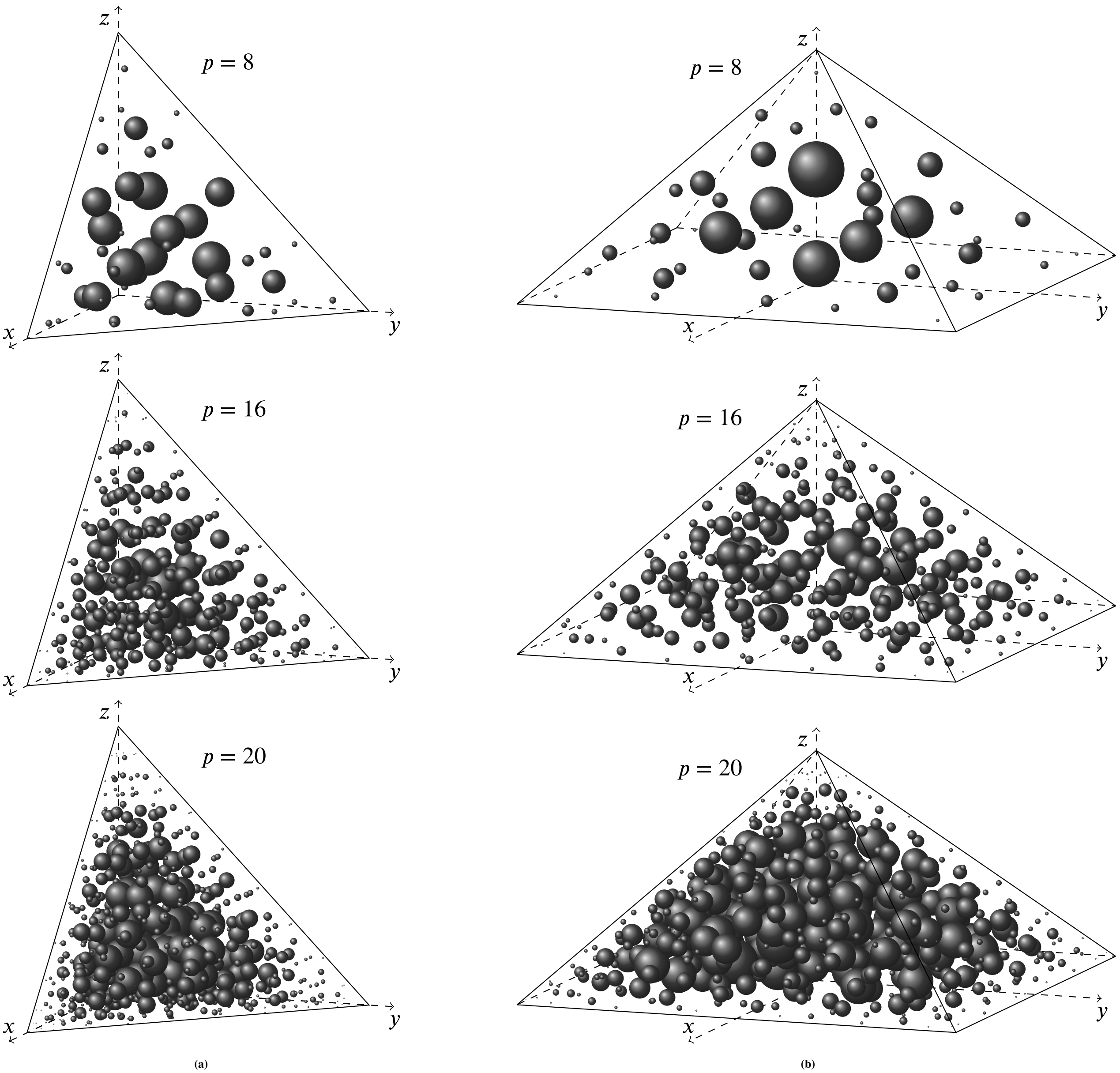


FIGURE 3 Location of cubature nodes for $p = 8$, $p = 16$ and $p = 20$. (a) Tetrahedron and (b) Pyramid. The center of each ball is used to represent the location of the cubature node, and its radius is proportional to the value of the cubature weight.

where I is the exact value of the integral, and I_q is the numerically computed value of the integral using the cubature rule.

5.1 | Verification tests on the biunit cube

For each p, 100 random polynomials $f(\boldsymbol{x})$ are generated and then integrated over the biunit cube, $C = [-1, 1]^3$. We choose the order of $f(\boldsymbol{x})$ to be the same as the order of the cubature to be checked. The accuracy of any cubature rule of order p should be within machine-precision for any polynomial up to order p over a mesh of tetrahedral or pyramidal elements that partitions Ω. The cube is first meshed with 12 tetrahedral elements and secondly by 6 pyramidal elements. The results of the numerical tests

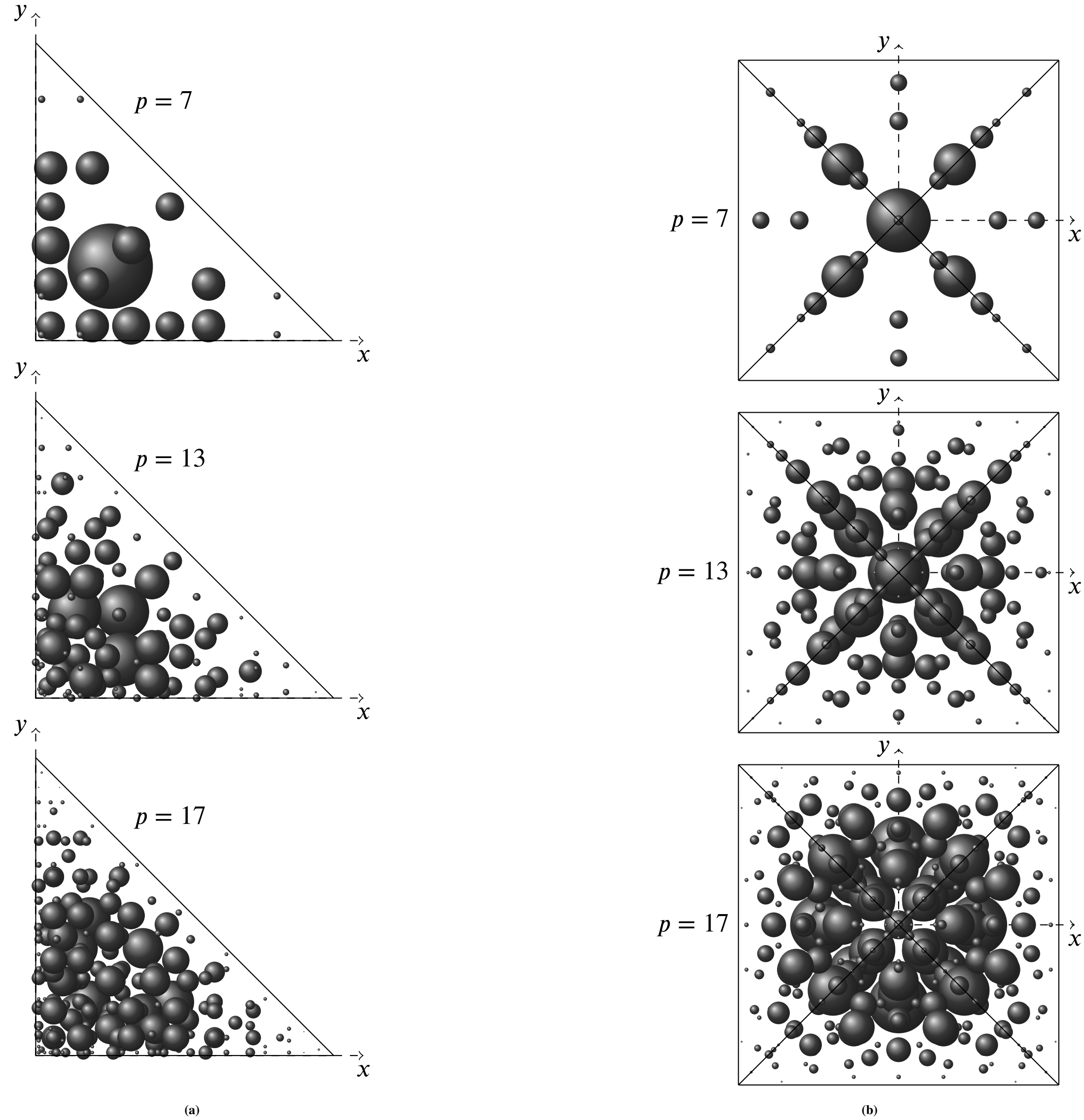


FIGURE 4 Location of cubature nodes on the xy-plane for $p = 7$, $p = 13$ and $p = 17$. (a) Tetrahedron and (b) Pyramid. The center of each ball is used to represent the location of the cubature node, and its radius is proportional to the value of the cubature weight.

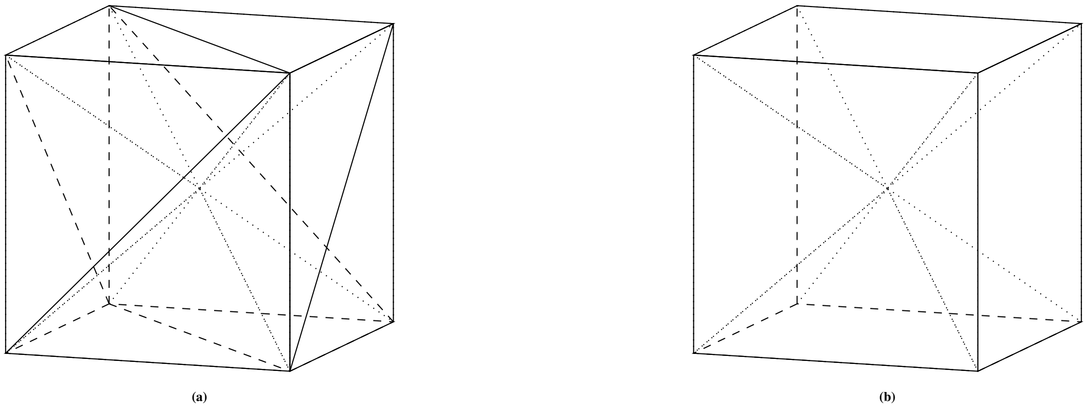


FIGURE 5 Cube meshed by (a) 12 tetrahedra and (b) 6 pyramids.

are presented in Table 6 for tetrahedral elements and in Table 7 for pyramidal elements. The maximum as well as the algebraic mean values of the relative integration errors from a set of 100 tests for each p are presented. It can be observed that the errors for the integration schemes of order $p \leq 10$ generated from this study and those from Witherden and Vincent[4] are of the same magnitude and oscillate about machine precision. For both integration schemes, the maximum error is $\mathcal{O}(10^{-14})$, with mean values of $\mathcal{O}(10^{-15})$. For $p \geq 10$, the relative errors by our schemes are also close to machine precision.

TABLE 6 Integration errors for test on biunit cube that is meshed with tetrahedra. Comparisons are made to results obtained using the schemes presented in Witherden and Vincent.[4] The listing is the maximum and mean values of the error from hundred tests for each p.

,

	This work			Witherden and Vincent[4]		
p	n	max	mean	n	max	mean
		$R \times 10^{16}$			$R \times 10^{16}$	
1	1	11	3	1	16	3
2	4	17	4	4	19	3
3	8	112	5	8	28	5
4	14	47	6	14	82	7
5	14	44	6	14	82	7
6	24	73	6	24	127	9
7	35	109	8	35	91	7
8	46	49	8	46	64	7
9	59	293	11	59	376	10
10	81	534	17	81	532	15
11	110	408	16	–	–	–
12	168	97	7	–	–	–
13	172	551	18	–	–	–
14	204	453	15	–	–	–
15	264	313	19	–	–	–
16	304	238	18	–	–	–
17	364	313	16	–	–	–
18	436	93	9	–	–	–
19	487	141	13	–	–	–
20	552	540	21	–	–	–

5.2 | Convergence tests

Convergence tests using exponential, trigonometric and weakly singular functions are conducted.[1] The test for the exponential function is performed on the biunit cube. The tests for weakly singular functions are done on the reference tetrahedron. One of the trigonometric functions is tested on the unit cube, whereas the other is assessed on the semi-cylindrical domain that also has curved sides. The meshing of the cube using tetrahedra and pyramids is shown in Fig. 5. In these tests, three types of cubature

TABLE 7 Integration errors for test on biunit cube that is meshed with pyramids. Comparisons are made to results obtained using the schemes presented in Witherden and Vincent.[4] The listing is the maximum and mean values of the error from hundred tests for each p.

,

	This work			Witherden and Vincent[4]		
p	n	max	mean	n	max	mean
		$R \times 10^{16}$			$R \times 10^{16}$	
1	1	6	1	1	6	1
2	5	33	3	5	11	3
3	6	22	3	6	40	4
4	10	33	5	10	28	4
5	15	39	4	15	100	7
6	23	86	8	24	41	5
7	31	37	4	31	132	6
8	47	76	6	47	121	6
9	62	321	9	62	78	8
10	80	138	10	83	246	9
11	103	663	25	–	–	–
12	127	71	8	–	–	–
13	152	400	19	–	–	–
14	184	257	13	–	–	–
15	234	193	13	–	–	–
16	285	165	12	–	–	–
17	319	403	15	–	–	–
18	357	271	13	–	–	–
19	418	197	11	–	–	–
20	489	111	13	–	–	–

schemes are compared: those proposed in this article, the symmetric rules from Witherden and Vincent[4], and in the case of tetrahedra the non-symmetric rules from Jaśkowiec and Sukumar.[1] We present the relative error in the integration versus the polynomial order p of the cubature scheme. The non-symmetric tetrahedral cubature rules have fewer number of integration points than the symmetric cubature rules, see Table 4. Therefore, for tetrahedra, the relative error in the integration versus the number of integration points in the mesh, n_g, is also shown.

5.2.1 | Exponential test function

We consider the exponential test function[1]

$$f(\boldsymbol{x}) = \exp(\alpha x + \beta y + \gamma z), \tag{26a}$$

whose integral over the biunit cube C is:

$$I = \int_C f(\boldsymbol{x})\,d\boldsymbol{x} = \frac{\big[\exp(\alpha) - \exp(-\alpha)\big]\big[\exp(\beta) - \exp(-\beta)\big]\big[\exp(\gamma) - \exp(-\gamma)\big]}{\alpha\beta\gamma}. \tag{26b}$$

For this test, we choose $\alpha = 5$, $\beta = 2$, and $\gamma = 1$, and the exact value of the integral is $I = 2.5302052232599806 \times 10^2$.

For the tetrahedral mesh, the relative error in integration as a function of p and n_g are presented in Figs. 6a and 6b, respectively. In Fig. 6c, the convergence curve for the relative error in integration versus p is shown for the pyramidal mesh. For both cases, we observe that our schemes and those of Witherden and Vincent[4] converge nonmonotonically. For $p > 10$, the accuracy of our schemes improve and reaches close to $\mathcal{O}(10^{-13})$ for $p = 20$. Our symmetric and non-symmetric cubature schemes display the same level of accuracy, but the non-symmetric rules require fewer number of integration points to reach a prescribed level of accuracy.

5.2.2 | Trigonometric test function

This test is taken from Chen et al,[22] where the trigonometric function

$$f(\boldsymbol{x}) = x^3 \sin(k_y \pi y)\sin(k_z \pi z) \tag{27a}$$

is considered. The integral of f over the unit cube [0, 1] is given by

$$I = \frac{\sin^2\left(\frac{k_y\pi}{2}\right)\sin^2\left(\frac{k_z\pi}{2}\right)}{\pi^2 k_y k_z}. \tag{27b}$$

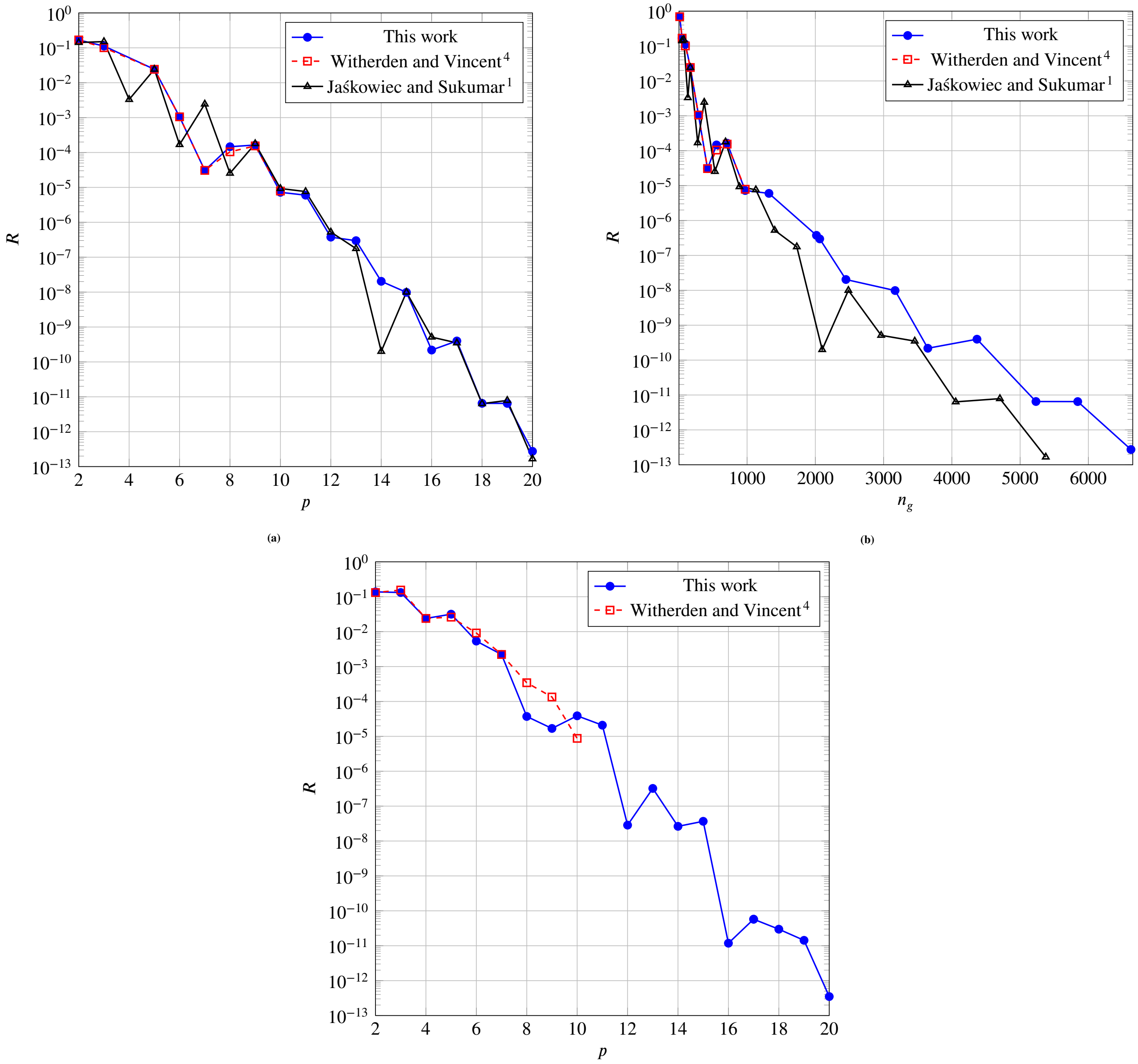
For $k_y = 2.5$ and $k_z = 1$, the value of the integral is $I = 0.2/\pi^2$. Numerical results are presented in Fig. 7, and reveals once again that the accuracy of our scheme and those of Witherden and Vincent[4] closely matches; in addition, all schemes show nonmonotonic convergence. In this test too, our symmetric and non-symmetric cubature rules deliver the same level of accuracy for a given p, with the latter scheme being more efficient.

5.2.3 | Weakly singular test functions over tetrahedra

In this test, we consider the weakly singular functions

$$f_1(\boldsymbol{x}) = \frac{1}{\sqrt{r}}, \qquad f_2(\boldsymbol{x}) = \frac{1}{r}, \qquad r = \sqrt{x^2 + y^2 + z^2}. \tag{28}$$

The exact values of their integrals over the reference tetrahedron are $I = 163/679$ and $I = 1531/4236$, respectively.[1] The relative error in the numerical integration as a function of the order p is presented in Fig. 8. We observe convergence with increase in p, but convergence is nonmonotonic using our cubature rules as well as those from Witherden and Vincent.[4] It is observed that the $p = 9$ scheme of Witherden and Vincent[4] failed in this test, which is due to fact that one point in this cubature



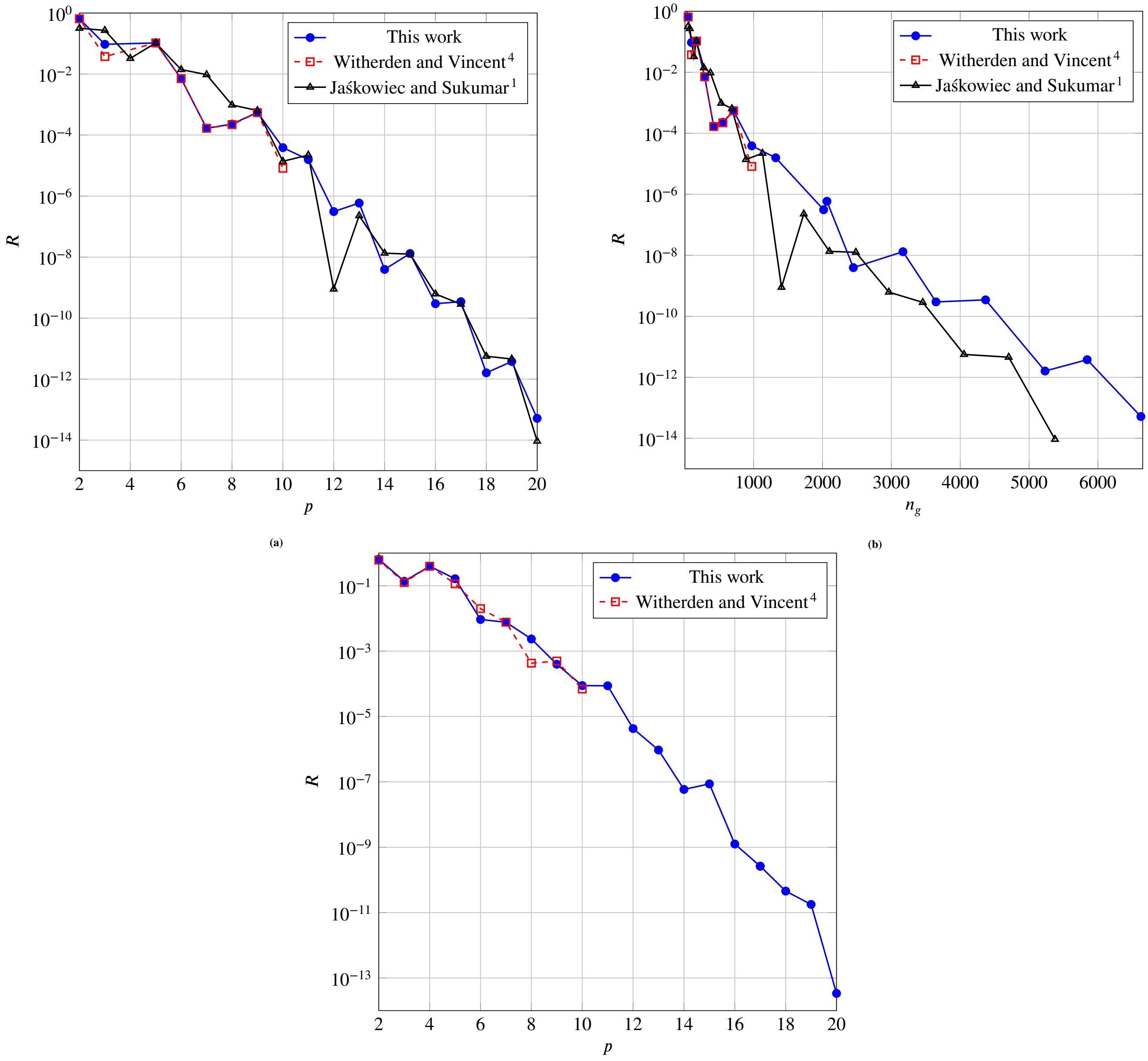
(a)

(b)

(c)

FIGURE 6 Relative error in the integration of the exponential function in (26) ($\alpha = 5$, $\beta = 2$, $\gamma = 1$) over the biunit cube, which is meshed with (a), (b) 12 tetrahedra and (c) 6 pyramids.

rule is very close to the origin where both functions are nearly singular. For our symmetric and non-symmetric cubature rules, the trends from previous tests persist. One noticeable observation is the dramatic reduction in error with the symmetric rule for $p = 20$, which is not attained by the non-symmetric rule of the same order (see Figs. 8a and 8b).



(a)

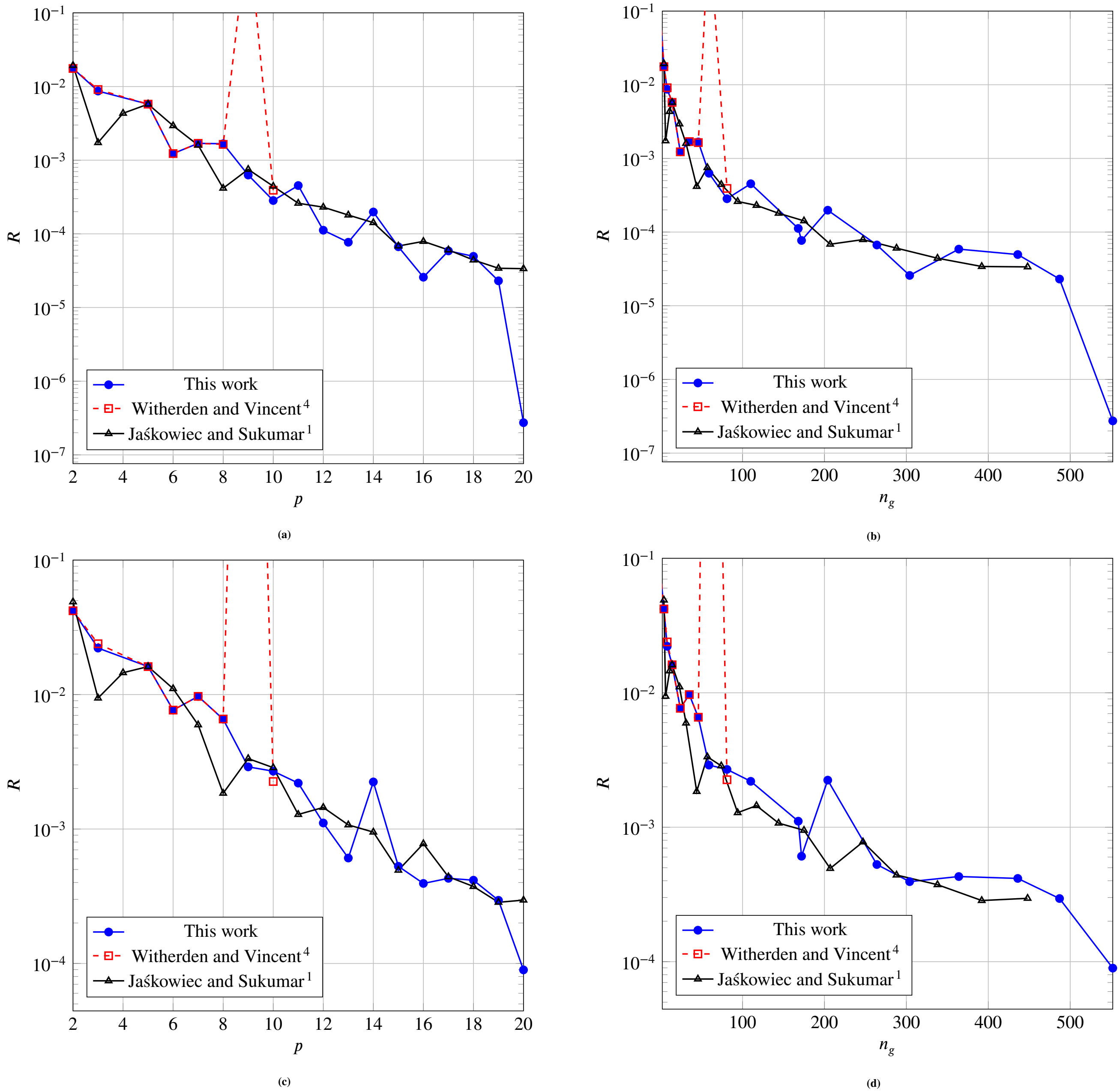
(b)

(c)

FIGURE 7 Relative error in the integration of the trigonometric function in (27) over the unit cube. The unit cube is meshed with (a), (b) 12 tetrahedra and (c) 6 pyramids.

5.2.4 | Integration over a semi-cylindrical domain

In this example, a function $f = \sin(x^2 + y^2)$ over a semi-cylindrical domain Ω of height 2, inner radius $R_1 = \sqrt{\pi/2}$ and outer radius $R_2 = \sqrt{3\pi}$ is considered, where the domain is illustrated in Jaśkowiec and Sukumar.[1] The exact value of the integral of f over Ω is equal to π. The geometric map is provided in Jaśkowiec and Sukumar.[1] Due to the geometric map, the domain Ω is covered by curved tetrahedral or pyramidal elements. The relative error in the numerical integration as a function of the order p is presented in Fig. 9. Up to $p = 10$, our schemes and those from Witherden and Vincent[4] have similar accuracy, and both



(a)

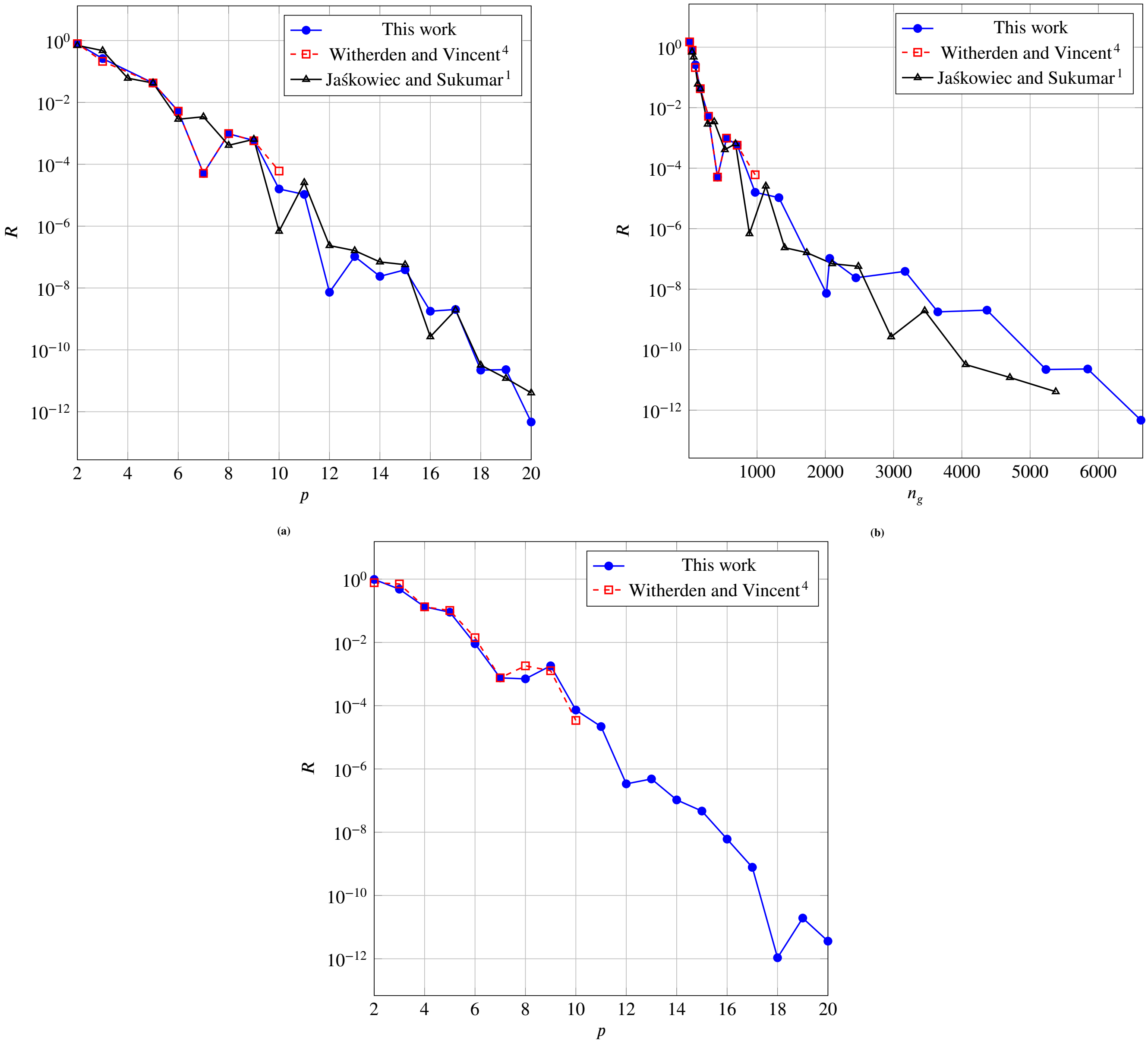
(b)

(c)

(d)

FIGURE 8 Relative error in the integration of weakly singular functions over the reference tetrahedron. (a), (b) $f_1(\boldsymbol{x}) = 1/\sqrt{r}$ and (c), (d) $f_2(\boldsymbol{x}) = 1/r$.

display nonmonotonic convergence. For $p > 10$, the cubature rules proposed in this work continue to show improved accuracy and convergence towards the exact value. This example also affirms the observation from previous tests that the non-symmetric cubature rules are more efficient in integrating nonpolynomial functions.



(a)

(b)

(c)

FIGURE 9 Relative error in the integration of the function $\sin(x^2 + y^2)$ over the semi-cylindrical domain using (a), (b) 12 tetrahedral and (c) 6 pyramidal elements with curved faces.

6 | CONCLUSIONS

A new algorithm for generating symmetric, high-order cubature schemes for tetrahedra and pyramids has been presented in this article. The algorithm consisted of five sequential phases to obtain high-precision cubatures. The algorithm was used to generate cubature schemes of orders from $p = 2$ to $p = 20$ for tetrahedral and pyramidal elements. The cubature rules consist of combinations of symmetry orbits that ensure that the rules are invariant to any affine transformation. The orbits were constructed by permutation stars that consist of free variables, which were determined via the search procedure. Application of symmetry orbits can lead to solving a reduced-scale cubature problem, as has been shown by others.[21,4] However, in this article we considered the full-scale problem, which we found to be suitable to generate cubature rules over the tetrahedron and the pyramid.[†] This increases the computational costs, but the algorithm is robust and high-order cubature rules were found that were efficient. We constructed tetrahedral cubature schemes up to $p = 10$ that have the same number of nodes as those in Witherden and Vincent,[4] but obtained fewer number of nodes for pyramidal cubatures of orders $p = 6$ and $p = 10$. In our previous work,[1] we generated non-symmetric cubature rules for tetrahedra up to $p = 20$. It is evident that the symmetry constraints lead to significant increase in the number of nodes in the cubature scheme. As a matter of comparison, for $p = 10$ and $p = 20$, we obtained $n = 81$ (71) and $n = 552$ (448), respectively, which is an increase of 14 percent and 23 percent in the number of nodes.

We conducted verification tests to ensure the machine-precision accuracy of the cubature rules, and performed convergence tests to demonstrate its performance on several test functions. For $p \leq 10$, our rules were compared to those of Witherden and Vincent.[4] Convergence was observed in all tests, though it was nonmonotonic with some oscillations. This is not unexpected since nonpolynomial functions were being integrated. Besides the test with weakly singular functions, close to machine-precision accuracy was attainable with a $p = 20$ cubature rule for all other tests. Convergence tests were also performed with the non-symmetric tetrahedral cubature rules from our previous work.[1] Consistent with expectations, these non-symmetric rules were more efficient (fewer number of integration points for a given level of accuracy) than the symmetric rules from the present work. All the cubature rules generated in this article are provided in the supplementary materials in extended and compact forms with 128 digits of precision. In the extended form all the nodes with their weights are listed using x, y, z, w values in each row. In the compact form only the values from the symmetry orbits are listed with associated weights. Additionally, the tetrahedral cubature schemes are also provided in barycentric coordinates. The files are given in text format and also in MATLAB® binary format.

ACKNOWLEDGEMENTS

The first author gratefully acknowledges the support provided by the Polish-U.S. Fulbright Commission and the Polish National Agency for Academic Exchange via scholarships at the University of California at Davis. The second author is grateful for the research support of Sandia National Laboratories to the University of California at Davis. The authors also thank Professor Linbo Zhang from Chinese Academy of Sciences, Beijing, for providing us a script to verify the extended precision of the symmetric tetrahedral cubature rules.

[†] We point out that the proposed algorithm has also been applied to the hexahedron and the prism. However, the algorithm was sensitive to the initial guess: converged cubature rules were found for $p = 2$ to $p = 11$ on these elements, with stagnation in the error resulting for $p > 11$. Further research to investigate these issues is needed.

References

1. Jaśkowiec J, Sukumar N. High-order cubature rules for tetrahedra. *International Journal for Numerical Methods in Engineering* 2020; 121(11): 2418–2436.

2. Šolín P, Segeth K, Doležel I. *Higher-Order Finite Element Methods*. Studies in Advanced MathematicsBoca Raton, FL: Chapman & Hall, CRC Press . 2004.

3. Yosibash Z. *p*-FEMs in biomechanics: Bones and arteries. *Computer Methods in Applied Mechanics and Engineering* 2012; 249–252: 169–184.

4. Witherden FD, Vincent PE. On the identification of symmetric quadrature rules for finite element methods. *Computers & Mathematics with Applications* 2015; 69(10): 1232–1241.

5. Papanicolopulos SA. New fully symmetric and rotationally symmetric cubature rules on the triangle using minimal orthonormal bases. *Journal of Computational and Applied Mathematics* 2016; 294: 39–48.

6. Williams DM, Shunn L, Jameson A. Symmetric quadrature rules for simplexes based on sphere close packed lattice arrangements. *Journal of Computational and Applied Mathematics* 2014; 266: 18–38.

7. Shunn L, Ham F. Symmetric quadrature rules for tetrahedra based on a cubic close-packed lattice arrangement. *Journal of Computational and Applied Mathematics* 2012; 236(17): 4348–4364.

8. Duffy MG. Quadrature over a pyramid or cube of integrands with a singularity at a vertex. *SIAM Journal on Numerical Analysis* 1982; 19(6): 1260–1262.

9. Kubatko EJ, Yeager BA, Maggi AL. New computationally efficient quadrature formulas for pyramid elements. *Finite Elements in Analysis and Design* 2013; 65: 63–75.

10. Zienkiewicz OC, Taylor RL. *The Finite Element Method: Solid mechanics*. Fluid DynamicsButterworth-Heinemann . 2000.

11. Bedrosian G. Shape functions and integration formulas for three-dimensional finite element analysis. *International Journal for Numerical Methods in Engineering* 1999; 35: 95–108.

12. Bergot M, Cohen G, Duruflé M. Higher-order finite elements for hybrid meshes using new nodal pyramidal elements. *Journal of Scientific Computing* 2010; 42: 345–381.

13. Owen SJ, Saigal S. Formation of pyramid elements for hexahedra to tetrahedra transitions. *Computer Methods in Applied Mechanics and Engineering* 2001; 190(34): 4505-4518.

14. Graglia RD, Gheorma IL. Higher order interpolatory vector bases on pyramidal elements. *IEEE Transactions on Antennas and Propagation* 1999; 47(5): 775–782.

15. Nigam N, Phillips J. High-order conforming finite elements on pyramids. *IMA Journal of Numerical Analysis* 2012; 32(2): 448–483.

16. Nigam N, Phillips J. Numerical integration for high order pyramidal finite elements. *ESAIM: Mathematical Modelling and Numerical Analysis* 2012; 46(2): 239–263.

17. Bergot M, Duruflé M. Approximation of H(div) with high-order optimal finite elements for pyramids, prisms and hexahedra. *Communications in Computational Physics* 2013; 14(5): 1372–1414.

18. Devloo PRB, Durán O, Gomes SM, Ainsworth M. High-order composite finite element exact sequences based on tetrahedral-hexahedral-prismatic-pyramidal partitions. *Computer Methods in Applied Mechanics and Engineering* 2019; 355: 952–975.

19. Danielson KT, Adley MD. Five node pyramid elements for explicit time integration in nonlinear solid dynamics. *Finite Elements in Analysis and Design* 2018; 141: 37–54.

20. Xiao H, Gimbutas Z. A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Computers & Mathematics with Applications* 2010; 59(2): 663–676.

21. Zhang L, Cui T, Liu H. A set of symmetric quadrature rules on triangles and tetrahedra. *Journal of Computational Mathematics* 2009; 27(1): 89–96.

22. Chen C, Křížek M, Liu L. Numerical integration over pyramids. *Advances in Applied Mathematics and Mechanics* 2013; 5(3): 309–320.

23. Silvester P. Symmetric quadrature formulae for simplexes. *Mathematics of Computation* 1970; 24(109): 95–100.

24. Felippa CA. A compendium of FEM integration formulas for symbolic work. *Engineering Computations* 2004; 21(8): 867–890.

25. Geevers S, Mulder WA, van der Vegt JJW. Efficient quadrature rules for computing the stiffness matrices of mass-lumped tetrahedral elements for linear wave problems. *SIAM Journal on Scientific Computing* 2019; 41(2): A1041–A1065.

26. Skala V. Barycentric coordinates computation in homogeneous coordinates. *Computers & Graphics* 2008; 32(1): 120–127.

27. Meyer CD. *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA: SIAM Publications . 2000.

28. Mousavi SE, Xiao H, Sukumar N. Generalized Gaussian quadrature rules on arbitrary polygons. *International Journal for Numerical Methods in Engineering* 2010; 82(1): 99-113.

29. Schlömer N. Quadpy. Available at https://pypi.org/project/quadpy/; 2020 (Accessed on May 15, 2020).

□