THE UNIVERSITY *of* EDINBURGH

NEW NUMERICAL INTEGRATION APPROACHES FOR FINITE ELEMENT SIMULATIONS

Weizhu Wang

Doctor of Philosophy

THE UNIVERSITY OF EDINBURGH

2024

To my family and friends.

Abstract

This thesis presents a comprehensive study on the development and application of novel numerical integration methods, particularly focusing on Gaussian-type cubature rules and their implications in Finite Element Method (FEM). The research is structured into three pivotal segments, each targeting different aspects of numerical integration to enhance the precision and efficiency of cubature rules within computational geometries.

The first segment addresses the derivation of explicit consistency conditions for constructing optimal fully symmetric cubature rules for tetrahedra. Utilizing a novel non-monomial fully symmetric polynomial basis, this work successfully defines the consistency conditions necessary for determining the most efficient rule structures, thereby minimizing the number of integration points required without compromising the accuracy.

In the second segment, the focus shifts to exploring rotational symmetry and multisymmetric polynomials in the moment equations for cubature rules. This includes the development of a new rotationally symmetric monomial basis, which simplifies the complicated system of moment equations. The resultant novel cubature rules, particularly for tetrahedra, demonstrate fewer integration points compared to existing rules, thus enhancing computational efficiency.

The final segment investigates the formulation of FEM and the patch test for new elements. It introduces a groundbreaking framework that leverages the established theory of cubature formulas to devise rules that not only pass the patch test but do so with fewer integration points. This framework is pivotal for advancing the blending of numerical methods into practical engineering applications, ensuring both reliability and efficiency.

Overall, the thesis encapsulates significant advancements in the field of numerical integration, presenting new methodologies and algorithms that refine the creation of cubature rules. These innovations provide substantial contributions to the domains of computational mathematics and engineering, particularly in the optimization of FEM. The results published within this thesis highlight the potential of these new approaches to set a foundation for future research in numerical integration methods.

Lay Summary

This thesis explores advanced numerical integration methods and their application in Finite Element Method (FEM), focusing on the development and validation of cubature rules-mathematical formulas for evaluating integrals numerically. These rules are vital in engineering and physics for calculating the properties of elements under various forces, a process central to FEM.

The research is divided into three main parts, each addressing different aspects of cubature rule development. The first part develops new consistency conditions for creating efficient and accurate cubature rules for tetrahedra, a common geometric element in three-dimensional modeling. The results, which are part of published research, streamline the process of identifying the best rules that minimize computation without losing accuracy.

The second part of the research investigates other types of symmetry, such as rotational symmetry, and examines multisymmetric polynomials in the moment equations of cubature rules. These studies refine methods for calculating cubature rules, allowing for greater flexibility in adapting to various geometric configurations and enhancing their applicability.

The third segment of the study addresses the practical application of cubature rules in FEM, particularly their performance in real-world engineering problems. It introduces a novel framework that simplifies the development of new cubature rules and ensures they meet industry standards by passing the patch test, which checks the convergence of new elements in FE simulations.

This thesis outlines each step of cubature rule construction, from theoretical foundations in polynomial mathematics to practical tests confirming their reliability in engineering applications. By combining innovative mathematical approaches with practical testing, the research enhances our ability to perform precise numerical integrations in complex engineering analyses, resulting in more accurate and reliable FEM models.

This work is crucial for engineers and researchers involved in computational simulations where precision and efficiency are paramount. It reduces computational costs and improves the reliability of simulations used to predict material behavior under various conditions, which is essential for designing safer and more efficient structures and components.

Acknowledgements

I would like to extend my deepest gratitude to my supervisor, Dr. Stefanos-Aldo Papanicolopulos, for his invaluable guidance, patience, and support throughout this thesis. His insights and expertise have been crucial in shaping this research and my growth as a scholar. Stefanos consistently provided encouragement and constructive feedback, essential in overcoming the challenges I faced during my studies.

I am particularly grateful for his flexibility and understanding, allowing me to explore my ideas while providing the necessary resources to refine them. The opportunities for collaboration and professional advice have been immensely beneficial.

Thank you, Stefanos, for your mentorship, perseverance, and belief in my potential. Your dedication to my project and faith in my abilities have been a great motivational force.

在我落笔的这一刻，从我踏入小学算起，已经是整整二十四年了。二十四年的求学路，充满了梦想和回忆，充满了一路坎坷与幸运。从四川的一个贫穷的小镇，到县城，到北京，到深圳，到广州，到德国，最后到了英国，这求学的小半辈子也算是走南闯北，行了万里路，而今天我又将在万卷书上画上一个重要的句号。

能够坚持到今天，我想要感谢很多，但是最多的就是我的父母，没有他们也就没有今天的我，虽然在求学的道路上大多数时候是我自己独自前行，但是他们以身作则教会了我很多重要的品格，比如善良勇敢，比如刻苦坚持，但是我认为最重要的是让我感受到了尊重以及由此带来的自信，这是支持我走到今天这一步的重要原因。

小时候家里很穷，记得母亲说过那时候基本上都是每顿饭都只吃青菜，一周只有我能够吃一次水煮蛋补充营养。原因一个是他们的工资收入少，再有就是父亲喜欢帮助那些乡下没钱读书的穷学生，垫付学费和书费。我记得两岁的时候因为拿了别人一毛钱去买冰棍而被父亲狠狠的打了一顿，他后来告诉我那一天他边打边哭，我也是从那一次明白了什么是贫穷。

我是幸运的，在父母的努力下，我们的家庭条件后来好了一些，不用跟别人在餐桌上抢肉吃了。但是我也慢慢体会到我和别的小孩不一样，不管是在县城还是在大城市，别人总是有我没有的玩具或文具，我十分羡慕，也很想要，可是我明白家里的情况，所以很少开口要买，因为我记得母亲说过最多的一句话就是人穷志不穷。上初中以后我喜欢上了看书，每个周末我都去书城，一坐就是一个下午，我什么书都喜欢看，传记，小说，科普，甚至是看不懂的政治，经济和哲学，有时候会看到天黑忘了回家。我还喜欢跟同学借书看，我发现他们的家里有好多有趣的书，比如儿童文学和武侠小说，还有各种模型，虽然我买不起，但是我人缘不错，他们都愿意借给我看，借给我玩。我至今仍然记得一本书叫做蓝熊船长的十三条半命，里面的故事充满了想象力让我现在都还有印象。

说来也惭愧，我的成绩虽然不算差，但是也不算拔尖，只是偶尔能够有一两次出彩的表现。感谢父母在我考好的时候提醒我，在我考差的时候鼓励我，让我一直能够顺利的升学，在还不错的学校学习。后来因为国家的经济发展，也因为父母勤奋工作，家里条件逐渐好了一些，在我大学毕业之后让我能够有机会出国去留学，看一看外面的世界，虽然我的父母自己都不知道外面的世界是什么样子。我至今都记得我第一次出国去德国上学的时候没有经验，本来八个多小时的旅程辗转了二十多个小时，最后才在法兰克福的机场降落。说实话，当我出机舱的那一刻，我的内心惶恐大过兴奋，在这个离家上万公里的完全陌生的环境，完全陌生的文字，陌生的文化，没有任何一个认识的人，那是一种扑面而来令人窒息的孤独和迷茫。还好，我是一个承受能力还不错的人，克服了一个又一个困难，幸运地找到了自己的另一半，也终于三年后在最难毕业的德国学校毕业了。

也许是因为惯性，也许是对知识最高处的风景的好奇，我选择了继续读博，这一次我来到了英国。果不其然，我的求学总是从磨难开始，一来英国就遇上了疫情，整整两年多一直被困在一个大概十平米的小房间，一边克服着内心的孤独和抑郁，一边在学术上蹒跚前行，在煎熬中学会了面对真实的自己。后来解封后我的学习和生活终于回到正轨，也交到了许多好朋友（彦凯，浩祺，香伊，小刘，万州，事阳等），我们一起吃火锅，聊天，徒步，打球，一切终于顺利了一次，直到今天我完成学业的时刻，想想也是不容易。

这就是我前半生求学的路，简单幸运而又充满了坎坷和辛酸。我爸总是开玩笑，说我很会考试，不管是中考还是高考还是出国，我都能压着分数线达到录取条件，我自己也笑了，我确实是幸运的，幸运的是父母教会了我那些宝贵的品质，幸运的是我面对困难的时候没有放弃过。

在一部电影《魔鬼代言人》中有一句话我记忆犹新：人在面对压力的时候会有两种表现，一个是崩溃，一个是更加坚强。我想我幸运的是后者吧。我今年三十岁，人生到这里才走了一小半，希望我剩下的日子里能够对得起我这二十多年的艰辛和执着，对得起自己和父母的付出，对得起老天爷给的幸运。愿前路天地广阔，一切安好。

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Weizhu Wang

Contents

Figures and Tables

Figures

Tables

Chapter 1

Introduction

1.1 Background and motivation

Numerical integration, known as quadrature in one dimension and cubature in higher dimensions, has a history predating calculus. This field has engaged many eminent researchers. Ancient Greek philosophers like Eudoxus and Archimedes developed methods for calculating areas and volumes. Throughout history, scientists worldwide have contributed significantly to numerical integration. For instance, as shown in Table 1.1, Liu Hui from ancient China used the circle-cutting technique to estimate the area of a circle. Inspired by a salesman's method of measuring wine barrel volume with a gauging rod, Kepler authored "Nova Stereometria Doliorum Vinariorum." At the end of his book, Kepler reflects on the balance between approximations and precise calculations, stating, "Et cum pocula mille mensi erimus, Conturbabimus ilia, ne sciamus" (translated as "When we have measured a thousand barrels, we will mix them up, so that we do not know").

Numerical integration and calculus are closely intertwined, with numerical integration as a practical application of calculus principles. The development of numerical integration evolved alongside calculus. For calculus, early contributions include Fermat's methods for finding maxima and minima and Wallis's introduction of limits in "Arithmetica Infinitorum," which advanced integration for any real exponent. The 17th century was pivotal with Newton and Leibniz independently formulating integration principles, laying the foundations of modern calculus. Their work revolutionized mathematics, profoundly impacting science, engineering, and economics, and cementing their status as central figures in mathematical history.

The advent of calculus enabled the addressing of intricate scenarios, such as solving boundary value problems using the Finite Element Method (FEM) developed in the 1900s, where the integrand may be non-polynomial and complex. This challenge led to the development of approximation methods for definite integrals, termed numerical integration by Gibb in 1915. Techniques like Simpson's method for one-dimensional cases were developed, followed by methods for multi-dimensional integrals. Notably, Ulam, Metropolis, and von Neumann pion-

eered the Markov Chain Monte Carlo method in 1945 at Los Alamos National Laboratory, foundational to Monte Carlo integration. Building on their work, Korobov introduced Lattice rules, a form of quasi-Monte Carlo rules, for numerical integration over the d-dimensional unit cube.

Gauss's publication on numerical integration introduced the Gaussian quadrature method, significantly enhancing prior methodologies and facilitating their application to higher-dimensional problems. This method has inspired subsequent mathematical research. In 1877, Maxwell presented the first cubature formula in its modern form. This thesis emphasizes Gaussian-type cubature, which estimates integrals as weighted sums of evaluations at specific points, for its computational efficiency. The goal is to approximate the integral accurately while minimizing function evaluations. This approach has gained widespread adoption due to its effectiveness in numerical computations. Despite the extensive collection of available cubature rules (Cools, 1999, 2003; Cools & Rabinowitz, 1993), ongoing research continues to uncover new results (Jaśkowiec & Sukumar, 2020b; Wang & Papanicolopulos, 2023; Williams, Shunn, & Jameson, 2014; Witherden & Vincent, 2015; Xiao & Gimbutas, 2010a), and theoretical work suggests many improvements on existing rules are yet to be discovered (Maeztu & Sainz de la Maza, 1995).

1.1.1 Numerical integration

In the 18th and 19th centuries, the challenge of solving integrals of basic functions analytically, compared to the ease of deriving their derivatives, highlighted the significance of numerical integration. This distinction emphasized its importance in numerical analysis. Practical reasons for favoring numerical integration include:

- The function f to be integrated may only be known at discrete points.
- Finding an elementary antiderivative may be difficult or impossible.
- Numerical approximation is often simpler than symbolic computation of the antiderivative.

The basic problem in numerical integration is to approximate a definite integral

$$\int_{\Omega} f(\boldsymbol{x}) d\Omega \tag{1.1}$$

where f is integrand and Ω is the domain of integral. Numerical integration uses numerical techniques to approximate integrals. Traditionally, "quadrature" refers to calculating the area under a curve defined by one variable, though it is also used for multivariable cases by some researchers. The Oxford English Dictionary defines quadrature as "the process of constructing geometrically a square equal in area to that of a given figure, especially a circle," highlighting its geometric origins and broader application. In higher dimensions, this concept is known as "cubature."

Table 1.1: Development of numerical integration prior to calculus.

Name	Time	Achievement
Eudoxus	4th Century BC	Introduced abstract proofs for finding areas and volumes.
Archimedes	3rd Century BC	Explored volumes and areas from algebraic curves; developed the method of exhaustion.
Babylonians	Before 50 BC	Employed the trapezoid rule for astronomical calculations.
Hui Liu	AD 263	Developed the circle-cutting technique to calculate areas and volumes.
L. Valerio	16th Century	Formalized the method for approximating areas under curves with sums of rectangles.
J. Kepler	1615	Published "Nova Stereometria Doliorum Vinariorum," exploring volume measurement methods.
P. Fermat	c. 1636	Derived a method for finding maxima or minima of expressions.
R. Descartes	1637	Established analytic geometry in "Discourse on the Method."
B. Cavalieri	1639	Introduced infinitesimals in geometric area calculations.
J. Wallis	1656	Advanced the concept of limits in "Arithmetica Infinitorum" and expanded integration theory.
I. Newton and R. Cotes	1676	The Newton-Cotes integration formula was introduced, enhancing numerical integration methods.
I. Newton and G. W. Leibniz	Late 17th Century	Formulated the foundations of infinitesimal calculus.

Table 1.2: Development of numerical integration after calculus

Name	Time	Achievement
I. Newton and G. W. Leibniz	17th Century	Formalized the foundations of infinitesimal calculus.
T. Simpson	1743	Introduced Simpson's rule for estimating definite integrals.
J. C. F. Gauss	1815	Described the Gaussian quadrature method for numerical integration, significantly enhancing existing techniques.
J.C. Maxwell	1877	Introduced the first modern cubature formula.
E.T. Whittaker	1912	Published "The Calculus of Observations."
D. Gibb	1915	First used the term "numerical integration" in "A Course in Interpolation and Numeric Integration for the Mathematical Laboratory."
S. Ulam, N. Metropolis, and J. von Neumann	1945	Developed the Markov Chain Monte Carlo method at Los Alamos National Laboratory for nuclear research.
J. Neumann and H. Goldstine	1947	Discussed numerical matrix inversion in "Numerical inverting of matrices of high order."
S.L. Sobolev	1950s	Advanced the creation of invariant cubature formulas.
N.M. Korobov	1959	Introduced Lattice rules.

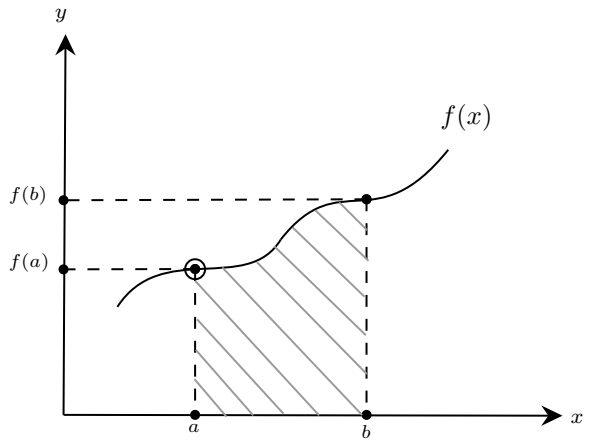


Figure 1.1: Integration of function $f(x)$ from a to b

The advent of automatic computing established quadrature as a key numerical task, significantly expanding the field of numerical integration of differential equations. This development broadened the scope of computational problems that could be addressed.

Quadrature is the simplest case of numerical integration, evaluating the integral

$$I = \int_b^a f(x)dx \tag{1.2}$$

which essentially calculates the area under a curve, as shown in Figure 1.1.

Numerous numerical integration methods have been developed, as shown in Figure 1.2. These methods are categorized into improper and proper integrals. Improper integrals deal with singularities in the integrand, for which the Tanh-Sinh method (Bailey, 2006) is a reliable alternative when Gaussian quadrature is less effective.

For proper integrals with smooth integrands, methods vary by the number of dimensions. In one dimension, classical integration formulas like the Newton-Cotes family, including trapezoidal and Simpson's rules, are notable for their historical significance. However, the trapezoidal and midpoint rules are the most practical. In higher dimensions, Romberg's method, using Richardson extrapolation to extend the trapezoidal rule, improves applicability.

Gaussian quadrature enhances traditional methods by allowing flexible selection of weighting coefficients and evaluation points, which need not be equally spaced. This increases degrees of freedom compared to Newton-Cotes formulas, potentially doubling the accuracy with the same number of evaluations. However, high-order integration does not ensure high accuracy unless the integrand closely resembles a polynomial, emphasizing the importance of the integrand's smoothness for optimal results.

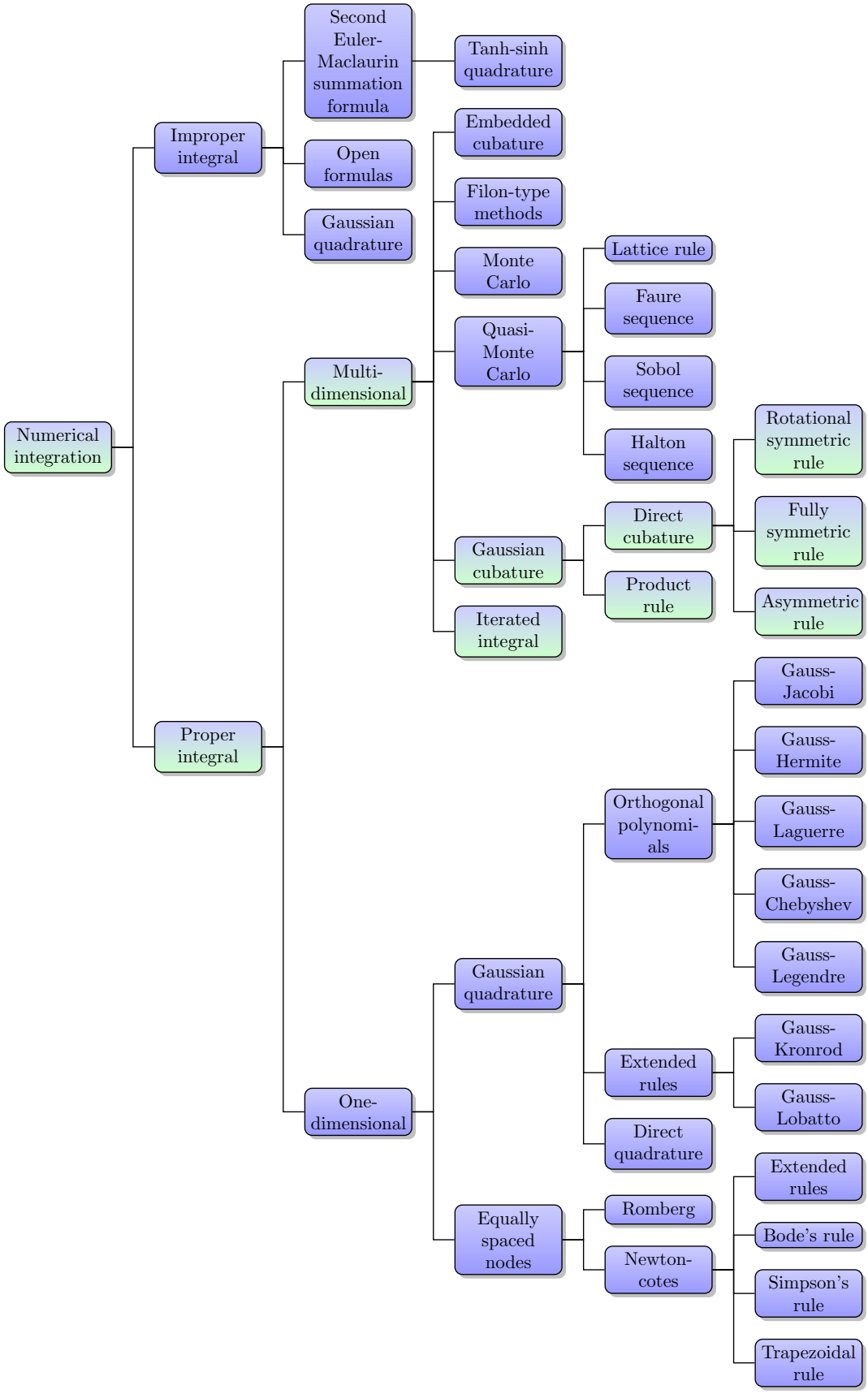


Figure 1.2: Classification of different numerical integration methods (there might be others). The methods marked in green are studied in this thesis.

Table 1.3: Weight functions, intervals, and recurrence relations for part of classic orthogonal polynomials.

Name	Weight function	Range	Recurrence relations
Gauss-Legendre	$w(x) = 1$	$-1 < x < 1$	$(j+1)P_{j+1} = (2j+1)xP_j - jP_{j-1}$
Gauss-Chebyshev	$w(x) = (1-x^2)^{-1/2}$	$-1 < x < 1$	$T_{j+1} = 2xT_j - T_{j-1}$
Gauss-Laugerre	$w(x) = x^{\alpha}e^{-x}$	$0 < x < \infty$	$(j+1)L^{\alpha}_{j+1} = (-x+2j+\alpha+1)L^{\alpha}_j - (J+\alpha)L^{\alpha}_{j-1}$
Gauss-Hermite	$w(x) = e^{-x^2}$	$-\infty < x < \infty$	$H_{j+1} = 2xH_j - 2jH_{j-1}$
Gauss-Jacobi	$w(x) = (1-x)^{\alpha}(1+x)^{\beta}$	$-1 < x < 1$	$c_j P^{(\alpha,\beta)}_{j+1} = (d_j + e_j x)P^{(\alpha,\beta)}_j - f_j P^{(\alpha,\beta)}_{j-1}$

With coefficients are defined as:
$c_j = 2(j+1)(j+\alpha+\beta+1)(2j+\alpha+\beta)$
$d_j = (2j+\alpha+\beta+1)(\alpha^2-\beta^2)$
$e_j = (2j+\alpha+\beta)(2j+\alpha+\beta+1)(2j+\alpha+\beta+2)$
$f_j = 2(j+\alpha)(j+\beta)(2j+\alpha+\beta+2)$

Literature distinguishes two main approaches to constructing Gaussian quadrature formulas: the indirect method and the direct method. The indirect method leverages existing quadrature rules, while the direct method develops new rules by either solving nonlinear equations or identifying polynomials that vanish at specific points using orthogonal polynomials and a corresponding weight function. This weight function can be chosen to address integrable singularities. Common orthogonal polynomials, their weight functions, intervals, and recurrence relationships are detailed in Table 1.3. When initial guesses for polynomial roots are uncertain, the Golub-Welsch algorithm (Golub & Welsch, 1969) is more effective than traditional methods like Newton's method.

Integrating functions over multi-dimensional regions is challenging for two main reasons. First, the number of function evaluations grows exponentially with the dimension n. For instance, a one-dimensional integral requiring 30 evaluations might need 27,000 evaluations for a three-dimensional cube using the product rule. Second, the integration domain in n-dimensional space is bounded by a potentially complex $(n-1)$-dimensional surface, which can be non-convex or non-simply connected, unlike the simple boundaries in one-dimensional integrals.

Various methods have been developed to approximate multi-dimensional integrals. Iterated integrals are the simplest approach and are especially effective when the function and domain exhibit symmetry, allowing for dimensional reduction. For example, in three dimensions, integrating a spherically symmetric function over a spherical domain can be reduced to a one-dimensional integral in polar coordinates.

Monte Carlo methods are preferred for complex boundaries, high-dimensional integration (thousands of dimensions), non-peaked integrands, and when lower accuracy is acceptable. They are well-suited for oscillatory or discontinuous integrands across many dimensions. For lower dimensions (up to hundreds), Quasi-Monte Carlo methods, using low-discrepancy sequences, offer faster convergence. Filon-type methods (Milovanović & Stanić, 2014) are suitable for highly oscillatory integrands.

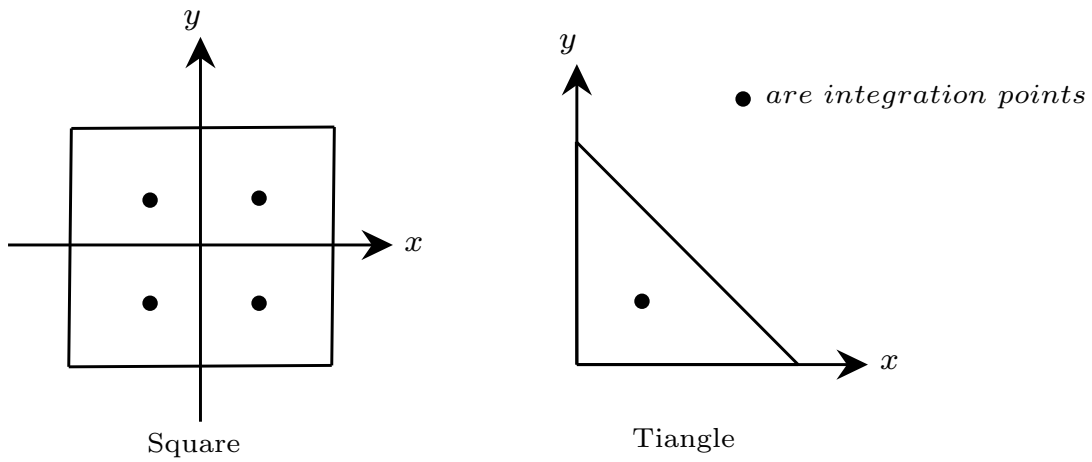


Figure 1.3: Gaussian type cubature rules for square and Triangle

Gaussian cubature is used for problems of moderate complexity. It becomes a product rule when generated from one-dimensional integrals across each dimension. This method is most effective for simple boundaries and smooth functions. For higher accuracy, Gaussian cubature and related methods outperform Monte Carlo methods, which converge more slowly.

Gauss-Kronrod quadrature enhances accuracy by combining the lower-degree Gaussian rule for primary estimates with the higher-degree Kronrod rule for error estimation or refined results. Similarly, embedded cubature (Cools & Haegemans, 1994; Genz & Malik, 1983) uses multiple quadrature rules of varying precision to estimate integrals over specified domains.

For strongly peaked integrands within well-defined regions, a compound rule can divide the integral into smoother segments, allowing for more accurate and efficient processing.

This research focuses on Gaussian-type cubature rules, found by directly solving moment equations, known for their accuracy and ease of implementation despite the challenge of determining the abscissas and weights (Cools, 1997). To simplify finding these rules, the study explores the invariance in cubature rules. Gaussian-type cubature rules are widely used in many FEM software packages. Figure 1.3 illustrates two examples.

Developing cubature rules involves the challenge of solving strongly nonlinear moment equations. Research focuses on creating efficient methods to solve these equations, tailoring cubature rules for specific needs like integration domain or accuracy level.

Notable efforts have targeted triangular (Papanicolopulos, 2015, 2016a, 2016b; Witherden & Vincent, 2015; Xiao & Gimbutas, 2010a; L. Zhang, Cui, & Liu, 2009) and tetrahedral domains (Chuluunbaatar, Chuluunbaatar, Gusev, & Vinitsky, 2022; Jaśkowiec & Sukumar, 2020a, 2020b; Shunn & Ham, 2012; Witherden & Vincent, 2015; L. Zhang et al., 2009), leveraging their symmetries. Preserving these symmetries optimizes computational effort and improves precision, making these rules essential for engineering and scientific applications involving symmetric domains (Begehr, Demidenko, & Matveeva, 2021; Sobolev, 1962).

Symmetry in cubature rules, ensuring invariance to transformations like vertex permutations, is crucial for practical applications. This invariance guarantees that numerical integration results are unaffected by vertex order and simplifies solving moment equations. Consequently, most established rules for triangles and tetrahedrons exhibit full symmetry.

Symmetric polynomials are crucial in developing Sobolev's invariant Gaussian-type cubature methods, a topic of significant historical and contemporary interest in mathematics. The study of symmetric polynomials began in the 18th century with Isaac Newton's work on the symmetric properties of equation roots. Early contributions by German researchers Schläfli (Schläfli, 1851; Schläfli et al., 1950) and Junker (1893) laid the foundation for this field.

Interest is growing in using symmetric polynomials and invariant theory to enhance cubature formulas, particularly for efficiently solving stochastic partial differential equations (Heitzinger, Pammer, & Rigger, 2018). These applications highlight the enduring importance and versatility of symmetric polynomials in contemporary mathematical challenges.

The value of variables in symmetric polynomials are the roots of a polynomial equation using elementary symmetric polynomials. The process of determining these roots can be significantly streamlined by examining the associated Galois group.

Galois groups are a central concept in the field of Galois theory, which explores the relationship between field extensions and polynomial roots. The theory provides a profound connection between the structure of a polynomial's roots and group theory, specifically through automorphisms of field extensions.

For a given polynomial, the Galois group represents the symmetries of its roots. These symmetries arise from the automorphisms of the field generated by the roots, which map the roots to one another while preserving the structure of the field. If the polynomial is irreducible over a base field, its Galois group captures how the roots of the polynomial are permuted.

For example, the Galois group of a quadratic polynomial consists of two elements, corresponding to swapping the two roots. Cubic and quartic polynomials have more complex Galois groups, and their roots can still be expressed using radicals. The Galois group of the general cubic polynomial is isomorphic to the symmetric group $\mathbb{S}_3$, which captures the permutations of its three roots. In this thesis, we will study the Galois group of a quartic polynomial, which is isomorphic to the symmetric group $\mathbb{S}_4$.

Symmetry in cubature rules defines orbits, sets of integration points unchanged by specific transformations. Each symmetry introduces various orbits, forming a "rule structure" specifying orbit counts. "Consistency conditions" (Mantel & Rabinowitz, 1977; Rabinowitz & Richter, 1969) help identify optimal rule structures by predicting solvable moment equations and estimating the most efficient point count.

Consistency conditions for fully symmetric rules on triangles are well-established and widely used (Lyness & Jespersen, 1975). Maeztu and Sainz de la Maza (Maeztu & Sainz de la Maza, 1995) presented these conditions and optimal rule structures for tetrahedrons. However, their limited detailed implementation has constrained their application in recent tetrahedron research (Jaśkowiec & Sukumar, 2020b; Shunn & Ham, 2012; Witherden & Vincent, 2015; L. Zhang et al., 2009), with (Chuluunbaatar et al., 2022) being a notable exception.

While fully symmetric cubature rules are common, they may not always be the most efficient in terms of integration points for a given accuracy. Exploring other symmetries, such as rotational symmetry and multisymmetry, could yield more efficient rules with fewer points. Research on rotational symmetry in cubature is limited (Gatermann, 1988; Papanicolopulos, 2016b; Xiao & Gimbutas, 2010b). Similarly, few studies address cubature for multisymmetric functions (Heitzinger et al., 2018; Rigger, 2017), and none focus on multisymmetric polynomials in cubature formulation. These gaps present intriguing research opportunities, explored in Chapter 3.

1.1.2 Cubature rules for FEM

The Finite Element Method (FEM) is widely used in academia and industry. Its accuracy depends on factors like the numerical integration method. The importance of numerical integration in FEM, particularly for problems with variable properties or coefficients in curvilinear coordinate systems, has been extensively studied for decades (Ciarlet & Raviart, 1972; Fix, 1972; Fried, 1973; Hellen, 1972; B. Irons, 1966; B. M. Irons, 1969, 1971; Strang, 1972; O. C. Zienkiewicz, 1971).

A crucial component of FEM is the stiffness matrix, integral to finite element formulation. Constructing this matrix involves integrating a potentially non-polynomial integrand if the Jacobian matrix is not constant. The standard FEM process iterates over discretized elements, creating an element stiffness matrix for each, which is then aggregated into the global stiffness matrix. This matrix is essential for solving the system of linear equations in FEM simulations. An example of an element stiffness matrix K_e is shown in Figure 1.4, where f_{ij} are integrand functions and Ω_e is the element domain.

$$K_e = \begin{bmatrix} \int_{\Omega_e} f_{11} d\Omega_e & \int_{\Omega_e} f_{12} d\Omega_e & \dots & \int_{\Omega_e} f_{1(n-1)} d\Omega_e & \int_{\Omega_e} f_{1n} d\Omega_e \\ \int_{\Omega_e} f_{21} d\Omega_e & \int_{\Omega_e} f_{22} d\Omega_e & \dots & \int_{\Omega_e} f_{2(n-1)} d\Omega_e & \int_{\Omega_e} f_{2n} d\Omega_e \\ \vdots & & \ddots & & \vdots \\ \int_{\Omega_e} f_{(n-1)1} d\Omega_e & \int_{\Omega_e} f_{(n-1)2} d\Omega_e & \dots & \int_{\Omega_e} f_{(n-1)(n-1)} d\Omega_e & \int_{\Omega_e} f_{(n-1)n} d\Omega_e \\ \int_{\Omega_e} f_{n1} d\Omega_e & \int_{\Omega_e} f_{n2} d\Omega_e & \dots & \int_{\Omega_e} f_{n(n-1)} d\Omega_e & \int_{\Omega_e} f_{nn} d\Omega_e \end{bmatrix}$$

Figure 1.4: Example of element stifffness matrix

The complexity of creating the element stiffness matrix depends on the problem and approximation used. Each entry is theoretically obtained by evaluating integrals from the weak form of the finite element problem. For simple elements and approximations, these integrals can be precomputed analytically, reducing the creation of stiffness matrix entries to algebraic operations. Here, element parameters like vertex coordinates are substituted into closed-form formulas. Consequently, assembling the global stiffness matrix mainly involves these quickly calculated element stiffness matrices (Fried, 1973).

In nonlinear problems (where coefficients depend on the solution) or with intricate curvilinear elements, integrals cannot be precomputed, necessitating numerical integration. This increases computational resources, especially for higher-order approximations. Thus, numerical integration methods are crucial in FEM, and selecting the appropriate method is vital to avoid significant issues.

Recent research on Gaussian-type cubature rules includes significant contributions from (Chuluunbaatar et al., 2022; Jaśkowiec & Sukumar, 2020a, 2020b; Papanicolopulos, 2016a, 2016b), who use iterative algorithms to solve moment equations numerically for new cubature rules. In contrast, Papanicolopulos (2015) addresses the moment equations for algebraic solutions. Gaussian-type rules are also crucial in the XFEM model, as discussed by Kästner et al. (2012).

Higher-order cubature rules which is the focus of this thesis play an especially important role in higher order problems in numerical techniques. Methods like high-order FEM, spectral element methods (Pozrikidis, 2005) offer significant advantages in terms of accuracy and efficiency, especially for solving complex PDEs over complicated geometries. One critical aspect of these methods is the accurate evaluation of integrals over elements (Banerjee & Suri, 1992), which is central to both the assembly of system matrices and the computation of physical quantities such as fluxes and stresses.

For hp-adaptive methods, solution is approximated using polynomials of varying degrees (denoted as "p") on elements of varying sizes (denoted as "h"). It enables modelling highly irregular solutions, inaccessible with standard finite element or finite difference approximations. For example, Demkowicz (2007) gives an detailed introduction to FE codes for Maxwell equations that support hp refinements on irregular meshes. Rachowicz and Demkowicz (2000, 2002) describe an implementation of the hp-adaptive, mixed FE method for the solution of steady-

state Maxwell's equations and infinite element for three-dimensional, time harmonic Maxwell's equations(Cecot, Rachowicz, & Demkowicz, 2003). Ainsworth and Parker (2021a, 2021b, 2022) develop a mass conserving, high order, mixed finite element method for Stokes flow using hp approximation. Approximation with high-order polynomial is common in hp-adaptive methods (Binev, 2018; Düster, Demkowicz, & Rank, 2006; Zander, Bog, Kollmannsberger, Schillinger, & Rank, 2015). Higher-order cubature rules on triangle and tetrahedral elements are employed in the p-version of the finite element method, as shown by Solin, Segeth, and Dolezel (2003), Yosibash (2012), and Schneider et al. (2018).

The accuracy of the method depends not only on the polynomial degree of the solution space but also on the precision with which integrals over finite elements are computed. Inadequate integration rules can lead to numerical instability, such as ill-conditioning of system matrices or loss of convergence, even when the polynomial degree is increased.

Higher-order cubature rules allow for the accurate evaluation of integrals when higher-order polynomials are used to approximate the solution. For example, if a finite element method uses a polynomial of degree p, the corresponding integration rule must have sufficient accuracy to exactly integrate the polynomial basis functions, which often requires rules that can integrate polynomials of degree 2p or higher. Without such precision, errors in the numerical integration could dominate the solution errors, making the method inefficient or even incorrect.

Higher-order cubature rules, while involving more integration points per element, often reduce the overall computational cost due to the exponential convergence of high-order methods. For a given accuracy, fewer degrees of freedom are required when using high-order elements, and hence fewer elements may be needed. The more accurate integration provided by higher-order cubature rules ensures that the method achieves its full potential in terms of accuracy per degree of freedom.

Moreover, the use of high-order cubature rules can significantly reduce the dispersion and dissipation errors in spectral element methods and high-order discontinuous Galerkin (DG) methods. This has been demonstrated in several finite element libraries such as MFEM(Anderson et al., 2021), MoFEM(Kaczmarczyk et al., 2020), getFEM++(Renard & Poulios, 2020), deal.II(Bangerth et al., 2016), FEniCS(Logg, Mardal, & Wells, 2012), FreeFEM++(Hecht, 2012), where efficient implementations of high-order cubature rules are key to achieving high performance in hp-adaptive frameworks.

Due to the limited precision with which the weights and coordinates of integration points can be stored in computer memory, truncation errors accumulate as the number of integration points increases, potentially leading to an unacceptable total error in the final result. Furthermore, FEM involves iterating over all integration points during computation. Therefore, reducing the number of integration points directly decreases the computational time required. This advantage becomes even more pronounced in dynamic problems or time-dependent simu-

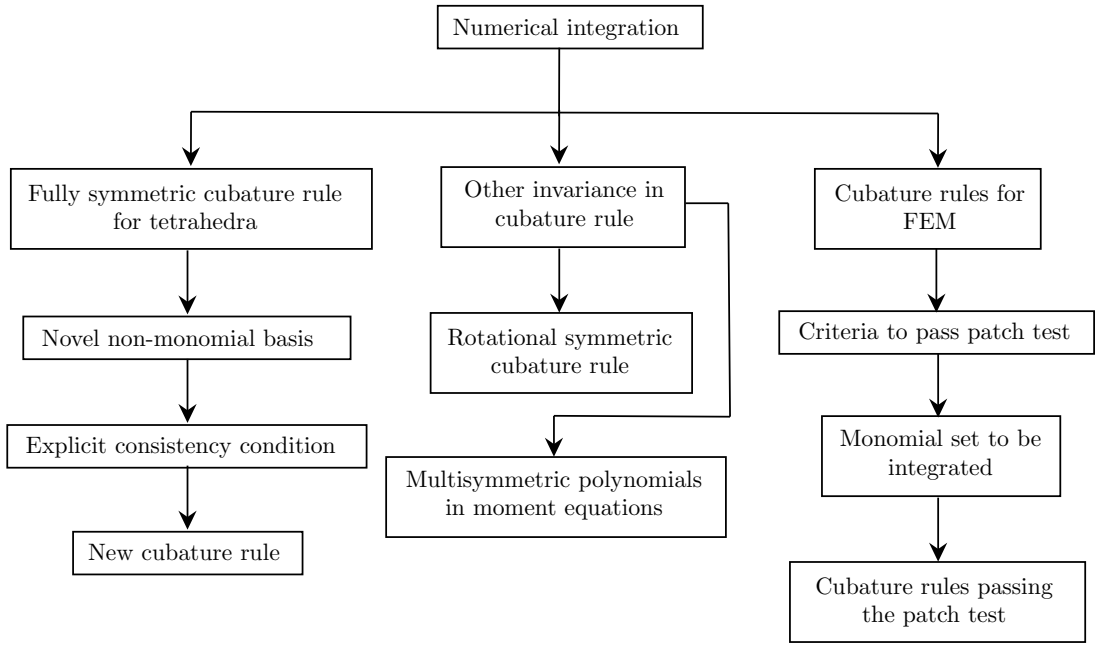


Figure 1.5: Structure of research

lations, where repeated evaluations over multiple time steps are necessary. Consequently, for accurate and efficient finite element simulations, it is crucial to develop cubature rules that deliver precise results with a minimal number of integration points while maintaining high quality.

Convergence is crucial in FE simulations, ensuring that an adequate number of elements yield results that approach the analytical solution of the underlying PDEs. Patch tests, introduced by B. Irons (1966), are essential for verifying simulation convergence. This concept has been further developed by studies such as (W. Zhang & Chen, 1997) and (Stummel, 1979), though Stummel's findings have faced contestation. O. Zienkiewicz and Taylor (1997) explored error estimation techniques' asymptotic effectiveness for various element patterns, while Talischi and Paulino (2014) addressed integration errors to meet patch test requirements for polygonal finite elements.

Today, commercial element developers are expected to rigorously perform patch tests before public release. Experts like MacNeal (1994) and O. C. Zienkiewicz, Taylor, and Zhu (2013) have thoroughly discussed the patch test's role in confirming FE simulation convergence and provided detailed criteria for its successful application. A key criterion is numerical integration, emphasizing its foundational role in convergence research, as discussed in Chapter 6.

1.2 Thesis outline

This thesis focuses on novel numerical integration methods and their application to FEM. Figure 1.5 shows the structure of the research, divided into three parts. Each part begins with Gaussian-type cubature but targets different aspects. The first part focuses on the consistency condition for constructing fully symmetric cubature rules in tetrahedrons, published in (Wang & Papanicolopulos, 2023). The second part explores other symmetries, such as rotational symmetry and multisymmetric polynomials in moment equations. The final part studies FEM formulation and patch tests for new elements, developing a framework to create cubature rules that pass patch tests with fewer integration points.

The remainder of this thesis is structured as follows. Chapter 2 covers foundational concepts and traditional methods for establishing cubature rules, highlighting both indirect and direct approaches. Indirect methods, like product and compound/copied rules, build on existing cubature rules. Direct methods derive new rules by solving nonlinear equations or identifying polynomials that vanish at integration points, a technique currently successful mainly in quadrature. Constructing higher-order cubature rules through direct methods leverages the symmetries and invariances of geometries and cubature rules. Symmetric cubature rules simplify moment equations using symmetric polynomial bases. The consistency condition is crucial for identifying the optimal rule structure, minimizing the number of integration points required.

Chapter 3 presents the first derivation of explicit expressions for the consistency conditions of fully symmetric cubature rules on the tetrahedron. Enhancing the method from previous research (Papanicolopulos, 2015), symmetric polynomials are used to develop a new fully symmetric basis for polynomial space. This innovation simplifies the analysis of moment equations, leading to optimal and quasi-optimal rule structures. Additionally, new cubature rules are developed, and the limitations of the current method for generating consistency conditions are discussed. This part of work has been published in (Wang & Papanicolopulos, 2023).

Chapter 4 explores rotational symmetry in cubature formulations and multisymmetric polynomials in moment equations. A new rotationally symmetric monomial basis is derived by incorporating a Vandermonde polynomial into a fully symmetric polynomial basis, creating a comprehensive system of moment equations. To achieve zero-dimensionality, additional equations are derived from the relationship between the Vandermonde polynomial and the discriminant. Using the Gröbner Bases method, algebraic solutions yield a novel degree 7 PI rule with fewer points than existing optimal fully symmetric PI rules. For higher degree solutions, the PHG code and Levenberg-Marquardt algorithm are used. Additionally, multisymmetric polynomials in cubature formulation for triangles are explored, simplifying the moment equations with elementary multisymmetric polynomials.

Chapter 5 discusses Finite Element formulation and the framework to evaluate an element, essential in Finite Element Analysis (FEA). It starts with the direct formulation approach for linear elasticity problems and examines the complexity of parametric element formulation. The chapter evaluates integrals crucial for FEA accuracy and describes methods for assessing new FEM elements through patch and standard tests. It also addresses issues from reduced integration, such as spurious modes and zero energy modes, and their impact on FE simulation accuracy and stability.

Chapter 6 introduces a new framework for deriving cubature rules that meet patch test standards. This approach begins with an integral to compute internal generalized forces, necessary for passing the patch test. By mapping to parametric space and using various interpolation techniques for displacement and geometry, the required monomial sets for exact integration are specified. This leads to moment equations, whose solutions provide the coordinates and weights for new cubature rules. Using a method similar to (Lyness & Jespersen, 1975), many cubature rules for quadrilateral and hexahedral elements, including novel ones, are independently obtained. However, passing the patch test does not guarantee successful real-world FE simulations, so the cubature rules are evaluated using selected standard tests, and the results are rigorously analyzed. This part of work is under review for publication and preprint is available at SSRN: `https://ssrn.com/abstract=4822034` or `http://dx.doi.org/10.2139/ssrn.4822034`.

Chapter 2

Methodology

2.1 Construction of cubature rules

This thesis focuses on cubature rules for precisely integrating a vector space of algebraic polynomials, with approaches varying based on chosen quality criteria. Fundamental knowledge is drawn from (Cools, 1997; Stroud, 1971).

The most general form of the approximation we consider in this thesis is

$$\int_{R_n} \cdots \int W(x_1,\ldots,x_n) f(x_1,\ldots,x_n) dx_1 \ldots dx_n \simeq \sum_{i=1}^{N} w_i f\left(x_1^{(i)},\ldots,x_n^{(i)}\right) \tag{2.1}$$

where R_n is a real domain within n-dimensional Euclidean space E_n. $W(x_1,\ldots,x_n)$ is a weight function of variables $x_1,...,x_n$. N is the number of integration points and w_i is an independent constant called the coefficients of the formulae or weights of the rules.

A weight function is often used in numerical integration (or cubature) to account for the specific distribution of points or the geometric and functional properties of the domain over which the integration is performed. It essentially adjusts the contribution of each point to the integral, emphasizing some points more than others depending on the problem's structure. In this thesis, $W(x_1,\ldots,x_n) \equiv 1$ because the domain and function being integrated are treated as uniform and focus the analysis on other aspects of the approximation, like the distribution of points or the selection of coefficients w_i, rather than on the intricacies of handling a weight function.

Definition 1. *Formulae (2.1) has degree d if it is exact for all polynomials in $x_1,...,x_n$ of degree at most d and if it is not exact for at least one polynomial of degree d + 1.*

Cubature rules are established through direct and indirect methods. Initially, a brief overview of indirect methods is provided with two-dimensional examples, though their principles are universally applicable. The main focus, direct methods, will be explored in depth in Section 2.1.2.

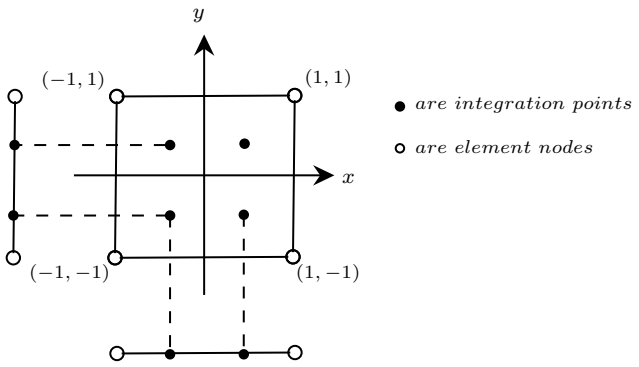


Figure 2.1: Product rule with $N_x = N_y = 2$.

Product rules

In FEM numerical integration, the "product rule" refers to applying quadrature rules in each dimension. Quadrature has been more extensively researched than cubature, so many researchers and commercial software use the product of two quadrature rules to estimate integrals over squares.

Definition 2. *The vector space of all algebraic polynomials in n variables of overall degree at most d is denoted by $\mathbb{P}_n^d$.*

In product rules, different quadrature formulas can be applied to each one-dimensional integral, which may vary in limits. If the quadrature formulae in x has degree d_x with N_x points and the formulae in y has degree d_y with N_y points, the resulting cubature rule will be exact for a space of polynomial between $\mathbb{P}_n^d$ and $\mathbb{P}_n^D$ with $d := min\{d_x, d_y\}$ and $D := max\{d_x, d_y\}$, and has $N = N_x \times N_y$ points.

For instance, let a N_x-point quadrature formulae with degree d be

$$\int_0^1 f(x)dx \simeq \sum_{i=1}^{N_x} w_i f(x_i) \tag{2.2}$$

then a $N_x \times N_y$ cubature rule is:

$$\int_0^1 \int_0^1 f(x,y)dxdy \simeq \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} w_i w_j f(x^{(i)}, y^{(j)}) \tag{2.3}$$

which has algebraic degree d in each variable. Figure 2.1 shows an example for $N_x = N_y = 2$.

This strategy is favored in FEM calculations for 2D and 3D rectangular or box-shaped elements. It simplifies integral computation in the stiffness matrix and load vector by breaking the integration into simpler, independent one-dimensional tasks.

Compound and copy rules

For unconventional integration areas without existing cubature rules, the area can be divided into smaller, standard regions with known cubature rules. Combining these rules forms a compound rule. If the cubature rule for a standard region lacks accuracy, the region can be further subdivided, and a scaled version of the cubature rule applied to each subregion, repeating the process until the desired accuracy is achieved.

For areas divisible into congruent subregions, the copy rule is relevant. For example, a square can be divided into m^2 identical smaller squares, each with a side length of $1/m$ of the original, applying the original cubature rule to each scaled square. Despite seeming costly, especially in higher dimensions, this method is attractive due to the ease of obtaining an error expansion.

2.1.1 Direct formulation and moment equations

This thesis focuses on the direct approach to constructing cubature rules, outlined in the following framework. A scaled integral of a function $f(x_1,...,x_n)$ over a domain $\Omega \subset \mathbb{R}^n$ with n-volume V is defined by:

$$I(f(x_1,...,x_n)) = \frac{\int_{\Omega} \dots \int f(x_1,\dots,x_n)dx_1 \dots dx_n}{V}. \tag{2.4}$$

A cubature rule of Gaussian type is defined as:

$$Q(f(x_1,...,x_n)) = \sum_{j=1}^{N} w_j f\left(x_1^{(j)},...,x_n^{(j)}\right) \tag{2.5}$$

where N is the number of integration points. Cubature rule (2.5) approximates (2.4) by a weighted sum of the values of the function at specific integration points $\left(x_1^{(j)},...,x_n^{(j)}\right)$:

$$I(f(x_1,...,x_n)) \approx Q(f(x_1,...,x_n)). \tag{2.6}$$

To develop a cubature rule precise for all polynomials in a given vector space, it is necessary and sufficient to ensure accuracy for any basis of that space. Therefore, a cubature rule must accurately integrate the basis functions $f_i(x_1,...,x_n)$ of the vector space $\mathbb{P}_n^d$ across N integration points as follows:

$$I(f_i(x_1,\dots,x_n)) = Q(f_i(x_1,\dots,x_n)) \quad i = 1 \dots dim\left(\mathbb{P}_n^d\right). \tag{2.7}$$

Solving Equations 2.7 derives cubature rules known as total degree rules.

When the functions $f_i = f_i(x_1, ..., x_n)$ are monomials, the left-hand side of Equation (2.7), known as moments, can be known in closed form or directly evaluated. By expressing the right-hand side as a weighted sum of the monomial values at specific points $\boldsymbol{x}^{(j)} = \left(x_1^{(j)}, ..., x_n^{(j)}\right)$ and fixing the number of points N, a system of nonlinear moment equations is established as shown in Equations (2.8). Solving these equations provides the coordinates $\boldsymbol{x}^{(j)}$ and weights w_j for the cubature rules.

$$I(f_i(\boldsymbol{x})) = \sum_{j=1}^{N} w_j f_i\left(\boldsymbol{x}^{(j)}\right) \quad i = 1 \ldots dim\left(\mathbb{P}_n^d\right) \tag{2.8}$$

In developing cubature rules directly, two distinct methods emerge:

- Identifying polynomials that vanish at the designated points of the formulae.
- Directly solving the system of nonlinear equations.

The former strategy, known as interpolatory cubature, has been successful in one-dimensional quadrature. However, most published cubature rules use the latter approach, called direct cubature. For non-interpolatory cubature rules, Steinitz's Austauschsatz (Davis, 1967) allows creating an interpolatory formula using a subset of the original points. The groundwork for constructing direct cubature is detailed in Section 2.1.2, with the polynomial search approach introduced here.

Definition 3. *If the weights of a cubature rule of degree d are uniquely determined by the points, the cubature rule is called an interpolatory cubature rule.*

Theorem 1 (Tchakaloff Theorem)**.** *Let I be an integral over an n-dimensional region Ω with a weight function that is non-negative in Ω and for which the integrals of all monomials exist. Then a cubature rule of degree d with $N < dim\left(\mathbb{P}_n^d\right)$ points exists with all points inside Ω and all weights positive.*

$Proof$: This theorem has been proved by many researches like (Bayer & Teichmann, 2006). □

When the points of a cubature rule are known, Equation (2.8) presents a system of $dim\left(\mathbb{P}_d^n\right)$ linear equations for N unknown weights. According to Theorem 1, an interpolatory cubature rule implies $N < dim\left(\mathbb{P}_d^n\right)$, ensuring the existence of N linearly independent polynomials $U_1, \ldots, U_N \in \mathbb{P}_d^n$ satisfying:

$$det\left(U_i\left(\boldsymbol{x}^{(j)}\right)_{i,j=1,\ldots,N}\right) \neq 0. \tag{2.9}$$

These polynomials U_i span a maximal vector space of polynomials that do not vanish at any of the given points. To fully span the vector space $\mathbb{P}_d^n$, an additional set of $k = dim\left(\mathbb{P}_d^n\right) - N$ polynomials $p_1, \ldots, p_k$ is required, forming a complete basis $S = \{U_1, \ldots, U_N, p_1, \ldots, p_k\}$ for $\mathbb{P}_d^n$, where:

$$span(S) = \mathbb{P}_d^n. \tag{2.10}$$

All polynomials in $\mathbb{P}_d^n$ can be expressed as a linear combination of the basis S, leading to a linear system:

$$p_i\left(\boldsymbol{x}^{(j)}\right) = \sum_{j=1}^{N} c_{ij} U_j\left(\boldsymbol{x}^{(j)}\right) \quad i = 1, \ldots, k, \tag{2.11}$$

with constant coefficients c_{ij}. Solving these equations for c_{ij} yields k linearly independent polynomials

$$R_i = p_i - \sum_{j=1}^{N} c_{ij} U_j \quad i = 1, \ldots, k \tag{2.12}$$

which vanish at the cubature rule points. Substituting the polynomials p_i in S with R_i fully characterizes a cubature rule by R_i, with U_i polynomials defining the weights system.

These characterizing polynomials R_i bridge orthogonal polynomials and ideal theory with cubature rules, a topic not covered in this thesis. Further details are available in (Stroud, 1971, Chapter3).

2.1.2 Symmetries and invariant theory for cubature rule

Solving the nonlinear Equations (2.8) directly determines the coordinates $\boldsymbol{x}^{(j)}$ and weights w_j of cubature rules. However, as the degree d and number of total variables increase, the complexity escalates due to the growing number $n_e = dim\left(\mathbb{P}_d^n\right)$ of equations. This complexity makes discovering new cubature rules challenging, highlighting the need to examine the integrand and cubature rule properties to streamline the computational process.

This thesis focuses on domains like triangles, tetrahedrons, squares, and cubes, which have geometric symmetries. Priority is given to cubature rules reflecting these symmetries in the distribution of integration points and weights. Incorporating symmetries into cubature formulations requires knowledge of group theory and invariant theory.

Let G be a finite group of transformations $g : \mathbb{R}^n \to \mathbb{R}^n$ and $|G|$ is the order of the group (also called the cardinality of group) meaning the number of elements in the group.

Definition 4. *A set $\Omega \subset \mathbb{R}^n$ is invariant with respect to G if it does not change under any transformation of the group, that is $g(\Omega) = \Omega \; \forall g \in G$. A function $f(\boldsymbol{x})$ is invariant with respect to G if it does not change under any transformation of the group, that is $f(g(\boldsymbol{x})) = f(\boldsymbol{x}) \; \forall g \in G$*

Definition 5. *The G-orbit of a point $\boldsymbol{x} \in \mathbb{R}^n$, denoted by $G(\boldsymbol{x})$, is the set $\{g(\boldsymbol{x}) : g \in G\}$. The point $\boldsymbol{x}$ is called a generator of $G(\boldsymbol{x})$.*

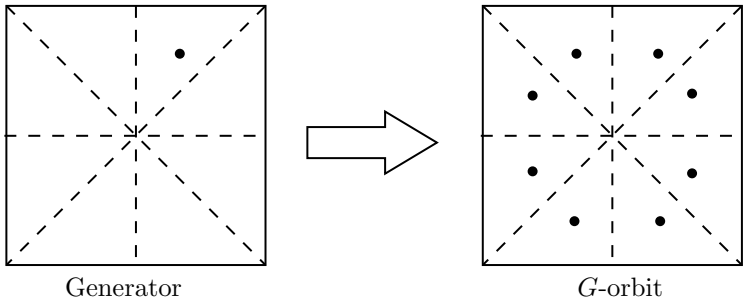


Figure 2.2: A G-orbit generated by a generator $\boldsymbol{x}$.

A G-orbit of a given point is obviously an invariant set w.r.t. G. The number of points in an orbit depends on the position of generator $\boldsymbol{x}$ in the domain.

For example, consider a square C_2 with $\{(x,y)|-1 \leq x,y \leq 1\}$, G is a group of linear transformation for which C_2 is G-invariant. The elements g_i with $i = 1,\ldots,8$ in the group G are expressed as matrices:

$$\begin{aligned} g_1 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, g_2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, g_3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, g_4 = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}, \\ g_5 &= \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, g_6 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, g_7 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, g_8 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \end{aligned} \tag{2.13}$$

For a generator

$$\boldsymbol{x} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \tag{2.14}$$

its G-orbit is generated by

$$O_G(\boldsymbol{x}) = \{g_i(\boldsymbol{x})|i = 1...8\}, \tag{2.15}$$

which written in the classic round brackets form is

$$\{(\alpha,\beta),(-\alpha,-\beta),(\beta,\alpha),(-\beta,-\alpha),(-\alpha,\beta),(\alpha,-\beta),(-\beta,\alpha),(\beta,-\alpha)\}. \tag{2.16}$$

Figure 2.2 shows the G-orbit generated by the generator $\boldsymbol{x}$.

Definition 6. *A cubature rule is invariant with respect to the group G if the region Ω is G-invariant (i.e. $g(\Omega) = \Omega\ \forall g \in G$), the set of integration points is a union of G-orbits, and all points in the same orbit have the same weight.*

The value of symmetry groups to the construction of cubature rules was mentioned by Sobolev in his work (Sobolev, 1962). $\mathbb{P}_d^n$ is a G-invariant vector space of algebraic polynomials, so that $g(f) \in \mathbb{P}_d^n$ for all $f \in \mathbb{P}_d^n$ and $g \in G$. The subspace $\mathbb{P}_d^n(G) \subset \mathbb{P}_d^n$, is a vector space of G-invariant algebraic polynomials

$$\mathbb{P}_d^n(G) = \{f \in \mathbb{P}_d^n | g(f) = f \text{ for all } g \in G\} \tag{2.17}$$

Theorem 2. *Let G be a finite group of linear transformations acting on $\mathbb{P}_d^n$. Then, every G-invariant linear functional on $\mathbb{P}_d^n$ is determined by its restriction to $\mathbb{P}_d^n(G)$.*

To prove the Theorem 2, we need some derivations. Assume $h \in G$ and $f \in \mathbb{P}_d^n$ then

$$h\left(\sum_{g\in G} g(f)\right) = \sum_{g\in G} h(g(f)) \tag{2.18}$$

and because $hg \in G$ Equation (2.18) can be written as

$$\sum_{g\in G} h(g(f)) = \sum_{g\in G} g(f) \tag{2.19}$$

which means that

$$f_G = \sum_{g\in G} g(f) \in \mathbb{P}_d^n(G). \tag{2.20}$$

Let I be a G-invariant linear functional on $\mathbb{P}_d^n$, we have

$$I(g(f)) = I(f) \tag{2.21}$$

for all $f \in \mathbb{P}_d^n$ and $g \in G$. Let $M = |G|$, then we have

$$I(f) = \frac{1}{M}\sum_{g\in G} I(g(f)) = I\left(\frac{1}{M}\sum_{g\in G} g(f)\right) = I\left(\frac{1}{M} f_G\right) \tag{2.22}$$

From Equations (2.18), (2.19) and (2.22) we can see for any $f \in \mathbb{P}_d^n$ there is a $f_G \in \mathbb{P}_d^n(G)$ such that the G-invariant lineal functional gives the same result for both f and f_G. Theorem 2 proved. □

For example, consider the group $G = \mathbb{Z}_2$, which consists of two elements: the identity transformation and a reflection across the x-axis. This is a finite group of linear transformations acting on $\mathbb{R}^2$.

Now, let $\mathbb{P}_1^2$ be the space of linear polynomials in two variables, which includes polynomials of the form

$$a_1 x + a_2 y \tag{2.23}$$

where a_1 and a_2 are coefficients. The group G acts on these polynomials by applying the reflection and identity transformations to the variables x and y.

A G-invariant linear functional is a linear functional that remains unchanged under the group action of G. For example, the integral of the polynomial over a symmetric region (like a circle) is invariant under reflections.

According to the theorem, every G-invariant linear functional is determined by its restriction to $\mathbb{P}_1^2(G)$, which refers to the subspace of G-invariant polynomials. In this case, a G-invariant polynomial would be one that does not change under reflection, such as a polynomial of the form

$$b_1 x^2 + b_2 y^2, \tag{2.24}$$

where the linear terms vanish under the reflection action of G.

Thus, the invariant functional can be computed solely by considering the action on the space of G-invariant polynomials, simplifying the overall problem of evaluating the functional.

Then we can have the so-called Sobolev's theorem.

Corollary 2.1 (Sobolev's Theorem)**.** *Let the cubature formula Q be G-invariant. The cubature formula has degree d if it is exact for all G-invariant polynomials of degree at most d and if it is not exact for at least one polynomial of degree d + 1.*

A G-invariant cubature rule can be written as

$$Q(f(\boldsymbol{x})) = \sum_{i=1}^{N^*} w_i M_i f\left(\boldsymbol{x}^{(i)}\right) \tag{2.25}$$

where N^* is the number of orbits and M_i is the number of points in orbit i which satisfy:

$$N = \sum_{i=1}^{N^*} M_i. \tag{2.26}$$

If we let $w_i' = w_i M_i$, Equation (2.25) becomes

$$Q(f(\boldsymbol{x})) = \sum_{i=1}^{N^*} w_i' f\left(\boldsymbol{x}^{(i)}\right). \tag{2.27}$$

Invariant theory is an important tool for the study of the vector spaces of invariant polynomials. It helps to simplify the system of nonlinear equations.

Definition 7. *If every invariant polynomial of G with degree up to at most d in n variables can be expressed in a set of G-invariant polynomials $\{f_1,\ldots,f_m\}$, then this set of polynomials is said to be an integrity basis for the vector space $\mathbb{P}_d^n(G)$. Each polynomial f_i is called a basic invariant polynomial of $\mathbb{P}_d^n(G)$.*

Leveraging the region's symmetry to structure the cubature formula significantly aids in developing new rules. The necessary and sufficient conditions in (2.7) are simplified using a G-invariant cubature formula (2.27), reducing the system of nonlinear equations to:

$$I(f_i(\boldsymbol{x})) = \sum_{j=1}^{N^*} w'_j f_i\left(\boldsymbol{x}^{(j)}\right) \quad \forall f_i \in B_G. \tag{2.28}$$

Here, f_i is one of the basic invariant polynomials and B_G represents a basis for the vector space $\mathbb{P}_d^n(G)$, which includes all G-invariant polynomials of degree d in n variables. A larger symmetry group G reduces the dimension of this space, simplifying the task of finding new cubature rules by decreasing the number of variables and equations involved.

Since a linear transformation does not change a polynomial's degree, homogeneous polynomials are typically chosen as basic invariant polynomials. A homogeneous polynomial is a polynomial in which all the terms have the same total degree. If the number of these polynomials m exceeds the number of variables n, they are not algebraically independent, and polynomial equations known as syzygies exist among $\{f_1,\ldots,f_m\}$. Recognizing and accounting for syzygies is crucial in determining the dimension of the invariant polynomial vector space, aiding the development of new cubature rules.

A G-invariant cubature rule, also called a symmetric cubature rule, can exhibit full symmetry or rotational symmetry, which are the two most important symmetries for this thesis.

Definition 8. *A region and a cubature rule in an n-dimensional cube are termed fully symmetric when they are invariant with respect to the following group of transformations:*

$$G_{CFS} = \{(x_1,\ldots,x_n) \to (s_1 x_{r_1},\ldots,s_n x_{r_n}) | s_i \in \{-1,1\}, i = 1,\ldots,n, \{r_1,\ldots,r_n\} = \{1,\ldots,n\}\}. \tag{2.29}$$

Orbit (2.16) is a fully symmetric orbit (G_{CFS}-orbit), and cubature rules consisting only of fully symmetric orbits are fully symmetric cubature rules.

Definition 9. *A region and a cubature rule for an n-simplex, expressed in barycentric coordinates, are termed fully symmetric when they are invariant with respect to the following group of transformations:*

$$G_{FS} = \{(L_1,\ldots,L_n) \to (L_{r_1},\ldots,L_{r_n}) | \{r_1,\ldots,r_n\} = \{1,\ldots,n\}\}. \tag{2.30}$$

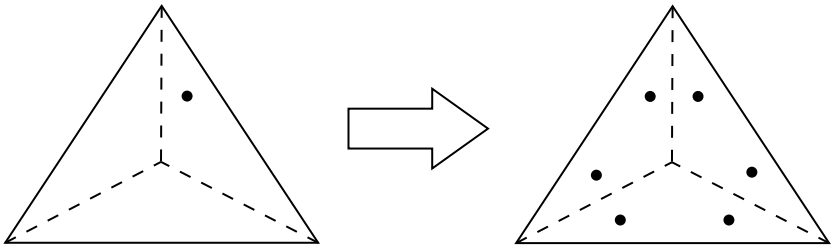


Figure 2.3: A G_{FS}-invariant orbit on a triangle.

Figure 2.3 shows an instance of fully symmetric orbit on a triangle. The definition of barycentric coordinates will be given in Section 3.1.1.

Let S_{ij} with $i \neq j$ be an operation that switch ith and jth component of the coordinates, for example, given a point (α, β, γ), we have

$$S_{12}((\alpha, \beta, \gamma)) = (\beta, \alpha, \gamma). \tag{2.31}$$

Definition 10. *A region and a cubature rule for an n-simplex, expressed in barycentric coordinates, are termed rotationally symmetric when they are invariant with respect to the following group of transformations:*

$$G_{RS} = \{(L_1, \ldots, L_n) \to \{S_{ij}(S_{kl}((L_1, \ldots, L_n)))|i, j, k, l \in \{1, \ldots, n\}, i \neq j, k \neq l, S_{ij} \neq S_{kl}\}\} \tag{2.32}$$

For instance, consider a triangle with a generator (L_1, L_2, L_3) has a rotationally symmetric orbit:

$$G_{RS}((L_1, L_2, L_3)) = \{(L_1, L_2, L_3), (L_3, L_1, L_2), (L_2, L_3, L_1)\} \tag{2.33}$$

Figure 2.4 shows an example of rotationally symmetric orbit (G_{RS}-orbit) for a generator $\boldsymbol{x} = (L_1, L_2, L_3)$ on a triangle. The cubature rules that consist of only rotationally symmetric orbits are rotationally symmetric cubature rules.

2.2 Consistency conditions

The so-called rule structure of a symmetric cubature rule is defined as

$$[n_0, n_1, \ldots, n_{p-1}] \tag{2.34}$$

where p is the number of orbit type, n_i is the number of type-i orbit in the symmetric cubature rule which satisfy

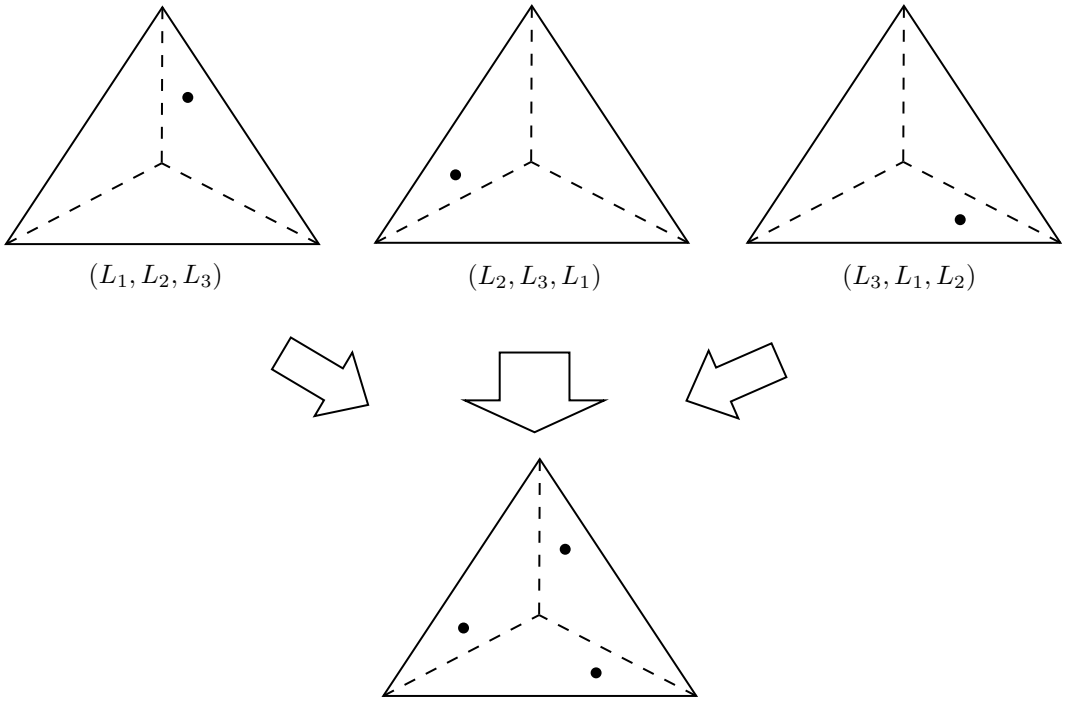


Figure 2.4: A G_{RS}-invariant orbit on a triangle.

$$\sum_{i=0}^{p-1} n_i = N^*. \tag{2.35}$$

Equation (2.27) then can be rewritten as

$$Q(f(\boldsymbol{x})) = \sum_{i=0}^{p-1} \sum_{j=1}^{n_i} w'_{ij} f\left(\boldsymbol{x}^{(ij)}\right), \tag{2.36}$$

and the moment equations (2.28) become:

$$I(f_i(\boldsymbol{x})) = \sum_{t=0}^{p-1} \sum_{k=1}^{n_t} w'_{tk} f_i\left(\boldsymbol{x}^{(tk)}\right) \quad \forall f_i \in B_G. \tag{2.37}$$

For any given rule structure, the symmetric cubature rule involves fewer unknown parameters than a generic cubature rule with a similar number of nodes. This disadvantage is offset by a significant reduction in the number of moment equations that need to be satisfied, as indicated by Theorem 2.1.

Developing a cubature rule requires solving a system of nonlinear moment equations, often posing significant computational challenges due to their size. However, prior knowledge of likely solution structures, encapsulated in consistency conditions, can guide the construction of optimal cubature rules.

Consistency conditions are based on assumptions about the number of integration points required for a cubature rule of a given degree (though not definitively accurate). These assume the moment equations and their subsystems are consistent only if not overdetermined. This is expressed as:

$$n_e \geq m_e \tag{2.38}$$

where n_e is the number of unknowns in the system, and m_e is the number of moment equations.

Establishing rule structures that meet consistency conditions is crucial for constructing a cubature rule with a specified degree of precision. The rule's structure influences the number of nodes and defines the relationships among the unknowns in the reduced moment equations. These conditions, expressed as linear inequalities, ensure that each subsystem has at least as many unknowns as equations. This aims to secure a certain precision d for the cubature rule. However, due to the complexity and nonlinearity of the moment equations, these conditions are neither necessary nor sufficient to guarantee a solution.

2.3 Solution of a polynomial system

Polynomial systems consist of equations set to zero, aiming to find variable values that satisfy all equations simultaneously. This area is foundational in algebraic geometry and computational mathematics, focusing on finding polynomial roots. Polynomials are central in abstract algebra and widely applicable in science and engineering, making this task a long-standing focus for mathematicians.

Efforts to solve polynomial systems date back to the Babylonians, with significant advancements during the Renaissance through solutions of cubic and quartic equations. In the 19th century, Niels Henrik Abel and Évariste Galois showed that general solutions in radicals for polynomials of degree five or higher are not feasible. This discovery led to Galois theory, enhancing the understanding of symmetries and group structures in polynomial equations.

Today, the resolution of polynomial systems has advanced significantly due to computational methods and algorithms, including:

- **Homotopy Continuation**: Transforms a complex polynomial system into a simpler one with known solutions, tracking these solutions through a continuous deformation or homotopy to the original system.
- **Gröbner Bases**: Converts a polynomial system into a simpler configuration, clarifying the solution space structure and facilitating solution finding.

- **Iterative methods**: Uses successive approximations to refine an initial guess towards the solution, useful when direct solutions are unfeasible.

These modern computational strategies enable precise searching or approximation of solutions, providing deeper insights into the solution space's structure and the dynamics of system solutions.

A generic system of m polynomial equations with n variables $v_1, v_2, .., v_n$ and real coefficients is overdetermined if $m > n$ and underdetermined if $m < n$.

A solution to a polynomial system is a tuple $(v_1, v_2, ..., v_n)$ that satisfies all the system's equations. A system is inconsistent if it has no solution and consistent if it has at least one. A consistent system is zero-dimensional if it has a finite number of solutions and positive-dimensional if it has infinitely many. For more details, see (Lazard, 2009).

Iterative methods like Newton's method can approximate individual solutions but not systematically find all solutions. We use the Levenberg-Marquardt algorithm, a refined version of Newton's method for optimizing nonlinear least squares problems. This method combines gradient descent and the Gauss-Newton method, offering greater flexibility and broader applicability than standard Newton's method.

Defining a complete solution set for positive-dimensional systems is challenging. Algebraic geometry can simplify the system for numerical resolution, using methods like Gröbner bases or, for zero-dimensional systems, rational univariate representation. However, the Homotopy Continuation method has been ineffective in our cases.

The Gröbner bases method is selected for our symbolic computations, which are carried out using the software $Maple$. A Gröbner basis for an ideal I in a polynomial ring is a set of polynomials $\{g_1, g_2, \ldots, g_t\}$ that generates the same ideal as the original set of polynomials but has desirable properties that simplify many algebraic operations. It is extensively used in symbolic computations to manipulate polynomials algebraically, for instance, in simplifying expressions or in determining properties of algebraic varieties.

Gröbner bases allow for the systematic elimination of variables from polynomial systems, making it easier to find solutions. For instance, given a system of multivariate polynomial equations, the Gröbner basis transforms the system into a simpler equivalent form, often triangular in structure, which can be solved step by step, akin to back substitution in linear systems. Besides, given a polynomial and an ideal, a Gröbner basis can be used to test whether the polynomial belongs to the ideal. This is crucial for many applications in algebraic geometry and coding theory.

Gröbner bases can also reduce polynomials into a canonical form with respect to a given term ordering. This makes it easier to compare polynomials and perform operations like division, simplification, and checking for redundancy in a set of generators for an ideal.

2.4 Quality of cubature rules

Each cubature rule is evaluated based on two criteria, represented by two letters. The first letter indicates the integration weights: 'P' for all positive, 'N' if at least one is negative (but all real), and 'C' if any weight is complex. The second letter describes the integration points: 'I' if all points are inside the domain, 'B' if at least one point is on the boundary (others inside), 'O' if at least one point is outside the domain (but all real), and 'C' if any point has complex coordinates. Preferred qualities, in descending order, are PI, NI, PB, NB, PO, NO, PC, NC, and CC.

For problems such as elliptic PDEs, the stiffness matrix is expected to be positive definite, meaning all of its eigenvalues should be positive. This property ensures that the resulting system of equations has a unique and stable solution. Positive definiteness reflects the physical nature of the problem, for example, diffusion or elasticity, where the energy is expected to increase as the solution moves away from equilibrium.

When cubature rules with negative weights are used to approximate the integrals, the resulting stiffness matrix entries can be negative or improperly weighted. This can distort the matrix's structure and lead to negative eigenvalues. In physical terms, negative eigenvalues imply that the system has an unstable mode or that the energy is not bounded from below, which is unphysical in many cases (e.g., in elasticity or diffusion problems).

In FEM, matrices like the stiffness matrix need to be well-conditioned for stable and accurate solutions. The conditioning of the matrix is highly dependent on the cubature rule's ability to correctly approximate the integrals. Negative weights can introduce instability in the form of:

- Spurious or non-physical oscillations in the solution.
- Poorly conditioned matrices that make iterative solvers (like Conjugate Gradient) inefficient or fail to converge.

Negative eigenvalues are a symptom of these instabilities because they indicate that the system is no longer behaving as expected. For example, in structural mechanics, this could indicate that the structure is unstable, and in fluid dynamics, it could lead to non-physical flow patterns. Therefore, in practical FEM applications, positive weights are generally preferred for quadrature and cubature rules to maintain matrix stability and accuracy, ensuring that the discretized PDE behaves as expected.

What's more, ensuring that the integration points lie inside the integration domain has several important implications for accuracy, stability, and physical realism in FEM calculations. Except the purpose of prevents numerical errors and instabilities like positive weights, inside-domain integration points ensure that integrals are computed correctly and consistently over the domain and respects the geometric and physical boundaries of the finite element domain, ensuring that the numerical method remains consistent with the underlying PDE and its weak form. Therefore, cubature rules with PI quality is the most favourable to FEM.

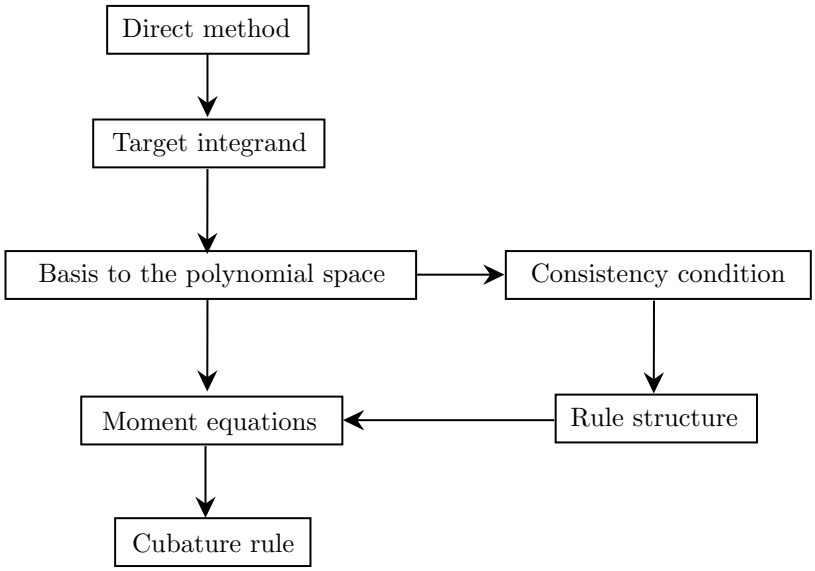


Figure 2.5: General idea to construct Gaussian type cubature rule

2.5 Summary

This chapter introduces the fundamental concepts and traditional methods for establishing cubature rules. Indirect methods, like product and compound/copied rules, build on existing rules. In contrast, novel cubature rules are mainly derived through direct methods, involving solving nonlinear equations or identifying polynomials that vanish at integration points, though this has been more successful in quadrature.

Constructing higher-order cubature rules from nonlinear equations is greatly enhanced by leveraging symmetries and invariances in the geometries and rules. Symmetric cubature rules use a symmetric polynomial basis, simplifying and resolving moment equations. Consistency conditions help identify the optimal rule structure with the fewest integration points. Figure 2.5 illustrates the approach to constructing a Gaussian-type cubature rule by directly solving moment equations.

Identifying a new cubature rule starts with defining the target integrand. For a total degree cubature rule with degree d, the integrand includes all polynomials up to degree d within the integration domain. Instead of generating moment equations for all polynomials, a basis of the vector space formed by these polynomials is used. Monomials up to degree d generate the moment equations. Considering the symmetries and invariances of the geometry and cubature rule, integration points are categorized into orbits. Consistency conditions determine the quantity of each orbit in the cubature formula. Solving these moment equations yields the new cubature rules.

Chapter 3

Consistency Conditions for Fully-symmetric Cubature on the Tetrahedron

3.1 Theoretical background

3.1.1 Barycentric coordinates

Barycentric coordinates offer a transformative approach for defining point positions within simplices, the fundamental polytopes in any dimension. Introduced by August Ferdinand Möbius in the 19th century, they are essential in computer graphics, geometric modeling, numerical analysis, and physics. Also known as area coordinates for triangles and volume coordinates for tetrahedra, barycentric coordinates provide a system relative to a simplex, making them invaluable for geometric challenges.

Central to barycentric coordinates is the idea that any point inside a simplex can be represented as a weighted average of the simplex's vertices, with the weights summing to one. Figure 3.1 illustrates this with a triangle subdivided into three areas S_{ABO}, S_{ACO} and S_{BCO} by a point O.

The barycentric coordinates, denoted as weights L_i, are calculated by the following equations:

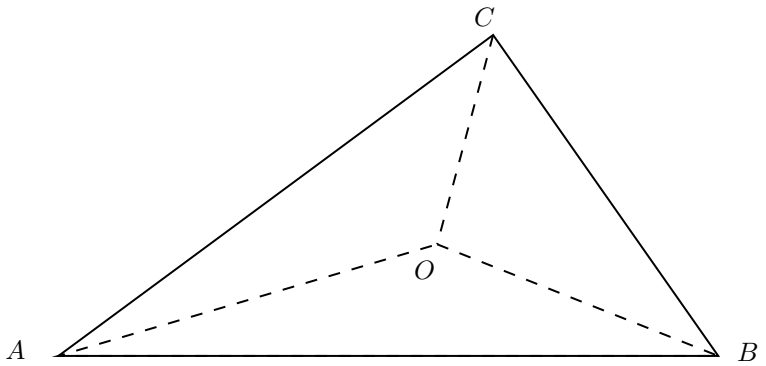


Figure 3.1: A triangle divided into three areas.

$$L_1 = \frac{S_{ABO}}{S_{ABO} + S_{ACO} + S_{BCO}} \tag{3.1a}$$

$$L_2 = \frac{S_{ACO}}{S_{ABO} + S_{ACO} + S_{BCO}} \tag{3.1b}$$

$$L_3 = \frac{S_{BCO}}{S_{ABO} + S_{ACO} + S_{BCO}} \tag{3.1c}$$

$$\tag{3.1d}$$

Barycentric coordinates, rooted in affine geometry, link algebraic expressions to geometric constructions. Unlike Cartesian coordinates, which rely on a fixed reference frame, barycentric coordinates are tailored to the simplex's geometry, enhancing intuitive understanding and manipulation of geometric relationships.

The generalization of Barycentric coordinates exists for shapes other than simplexes, although the process and properties are more complex. Barycentric coordinates are typically defined for simplexes because they offer a natural and convenient way of expressing a point inside the shape as a weighted combination of its vertices. However, various extensions of barycentric coordinates have been developed for polytopes and other more general shapes. For exmaple, for convex polygons, generalized barycentric coordinates (often referred to as harmonic, Wachspress, or mean value coordinates) can be used to express any point inside the polygon as a weighted sum of its vertices. They retain several important properties of classical barycentric coordinates, such as non-negativity (weights are positive inside the shape) and partition of unity (the sum of the weights equals 1). However, they are often more complicated to compute because of the more intricate geometry of the polytope.

In numerical methods like FEM, barycentric coordinates are crucial for formulating and solving PDEs over varied domains. They precisely define functions and variables on the complex geometries of mesh elements, enhancing numerical approximations' precision and efficiency.

Consider a tetrahedron defined by four vertices with Cartesian coordinates $(x_{|\kappa}, y_{|\kappa}, z_{|\kappa})$ where $\kappa = 1 \dots 4$. A point with Cartesian coordinates (x, y, z) can be described, in relation to the tetrahedron, by the barycentric coordinates (L_1, L_2, L_3, L_4) such that

$$x = L_1 x_{|1} + L_2 x_{|2} + L_3 x_{|3} + L_4 x_{|4} \tag{3.2a}$$

$$y = L_1 y_{|1} + L_2 y_{|2} + L_3 y_{|3} + L_4 y_{|4} \tag{3.2b}$$

$$z = L_1 z_{|1} + L_2 z_{|2} + L_3 z_{|3} + L_4 z_{|4} \tag{3.2c}$$

$$1 = L_1 + L_2 + L_3 + L_4 \tag{3.2d}$$

Barycentric coordinates are interdependent, as shown in Equation (3.2d). While they increase the number of unknowns, they simplify calculations, especially with symmetry. Barycentric coordinates can be defined without the normalization in Equation (3.2d), but with this normalization, they are called volume coordinates.

In terms of polynomial representation, a polynomial of degree d in Cartesian coordinates can be reformulated as a homogeneous polynomial in barycentric coordinates. Specifically, it can be expressed as a linear combination of monomials $L_1^i L_2^j L_3^k L_4^{d-i-j-k}$, where L_1, L_2, L_3 and L_4 are the barycentric coordinates.

3.1.2 Symmetric polynomials

In algebra, symmetric polynomials are essential, connecting basic operations with fields like algebraic geometry, number theory, and invariant theory. Defined by their invariance to variable permutations, their complex structure impacts both theoretical exploration and practical applications.

A symmetric polynomial is a multivariate polynomial that remains unchanged under any permutation of its variables (Vinberg, 2003). For n variables $v_1, v_2, \ldots, v_n$, the elementary symmetric polynomials, denoted $\tilde{v}_k$, are defined as the sum of all products of k distinct variables v_i, with an alternating sign based on the parity of k:

$$\tilde{v}_k = (-1)^k \sum_{1 \le i_1 < i_2 < \ldots < i_k \le n} v_{i_1} v_{i_2} \ldots v_{i_k} \tag{3.3}$$

where $\tilde{v}_0 = 1$. This summation encompasses all possible combinations of k distinct indices from the set $\{1 \ldots n\}$. For example, consider $n = 2$ we have two elementary symmetric polynomials:

$$\tilde{v}_1 = -(v_1 + v_2) \tag{3.4a}$$

$$\tilde{v}_2 = v_1 v_2 \tag{3.4b}$$

A symmetric polynomial

$$V = v_1^2 v_2 + v_2^2 v_1 + v_1 v_2 - v_1 - v_2 \tag{3.5}$$

therefore can be expressed as

$$V = -\tilde{v}_1 \tilde{v}_2 + \tilde{v}_1 + \tilde{v}_2. \tag{3.6}$$

Every symmetric polynomial in the variables v_i can be uniquely represented as a polynomial in the elementary symmetric polynomials $\tilde{v}_k$. The actual values of v_i are derivable from $\tilde{v}_k$ by solving the polynomial equation for v:

$$\sum_{j=0}^{n} \tilde{v}_{n-j} v^j = 0. \tag{3.7}$$

which, for instance, when $n = 2$ becomes

$$v^2 + \tilde{v}_1 v + \tilde{v}_2 = 0 \tag{3.8}$$

This foundational theorem highlights the universality and adaptability of symmetric polynomials, allowing any symmetric polynomial to be reduced to a finite set of elementary symmetric polynomials for analysis.

Symmetric polynomials are crucial in more than just mathematical theory. In equation theory, they help eliminate variables and solve polynomial equations using symmetric functions of the roots. In algebraic geometry, they are essential for classifying polynomial functions and studying algebraic varieties, particularly using resultants and discriminants.

Furthermore, in representation theory, symmetric polynomials are key to invariant theory for symmetric groups and the theory of Schur functions, which are important in combinatorics and the representation theory of general linear groups. In number theory, they help explore Galois groups and fields formed by polynomial roots, providing insights into the solvability of equations and the structure of algebraic numbers.

3.2 Bases for cubature on the tetrahedron

Cubature bases rely on polynomial exactness, requiring accurate integration of polynomials up to a specified degree, as stated in Theorem 2.1. Choosing the right polynomial basis is crucial for efficiency and accuracy, simplifying the rule's construction and enhancing numerical stability.

In cubature formulation, a polynomial basis spans all polynomials up to a certain degree within the integration domain. Common bases include:

- **Monomial bases**: Consisting of monomials $x^i y^j$, they are straightforward but can lead to numerical instability due to ill-conditioning.
- **Orthogonal polynomials**: These are orthogonal with respect to a weight function on the integration domain, resulting in well-conditioned systems and better numerical properties.

- **Other bases**: Depending on the integration domain's geometry and specific problem requirements.

This thesis discusses only monomial and non-monomial bases, as orthogonal polynomial bases are more commonly used in quadrature than cubature.

3.2.1 Asymmetric basis

Consider the vector space of polynomials in three dimensions with degree up to d and no symmetry, a basis of $\mathbb{P}_d^3$ is the set of monomials $x^i y^j z^k$ with total degree $i+j+k \leq d$, which has dimension

$$m_{\text{asym}} = \frac{(d+3)(d+2)(d+1)}{6}. \tag{3.9}$$

In this case there are four unknowns for every integration point (three coordinates and one weight) so the consistency condition is

$$4n_K \geq m_{\text{asym}}. \tag{3.10}$$

This consistency condition applies to any domain in $\mathbb{R}^3$ and is relevant for asymmetric cubature rules, which are not necessarily invariant. Typically, for degree $d = 9$, the number of points n_K should be at least 55. However, exceptions exist, such as cubature rules using only 52 points on a cube (Mantel & Rabinowitz, 1977) and 53 points on a tetrahedron (Beckers & Haegemans, 1990).

For tetrahedra, a practical basis is the set of monomials $L_1^i L_2^j L_3^k L_4^{d-i-j-k}$ of total degree d. Barycentric coordinates facilitate describing integration points independent of tetrahedron shape and streamline invariant rule formulation, as tetrahedral symmetry can be described through vertex or barycentric coordinate permutations.

3.2.2 Fully symmetric basis

Fully symmetric bases simplify calculations, enhance numerical stability, and improve algorithm efficiency for multidimensional problems. By leveraging problem domain symmetries, they minimize unique evaluations needed for integration. Symmetric regions contribute equally, allowing single computations to be replicated across all counterparts, significantly reducing computational load.

Fully symmetric cubature rules often achieve higher accuracy with fewer points compared to non-symmetric ones, especially for symmetric integrands. This efficiency arises because symmetric bases effectively capture the behavior of symmetric functions, enabling precise integration with fewer points.

Using fully symmetric bases simplifies the formulation and resolution of problems, especially in symmetric domains and functions. This congruence leads to more straightforward expressions for the basis functions and their interrelations. In high-dimensional integration and approximation, the curse of dimensionality is a significant challenge. Fully symmetric bases mitigate this by using problem domain symmetries to reduce the exponential increase in computational resources.

We consider here fully symmetric rules on tetrahedra, which are invariant with respect to the transformations of the symmetric group G_{FS} in Definition 9.

For a specified degree d, the G_{FS}-invariant polynomials are the symmetric polynomials formed from the four barycentric coordinates. These polynomials can be represented by a basis consisting of the products $\tilde{L}_1^t \tilde{L}_2^i \tilde{L}_3^j \tilde{L}_4^k$, where $t+2i+3j+4k=d$. The elementary symmetric polynomials in the barycentric coordinates are defined by equation (3.3) as

$$\tilde{L}_1 = -(L_1+L_2+L_3+L_4) = -1 \tag{3.11a}$$

$$\tilde{L}_2 = L_1L_2+L_1L_3+L_1L_4+L_2L_3+L_2L_4+L_3L_4 \tag{3.11b}$$

$$\tilde{L}_3 = -(L_1L_2L_3+L_1L_3L_4+L_1L_2L_4+L_2L_3L_4) \tag{3.11c}$$

$$\tilde{L}_4 = L_1L_2L_3L_4. \tag{3.11d}$$

According to equation (3.11a), the G_{FS}-invariant basis comprises products $\tilde{L}_2^i \tilde{L}_3^j \tilde{L}_4^k$ where $2i+3j+4k \leq d$. The number of elements in this basis, and consequently the number of moment equations m_e, equals the count of non-negative integer solutions to $2i+3j+4k \leq d$, which can be calculated as per (OEIS A001400, 2022b):

$$m_\mathrm{e} = \left\lfloor \frac{(d+4)^3+3(d+4)^2-9(d+4)\big((d+4) \bmod 2\big)}{144} \right\rceil \tag{3.12}$$

where $\lfloor x \rceil$ denotes the nearest integer to x. Comparison between equations (3.12) and (3.9) indicates that the fully symmetric case involves significantly fewer moment equations than the asymmetric case, with the ratio reducing to $1/24$ as $d \to \infty$.

The symmetric group G_{FS} generates five orbit types, classified by the number of repeated barycentric coordinates in the generator. These types, labeled 0 to 4, are detailed in Table 3.1, which also shows the number of points and variables introduced in the moment equations for each orbit type.

The total number of points n_K for a rule with orbit structure $[n_0, n_1, n_2, n_3, n_4]$ is:

$$n_K = n_0 + 4n_1 + 6n_2 + 12n_3 + 24n_4 \tag{3.13}$$

Table 3.1: Types of orbits for fully symmetry in tetrahedron

Orbit type	Generator	Condition	points	variables
0	$(1/4,\, 1/4,\, 1/4,\, 1/4)$	—	1	1
1	$(\alpha,\, \alpha,\, \alpha,\, 1-3\alpha)$	$\alpha \neq 1/4$	4	2
2	$(\alpha,\, \alpha,\, 1/2-\alpha,\, 1/2-\alpha)$	$\alpha \neq 1/4$	6	2
3	$(\alpha,\, \alpha,\, \beta,\, 1-2\alpha-\beta)$	$\alpha \neq \beta,\, 3\alpha+\beta \neq 1,\, \alpha+\beta \neq 1/2$	12	3
4	$(\alpha,\, \beta,\, \gamma,\, 1-\alpha-\beta-\gamma)$	all coordinates distinct	24	4

while the number of unknowns is

$$n_e = n_0 + 2n_1 + 2n_2 + 3n_3 + 4n_4 \tag{3.14}$$

Equations (3.12) and (3.14) help derive a consistency condition for fully symmetric rules. However, a more precise set of conditions can be achieved using an alternative basis that maximizes the number of zero elements across as many orbit types as possible.

3.2.3 A simpler fully symmetric basis

Substituting L for the generic variable v in equation (3.7), and using equation (3.11a), gives

$$L^4 - L^3 + \tilde{L}_2 L^2 + \tilde{L}_3 L + \tilde{L}_4 = 0. \tag{3.15}$$

Studying the multiplicity of the roots of (3.15), considered as a quartic function in L, gives the relation between $\tilde{L}_2$, $\tilde{L}_3$ and $\tilde{L}_4$ for each orbit type. This study is greatly simplified by considering the depressed quartic, therefore we use the transformation

$$l_\kappa = L_\kappa - 1/4 \qquad \text{with } \kappa = 1 \ldots 4 \tag{3.16}$$

so that the elementary symmetric polynomials are

$$\tilde{l}_1 = -(l_1 + l_2 + l_3 + l_4) = 0 \tag{3.17a}$$
$$\tilde{l}_2 = l_1 l_2 + l_1 l_3 + l_1 l_4 + l_2 l_3 + l_2 l_4 + l_3 l_4 \tag{3.17b}$$
$$\tilde{l}_3 = -(l_1 l_2 l_3 + l_1 l_3 l_4 + l_1 l_2 l_4 + l_2 l_3 l_4) \tag{3.17c}$$
$$\tilde{l}_4 = l_1 l_2 l_3 l_4 \tag{3.17d}$$

and equation (3.7) becomes

$$l^4 + \tilde{l}_2 l^2 + \tilde{l}_3 l + \tilde{l}_4 = 0. \tag{3.18}$$

The discriminant of (3.18) with respect to l is

$$\Delta = -27\tilde{l}_3^4 - 4\tilde{l}_2(\tilde{l}_2^2 - 36\tilde{l}_4)\tilde{l}_3^2 + 16\tilde{l}_4(\tilde{l}_2^2 - 4\tilde{l}_4)^2. \tag{3.19}$$

We therefore have the following cases (Rees, 1922):

- Type-0 orbit (four equal roots, all zero) for $\tilde{l}_2 = \tilde{l}_3 = \tilde{l}_4 = 0$
- Type-1 orbit (three equal roots) for $8\tilde{l}_2^3 + 27\tilde{l}_3^2 = 0$ and $\tilde{l}_2^2 + 12\tilde{l}_4 = 0$, with $\tilde{l}_2 \neq 0$
- Type-2 orbit (two pairs of equal roots) for $\tilde{l}_3 = 0$ and $\tilde{l}_2^2 - 4\tilde{l}_4 = 0$, with $\tilde{l}_2 \neq 0$
- Type-3 orbit (only one pair of equal roots) for $\Delta = 0$ but none of the previous cases holding
- Type-4 orbit (four distinct roots) for $\Delta \neq 0$.

Note that the conditions given for orbits of types 0, 1 and 2 ensure that $\Delta = 0$. Further simplification is achieved by introducing the quantities

$$p = -\frac{2\tilde{l}_2}{3}, q = -\tilde{l}_3, r = \frac{\tilde{l}_2^2 + 12\tilde{l}_4}{9} \tag{3.20}$$

resulting in the following simpler conditions

- Type-0 orbit for $p = q = r = 0$
- Type-1 orbit for $p^3 - q^2 = 0$ and $r = 0$, with $p \neq 0$
- Type-2 orbit for $q = 0$ and $p^2 - r = 0$, with $p \neq 0$
- Type-3 orbit for $\Delta = 0$ but none of the previous cases holding
- Type-4 orbit for $\Delta \neq 0$

where now

$$\Delta = 27\big(-q^4 + 2p(p^2 - 3r)q^2 - (p^2 - 4r)(p^2 - r)^2\big) \tag{3.21}$$

We therefore consider the fully symmetric monomial basis $p^i q^j r^k$ with weighted degree $2i + 3j + 4k \leq d$. This basis is simpler than the previous ones, as it is easier to express the conditions holding on orbits of types 0 to 3.

3.2.4 Fully symmetric basis for consistency conditions

Adopting a basis with many zero elements for various orbit types yields more precise consistency conditions. This can be achieved by starting with the monomial basis $p^i q^j r^k$ and then splitting and transforming these elements through linear combinations with constant coefficients. This process is summarized in Figure 3.2 and detailed in the text.

Orbits of types 0 to 3 are identified using the condition $\Delta = 0$. As shown in Equation (3.21), Δ is a polynomial of degree 3 in r. Consequently, we categorize the monomial $p^i q^j r^k$ by the degree of r into the following groups:

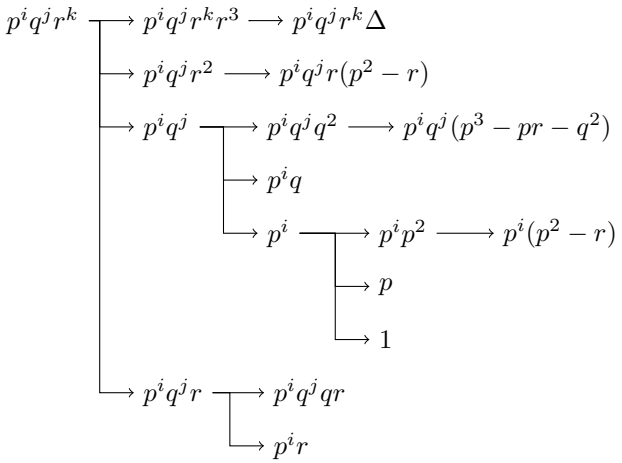


Figure 3.2: Diagram showing the derivation of the fully symmetric basis for consistency conditions

$$[p^iq^jr^k] \to [p^iq^jr^kr^3, p^iq^jr^2, p^iq^j, p^iq^jr] \tag{3.22}$$

Here, i,j,k can assume different values for each term. For simplicity, we assume $i,j,k \geq 0$ and that all monomials and polynomials are of a total weighted degree up to d. The swap of the last two terms in the sequence will be clarified subsequently.

The terms $p^iq^jr^kr^3$ can be transformed into $p^iq^jr^k\Delta$ by taking linear combinations with other terms, rendering them zero for orbit types 0 to 3. The remaining terms, which have a degree in r less than 3, cannot be linearly combined to include a Δ factor, and thus do not zero out for type-3 orbits.

To advance our analysis, we employ algebraic geometry methods (Cox, Little, & O'Shea, 2013) to establish that any polynomial in p,q,r that zeros out for orbit types 0 to 2 can be expressed as a sum of polynomials with factors p^3-pr-q^2, $r(p^2-r)$, or qr. These three polynomials are crucial as they generate the radical of the product of the ideals formed by the polynomials that must zero out for each orbit type from 0 to 2.

The terms $p^iq^jr^2$ in Equation (3.22), when linearly combined with the terms p^iq^jr, result in the expression $p^iq^jr(p^2-r)$. From the original split in Equation (3.22), this process separates the remaining terms p^iq^j and p^iq^jr, which are further divided as follows:

$$[p^iq^j, p^iq^jr] \to [p^iq^jq^2, p^iq, p^i, p^iq^jqr, p^ir] \tag{3.23}$$

Table 3.2: Fully symmetric basis for consistency conditions. The column "orbit types" lists the orbit types for which the elements are not necessarily zero.

elements	orbit types	weighted degree	number of elements	
$p^i q^j r^k \Delta$	4	$2i+3j+4k+12$	$m_{p3}(d-12)$	m_4
$p^i q^j r(p^2-r)$	3,4	$2i+3j+8$	$m_{p2}(d-8)$	$\}m_3$
$p^i q^j qr$	3,4	$2i+3j+7$	$m_{p2}(d-7)$	
$p^i q^j(p^3-pr-q^2)$	3,4	$2i+3j+6$	$m_{p2}(d-6)$	
$p^i(p^2-r)$	1,3,4	$2i+4$	$m_{p1}(d-4)$	$\}m_1$
$p^i q$	1,3,4	$2i+3$	$m_{p1}(d-3)$	
$p^i r$	2,3,4	$2i+4$	$m_{p1}(d-4)$	m_2
p	1,2,3,4	2	$m_{p0}(d-2)$	m_{12}
1	0,1,2,3,4	0	1	m_0

In Equation (3.23), combining the first term on the right-hand side with the second, fourth, and fifth terms results in $p^i q^j(p^3-pr-q^2)$, effectively utilizing these components. Meanwhile, the fourth term $p^i q^j(qr)$ already matches the required form, collectively covering all terms that zero out for orbit types 0 to 2.

The third term on the right-hand side of Equation (3.23), represented as $[p^i]$, is further subdivided into the following components:

$$[p^i] \to [p^i p^2, p, 1] \tag{3.24}$$

In this breakdown, combining the terms $p^i p^2$ with the terms $p^i r$ produces $p^i(p^2-r)$, which are zero for orbit types 0 and 2.

Compiling all the terms derived from the splits and combinations provides a new fully symmetric basis, detailed in Table 3.2, which moves beyond solely monomial terms. The method of developing this new basis from the monomial structure in p,q,r guarantees that it remains a basis for the same vector space of polynomials. Moreover, this basis optimizes the number of elements that are zero for various orbit types. This is because no linear combination of elements that are zero for fewer orbit types can yield a polynomial that is zero for more orbit types, thereby maximizing efficiency in addressing symmetries within the polynomial space.

3.3 Consistency conditions and (quasi-)optimal rules

In the previous section, we developed a new fully symmetric basis for the vector space of G_{FS}-invariant polynomials, optimized to maximize zero elements across many orbit types. This configuration aids in deriving consistency conditions and establishing estimated lower bounds on the number of integration points required for cubature rules in tetrahedra.

3.3.1 Number of basis elements equations

The last column in Table 3.2 lists the number of basis elements for each element type, representing the count of non-negative integer solutions to the indices in the weighted degree equation, where the weighted degree does not exceed the degree d. This enumeration reflects the scope of possible configurations within the constraints of the given polynomial degree. Specifically, $m_{p3}(d)$ is given by eq. (3.12), extended to also cover negative values of d

$$m_{p3}(d) = \left\lfloor \frac{(d+4)^3 + 3(d+4)^2 - 9(d+4)\big((d+4) \bmod 2\big)}{144} \right\rfloor [\![d \geq 0]\!] \tag{3.25a}$$

where the Iverson brackets $[\![\ldots]\!]$ are defined as Knuth (1992)

$$[\![S]\!] = \begin{cases} 0 & \text{if } S \text{ is false} \\ 1 & \text{if } S \text{ is true} \end{cases} \tag{3.25b}$$

Similarly, $m_{p2}(d)$ is the number of non-negative integer solutions of $2i+3j \leq d$, given by OEIS A001400 (2022a)

$$m_{p2}(d) = \left\lfloor \frac{(d+3)^2}{12} \right\rfloor [\![d \geq 0]\!] \tag{3.25c}$$

and $m_{p1}(d)$ is the number of non-negative integer solutions of $2i \leq d$, that is

$$m_{p1}(d) = \left\lfloor \frac{d+2}{2} \right\rfloor [\![d \geq 0]\!] \tag{3.25d}$$

where $\lfloor x \rfloor$ is the largest integer that is smaller or equal to x. Finally, $m_{p0}(d)$ is simply given by

$$m_{p0}(d) = [\![d \geq 0]\!] \tag{3.25e}$$

Table 3.2 organizes the basis elements into six groups based on the orbit types for which they may be non-zero. These groups are: $\{4\}, \{3,4\}, \{2,3,4\}, \{1,3,4\}, \{1,2,3,4\}$ and $\{0,1,2,3,4\}$ containing $m_4, m_3, m_2, m_1, m_{12}$ and m_0 elements, respectively. Using the information in Table 3.2 and the equations referenced as (3.25), the count of basis elements in each category is accurately determined by equations:

Table 3.3: Number of basis elements for fully symmetry in each orbit group

Degree	m_0	m_{12}	m_1	m_2	m_3	m_4	m_e
0	1	0	0	0	0	0	1
1	1	0	0	0	0	0	1
2	1	1	0	0	0	0	2
3	1	1	1	0	0	0	3
4	1	1	2	1	0	0	5
5	1	1	3	1	0	0	6
6	1	1	4	2	1	0	9
7	1	1	5	2	2	0	11
8	1	1	6	3	4	0	15
9	1	1	7	3	6	0	18
10	1	1	8	4	9	0	23
11	1	1	9	4	12	0	27
12	1	1	10	5	16	1	34
13	1	1	11	5	20	1	39
14	1	1	12	6	25	2	47
15	1	1	13	6	30	3	54
16	1	1	14	7	36	5	64
17	1	1	15	7	42	6	72
18	1	1	16	8	49	9	84
19	1	1	17	8	56	11	94
20	1	1	18	9	64	15	108

$$m_4 = m_{p3}(d-12) \tag{3.26a}$$

$$m_3 = \left\lfloor \left(\frac{d}{2}-2\right)^2 \right\rfloor [\![d \geq 6]\!] \tag{3.26b}$$

$$m_2 = \left\lfloor \frac{d}{2}-1 \right\rfloor [\![d \geq 4]\!] \tag{3.26c}$$

$$m_1 = (d-2)[\![d \geq 2]\!] \tag{3.26d}$$

$$m_{12} = [\![d \geq 2]\!] \tag{3.26e}$$

$$m_0 = 1 \tag{3.26f}$$

Table 3.3 lists the number of basis elements in each element group, and the total number of elements, for degrees $d \leq 20$.

3.3.2 Consistency conditions

Every basis element must fulfill Equation (2.28), making each element group a subsystem within the moment equations. For consistency, the number of unknowns for all orbits in any group must be at least equal to the number of equations exclusively involving those orbits. By applying this rule to each group, we derive the consistency conditions

$$n_0 + 2n_1 + 2n_2 + 3n_3 + 4n_4 \geq m_4 + m_3 + m_2 + m_1 + m_{12} + m_0 \tag{3.27a}$$
$$2n_1 + 2n_2 + 3n_3 + 4n_4 \geq m_4 + m_3 + m_2 + m_1 + m_{12} \tag{3.27b}$$
$$2n_1 \quad + 3n_3 + 4n_4 \geq m_4 + m_3 \quad + m_1 \tag{3.27c}$$
$$2n_2 + 3n_3 + 4n_4 \geq m_4 + m_3 + m_2 \tag{3.27d}$$
$$3n_3 + 4n_4 \geq m_4 + m_3 \tag{3.27e}$$
$$4n_4 \geq m_4 \tag{3.27f}$$

An additional condition is that there can be at most one orbit of type 0, i.e. $n_0 \in \{0,1\}$. This means, since $m_0 = 1$, that equation (3.27b) can be omitted as it is implied by equation (3.27a). The consistency conditions can then be written as

$$n_0 \in \{0,1\} \tag{3.28a}$$
$$n_0 + 2n_1 + 2n_2 + 3n_3 + 4n_4 \geq m_4 + m_3 + m_2 + m_1 + m_{12} + m_0 \tag{3.28b}$$
$$2n_1 \quad + 3n_3 + 4n_4 \geq m_4 + m_3 \quad + m_1 \tag{3.28c}$$
$$2n_2 + 3n_3 + 4n_4 \geq m_4 + m_3 + m_2 \tag{3.28d}$$
$$3n_3 + 4n_4 \geq m_4 + m_3 \tag{3.28e}$$
$$4n_4 \geq m_4 \tag{3.28f}$$

3.3.3 Consistent rule structures

For a given degree d, a rule structure $[n_0, n_1, n_2, n_3, n_4]$ is consistent if it satisfies the consistency conditions (3.28). Consistent rule structures with a given maximum number of points can be easily calculated using algorithm 1.

A rule structure is considered optimal for a given degree d if no other consistent structures for that degree exist with fewer points. Typically, identifying optimal consistent rule structures is framed as an integer linear programming problem (Mantel & Rabinowitz, 1977). However, we suggest a more straightforward method to tackle this optimization challenge. Considering the consistency conditions (3.28) in reverse order, we estimate an optimal consistent rule structure as

Algorithm 1 Pseudo code for consistent rule structures

$d \leftarrow$ cubature rule degree
$n_K^m \leftarrow$ desired maximum number of integration points
for $n_4 \leftarrow \left\lceil \frac{m_4}{4} \right\rceil$ to $\left\lfloor \frac{n_K^m}{24} \right\rfloor$ **do**
  for $n_3 \leftarrow \max\left\{ \left\lceil \frac{m_4+m_3-4n_4}{3} \right\rceil, 0 \right\}$ to $\left\lfloor \frac{n_K^m-24n_4}{12} \right\rfloor$ **do**
    for $n_2 \leftarrow \max\left\{ \left\lceil \frac{m_4+m_3+m_2-4n_4-3n_3}{2} \right\rceil, 0 \right\}$ to $\left\lfloor \frac{n_K^m-24n_4-12n_3}{6} \right\rfloor$ **do**
      for $n_1 \leftarrow \max\left\{ \left\lceil \frac{m_4+m_3+m_1-4n_4-3n_3}{2} \right\rceil, 0 \right\}$ to $\left\lfloor \frac{n_K^m-24n_4-12n_3-6n_2}{4} \right\rfloor$ **do**
        for $n_0 \leftarrow \max\{m_\mathrm{e}-4n_4-3n_3-2n_2-2n_1, 0\}$ to $\min\{n_K^m-24n_4-12n_3-6n_2-4n_1, 1\}$ **do**
          save $[n_0, n_1, n_2, n_3, n_4]$
        end for
      end for
    end for
  end for
end for

$$n_4^* = \left\lceil \frac{m_4}{4} \right\rceil \tag{3.29a}$$
$$n_3^* = \left\lceil \frac{m_4 + m_3 - 4n_4^*}{3} \right\rceil \tag{3.29b}$$
$$n_2^* = \left\lceil \frac{m_4 + m_3 + m_2 - 3n_3^* - 4n_4^*}{2} \right\rceil \tag{3.29c}$$
$$n_1^* = \left\lfloor \frac{m_\mathrm{e} - 2n_2^* - 3n_3^* - 4n_4^*}{2} \right\rfloor \tag{3.29d}$$
$$n_0^* = m_\mathrm{e} - 2n_1^* - 2n_2^* - 3n_3^* - 4n_4^* \tag{3.29e}$$

Algorithm 1, set with a maximum number of points equal to those of the rule structure outlined in Equation (3.29), demonstrates that this rule is not only optimal but also unique, at least for degrees up to an impractically high $d = 200$.

Table 3.4 shows the optimal consistent rule structures for degree $d \leq 20$, and the corresponding number of points. This number of points represents an estimate of the lower bound for the number of integration points.

It is crucial to note that consistency conditions merely offer an estimate of which cubature rule structures might result in actual cubature rules. Meeting these conditions does not ensure that the system of moment equations is consistent or that a cubature rule with a given structure definitively exists.

Table 3.4: Optimal consistent rule structures for tetrahedra

Degree	Number of points	n_0	n_1	n_2	n_3	n_4
0	1	1	0	0	0	0
1	1	1	0	0	0	0
2	4	0	1	0	0	0
3	5	1	1	0	0	0
4	11	1	1	1	0	0
5	14	0	2	1	0	0
6	24	0	3	0	1	0
7	30	0	3	1	1	0
8	43	1	3	1	2	0
9	52	0	4	2	2	0
10	68	0	5	2	3	0
11	81	1	5	2	4	0
12	117	1	5	2	5	1
13	133	1	6	2	6	1
14	163	1	6	3	8	1
15	190	0	7	3	10	1
16	233	1	7	4	11	2
17	266	0	8	3	14	2
18	318	0	9	3	16	3
19	355	1	9	3	19	3
20	415	1	9	5	21	4

Table 3.5: The definition of p, q and r in barycentric coordinates for type-1, -2 and -3 orbit

Type of orbits	p	q	r
type-1	$4l_1^2$	$-8l_1^3$	0
type-2	$4l_4^2/3$	0	$16l_4^4/9$
type-3	$4l_1^2+2(l_1+l_4)^2/3$	$-2l_1(l_1+l_4)^2$	$(4l_1^2-(l_1+l_4)^2)^2/9$

In practical applications, only cubature rules of quality PI, or possibly NI, are typically deemed acceptable. Thus, even if rules with an optimal consistent structure exist, their quality may not meet these standards. In such cases, it is worthwhile to explore quasi-optimal consistent rule structures, which adhere to the consistency conditions but may have slightly more integration points than the optimal ones. For a given degree, the search generally focuses on rules with fewer integration points than any known PI rule. These quasi-optimal structures can be easily computed using Algorithm 1.

3.4 New results for cubature rules

Consistency conditions effectively narrow the search space for higher-quality cubature rules or those with fewer points than currently known. Additionally, selecting an appropriate polynomial basis simplifies the computations required to solve the moment equations and formulate specific rules.

3.4.1 Moment equations simplification

The fully symmetric basis in Table 3.2 establishes explicit consistency conditions. However, further simplification is needed to effectively address the derived moment equations.

For orbits of type-1, -2, and -3, the definitions in (3.20) are applied, and the elementary symmetric polynomials are expressed in terms of barycentric coordinates using the formulas in (3.17). The definition of these orbits in Table 3.1 indicates that barycentric coordinates may repeat for orbits 1, 2, and 3, which allows for additional simplification of the expressions. The specific definitions of p, q and r in barycentric coordinates for these types of orbits are detailed in Table 3.5.

For example, using the definition (2.37), the moment equations derived from the fully symmetric basis in Table 3.2 for a degree 7 cubature rule with the structure $[1,1,1,2,0]$ (the optimal PI rule identified in (Jaśkowiec & Sukumar, 2020b)) are as follows:

$$I(1) = \sum_{k=1}^{1} w_{0,k} + 4\sum_{k=1}^{1} w_{1,k} + 6\sum_{k=1}^{1} w_{2,k} + 12\sum_{k=1}^{2} w_{3,k} \tag{3.30a}$$

$$I(p) = 4\sum_{k=1}^{1} w_{1,k}p_{1,k} + 6\sum_{k=1}^{1} w_{2,k}p_{2,k} + 12\sum_{k=1}^{2} w_{3,k}p_{3,k} \tag{3.30b}$$

$$I(q) = 4\sum_{k=1}^{1} w_{1,k}q_{1,k} + 12\sum_{k=1}^{2} w_{3,k}q_{3,k} \tag{3.30c}$$

$$I(r) = 6\sum_{k=1}^{1} w_{2,k}r_{2,k} + 12\sum_{k=1}^{2} w_{3,k}r_{3,k} \tag{3.30d}$$

$$I\left(p^2 - r\right) = 4\sum_{k=1}^{1} w_{1,k}\left(p_{1,k}^2 - r_{1,k}\right) + 12\sum_{k=1}^{2} w_{3,k}\left(p_{3,k}^2 - r_{3,k}\right) \tag{3.30e}$$

$$I(pq) = 4\sum_{k=1}^{1} w_{1,k}p_{1,k}q_{1,k} + 12\sum_{k=1}^{2} w_{3,k}p_{3,k}q_{3,k} \tag{3.30f}$$

$$I(pr) = 6\sum_{k=1}^{1} w_{2,k}p_{2,k}r_{2,k} + 12\sum_{k=1}^{2} w_{3,k}p_{3,k}r_{3,k} \tag{3.30g}$$

$$I\left(p^2q\right) = 4\sum_{k=1}^{1} w_{1,k}p_{1,k}^2q_{1,k} + 12\sum_{k=1}^{2} w_{3,k}p_{3,k}^2q_{3,k} \tag{3.30h}$$

$$I\left(p\left(p^2 - r\right)\right) = 4\sum_{k=1}^{1} w_{1,k}p_{1,k}\left(p_{1,k}^2 - r_{1,k}\right) + 12\sum_{k=1}^{2} w_{3,k}p_{3,k}(p_{3,k}^2 - r_{3,k}) \tag{3.30i}$$

$$I\left(p^3 - pr - q^2\right) = 12\sum_{k=1}^{2} w_{3,k}\left(p_{3,k}^3 - p_{3,k}r_{3,k} - q_{3,k}^2\right) \tag{3.30j}$$

$$I(qr) = 12\sum_{k=1}^{2} w_{3,k}q_{3,k}r_{3,k} \tag{3.30k}$$

where $w_{i,k}$, $p_{i,k}$, $q_{i,k}$ and $r_{i,k}$ are the values of w, p, q and r for kth type-i orbit.

Let $l_{i,j,k}$ be the ith barycentric coordinate for the kth type-j orbit. Equation (3.30) forms a positive-dimensional system due to the interdependence of p, q, and r. This redundancy can be eliminated by substituting the definitions from Table 3.5. To further simplify and solve the moment equations, new variables are introduced as follows:

$$a_k = -2l_{1,1,k} \tag{3.31a}$$

$$b_k = \frac{4l_{4,2,k}^2}{3} \tag{3.31b}$$

$$c_k = -2l_{1,3,k} \tag{3.31c}$$

$$d_k = (l_{1,3,k} + l_{4,3,k})^2 \tag{3.31d}$$

The variables for type-1, -2, and -3 orbits are significantly simplified, as shown in Table 3.6, reducing the number of variables and making the moment equations easier to solve.

The moment equations (3.30) then can be written as

Table 3.6: Substitution of $p_{i,k}$, $q_{i,k}$ and $r_{i,k}$ with a_k, b_k, c_k and d_k for type-1, -2 and -3 orbit

Type-i orbits	$p_{i,k}$	$q_{i,k}$	$r_{i,k}$
type-1	a_k^2	a_k^3	0
type-2	b_k	0	b_k^2
type-3	$(c_k^2+2d_k)/3$	$c_k d_k$	$(c_k^2-d_k)^2/9$

$$I(1) = \sum_{k=1}^{1} w_{0,k} + 4\sum_{k=1}^{1} w_{1,k} + 6\sum_{k=1}^{1} w_{2,k} + 12\sum_{k=1}^{2} w_{3,k} \tag{3.32a}$$

$$I(p) = 4\sum_{k=1}^{1} w_{1,k}a_k^2 + 6\sum_{k=1}^{1} w_{2,k}b_k + 12\sum_{k=1}^{2} w_{3,k}\frac{\left(c_k^2+2d_k\right)}{3} \tag{3.32b}$$

$$I(q) = 4\sum_{k=1}^{1} w_{1,k}a_k^3 + 12\sum_{k=1}^{2} w_{3,k}c_k d_k \tag{3.32c}$$

$$I(r) = 6\sum_{k=1}^{1} w_{2,k}b_k^2 + 12\sum_{k=1}^{2} w_{3,k}\frac{\left(c_k^2-d_k\right)^2}{9} \tag{3.32d}$$

$$I\left(p^2-r\right) = 4\sum_{k=1}^{1} w_{1,k}a_k^4 + 12\sum_{k=1}^{2} w_{3,k}\left(\frac{\left(c_k^2+2d_k\right)^2}{3} - \frac{\left(c_k^2-d_k\right)^2}{9}\right) \tag{3.32e}$$

$$I(pq) = 4\sum_{k=1}^{1} w_{1,k}a_k^5 + 12\sum_{k=1}^{2} w_{3,k}\frac{\left(c_k^2+2d_k\right)}{3}c_k d_k \tag{3.32f}$$

$$I(pr) = 6\sum_{k=1}^{1} w_{2,k}b_k^3 + 12\sum_{k=1}^{2} w_{3,k}\frac{\left(c_k^2+2d_k\right)}{3}\frac{\left(c_k^2-d_k\right)^2}{9} \tag{3.32g}$$

$$I\left(p^2q\right) = 4\sum_{k=1}^{1} w_{1,k}a_k^7 + 12\sum_{k=1}^{2} w_{3,k}\frac{\left(c_k^2+2d_k\right)^2}{3}c_k d_k \tag{3.32h}$$

$$I\left(p\left(p^2-r\right)\right) = 4\sum_{k=1}^{1} w_{1,k}a_k^6 + 12\sum_{k=1}^{2} w_{3,k}\frac{\left(c_k^2+2d_k\right)}{3}\left(\frac{\left(c_k^2+2d_k\right)^2}{3} - \frac{\left(c_k^2-d_k\right)^2}{9}\right) \tag{3.32i}$$

$$I\left(p^3-pr-q^2\right) = 12\sum_{k=1}^{2} w_{3,k}\left(\frac{\left(c_k^2+2d_k\right)^3}{3} - \frac{\left(c_k^2+2d_k\right)}{3}\frac{\left(c_k^2-d_k\right)^2}{9} - c_k d_k^2\right) \tag{3.32j}$$

$$I(qr) = 12\sum_{k=1}^{2} w_{3,k}c_k d_k\frac{\left(c_k^2-d_k\right)^2}{9} \tag{3.32k}$$

Because integral is a linear functional, the left-hand side of equations (3.32) are the linear combination of integrals

$$I_{i,j,k} = \frac{1}{V}\int_{\Omega} p^i q^j r^k d\Omega \tag{3.33}$$

whose integrand $p^i q^j r^k$ can be expanded into a polynomial in the barycentric coordinates L_1, L_2, L_3 and L_4. Using the formula

$$\hat{I}_{i,j,k,l} = \frac{1}{V}\int_{\Omega} L_1^i L_2^j L_3^k L_4^l d\Omega = \frac{6i!\,j!\,k!\,l!}{(i+j+k+l+3)!}, \tag{3.34}$$

we can integrate separately each term of the polynomial and add the results which leads to

$$I_{0,0,0} = 1, \quad I_{1,0,0} = \frac{1}{20}, \quad I_{0,1,0} = \frac{1}{120}, \quad I_{2,0,0} = \frac{19}{5040}, \quad I_{0,0,1} = \frac{1}{1260}, \quad I_{1,1,0} = \frac{1}{1120}, \quad .... \tag{3.35}$$

and the left-hand side of equations (3.32) are

$$\begin{aligned} &I(1) = 1, \quad I(p) = \frac{1}{20}, \quad I(q) = \frac{1}{120}, \quad I(r) = \frac{1}{1260}, \quad I\left(p^2 - r\right) = \frac{1}{336}, \quad I(pq) = \frac{1}{1120}, \\ &I(pr) = \frac{1}{19440}, \quad I\left(p^2 q\right) = \frac{67}{604800}, \quad I\left(p\left(p^2 - r\right)\right) = \frac{59}{181440}, \quad I\left(p^3 - pr - q^2\right) = \frac{1}{22680}, \\ &I(qr) = \frac{1}{151200} \end{aligned} \tag{3.36}$$

3.4.2 Solution strategy

The moment equations (3.32) can be further simplified by applying the technique outlined in (Rabinowitz & Richter, 1969). According to Table 3.2, the variables $w_{1,k}$ and a_k for type-1 orbits are included solely in the equations $Q\left(p^i\left(p^2 - r\right)\right)$, $Q\left(p^i q\right)$, $Q(p)$ and $Q(1)$. Similarly, the variables $w_{2,k}$ and b_k for type-2 orbits appear only in the equations $Q\left(p^i r\right)$, $Q(p)$ and $Q(1)$. This focused inclusion simplifies the computational process by reducing the interaction between variables across different types of equations.

When dealing with cubature rules that include a type-0 orbit ($n_0 = 1$), the equation $Q(1)$ is primarily utilized to compute the weight $w_{0,1}$ assuming other weights are known. The weight $w_{1,k}$ can be eliminated from $Q\left(p^i\left(p^2 - r\right)\right)$, $Q\left(p^i q\right)$, $Q(p)$ and $Q(1)$ by the method described in (Papanicolopulos, 2015) which returns a system of equations:

$$\sum_{k=0}^{n_1} J_{i-k}\tilde{a}_k = 0 \quad i = n_1 + 2..d \tag{3.37}$$

where J_i is a new variable which is defined as

$$J_i = 4\sum_{k=1}^{n_1} w_{1,k} a_k^i \quad i = 2..d \tag{3.38}$$

and $\tilde{a}_m$ are elementary symmetric polynomials

$$\begin{aligned}
\tilde{a}_0 =& 1 \\
\tilde{a}_1 =& a_1 + a_2 + ... + a_{n_1} \\
\tilde{a}_2 =& a_1a_2 + a_1a_3 + ... + a_{n_1-1}a_{n_1} \\
& ... \\
\tilde{a}_{n_1} =& a_1a_2a_3...a_{n_1}
\end{aligned} \tag{3.39}$$

Similarly, the weight $w_{2,k}$ can be eliminated from $Q\left(p^i r\right)$, $Q(p)$ and $Q(1)$. The system of equations returned is

$$\sum_{k=0}^{n_2} K_{i-k}\tilde{b}_k = 0 \quad i = n_2 + 1..\lfloor\frac{d}{2}\rfloor \tag{3.40}$$

where

$$K_i = 6\sum_{k=1}^{n_2} w_{2,k}b_k^i \quad i = 1..\lfloor\frac{d}{2}\rfloor \tag{3.41}$$

and $\tilde{b}_k$ are symmetric polynomials

$$\begin{aligned}
\tilde{b}_0 =& 1 \\
\tilde{b}_1 =& b_1 + b_2 + ... + b_{n_2} \\
\tilde{b}_2 =& b_1b_2 + b_1b_3 + ... + b_{n_2-1}b_{n_2} \\
& ... \\
\tilde{b}_{n_2} =& b_1b_2b_3...b_{n_2}
\end{aligned} \tag{3.42}$$

Moment equations (3.32) then can be further simplified as

$$1 = \sum_{k=1}^{1} w_{0,k} + 4\sum_{k=1}^{1} w_{1,k} + 6\sum_{k=1}^{1} w_{2,k} + 12\sum_{k=1}^{2} w_{3,k} \tag{3.43a}$$

$$\frac{1}{20} = J_2 + K_1 + 12\sum_{k=1}^{2} w_{3,k}\frac{\left(c_k^2 + 2d_k\right)}{3} \tag{3.43b}$$

$$\frac{1}{120} = J_3 + 12\sum_{k=1}^{2} w_{3,k}c_k d_k \tag{3.43c}$$

$$\frac{1}{1260} = K_2 + 12\sum_{k=1}^{2} w_{3,k}\frac{\left(c_k^2 - d_k\right)^2}{9} \tag{3.43d}$$

$$\frac{1}{336} = J_4 + 12\sum_{k=1}^{2} w_{3,k}\left(\frac{\left(c_k^2+2d_k\right)^2}{3} - \frac{\left(c_k^2-d_k\right)^2}{9}\right) \tag{3.43e}$$

$$\frac{1}{1120} = J_5 + 12\sum_{k=1}^{2} w_{3,k}\frac{\left(c_k^2+2d_k\right)}{3}c_k d_k \tag{3.43f}$$

$$\frac{1}{19440} = K_3 + 12\sum_{k=1}^{2} w_{3,k}\frac{\left(c_k^2+2d_k\right)}{3}\frac{\left(c_k^2-d_k\right)^2}{9} \tag{3.43g}$$

$$\frac{67}{604800} = J_7 + 12\sum_{k=1}^{2} w_{3,k}\frac{\left(c_k^2+2d_k\right)^2}{3}c_k d_k \tag{3.43h}$$

$$\frac{59}{181440} = J_6 + 12\sum_{k=1}^{2} w_{3,k}\frac{\left(c_k^2+2d_k\right)}{3}\left(\frac{\left(c_k^2+2d_k\right)^2}{3} - \frac{\left(c_k^2-d_k\right)^2}{9}\right) \tag{3.43i}$$

$$\frac{1}{22680} = 12\sum_{k=1}^{2} w_{3,k}\left(\frac{\left(c_k^2+2d_k\right)^3}{3} - \frac{\left(c_k^2+2d_k\right)}{3}\frac{\left(c_k^2-d_k\right)^2}{9} - c_k d_k^2\right) \tag{3.43j}$$

$$\frac{1}{151200} = 12\sum_{k=1}^{2} w_{3,k}c_k d_k\frac{\left(c_k^2-d_k\right)^2}{9} \tag{3.43k}$$

(3.43l)

System generated by Equations (3.43b) to (3.43l) is positive-dimensional because variables J_i and K_i are not independent themselves. They must satisfy the following constraints:

$$0 = J_2\tilde{a}_1 + J_3 \tag{3.44a}$$

$$0 = J_3\tilde{a}_1 + J_4 \tag{3.44b}$$

$$0 = J_4\tilde{a}_1 + J_5 \tag{3.44c}$$

$$0 = J_5\tilde{a}_1 + J_6 \tag{3.44d}$$

$$0 = J_6\tilde{a}_1 + J_7 \tag{3.44e}$$

$$0 = K_1\tilde{b}_1 + K_2 \tag{3.44f}$$

$$0 = K_2\tilde{b}_1 + K_3 \tag{3.44g}$$

After eliminating the weights $w_{1,k}$ and $w_{2,k}$, the system, governed by Equations (3.43) and (3.44), has 18 unknowns, matching the number of equations. This elimination process adds more equations but introduces invariance among orbits of the same type, reducing the number of algebraic solutions and simplifying the system.

Considering the case where the cubature rules have no type-0 orbit ($n_0 = 0$), it would be easier for calculation by introducing an additional equation

$$J_1 = \sum_{k=1}^{n_1} w_{1,k} a_k \tag{3.45}$$

where J_1 is a new variable defined similar to (3.38). Eliminating the weights $w_{1,k}$ from equations $Q\left(p'^i\left(p'^2 - r'\right)\right)$, $Q\left(p'^i q'\right)$, $Q(p')$, $Q(1)$ and (3.45) will result in a system of equations like (3.37), except that the index i in (3.37) is now ranging from $n_1 + 2$ to d instead of n_1 to d.

Cubature rules exhibit symmetry under the permutation of orbits of the same type, as reflected in the moment equations (3.32). These equations are invariant under the permutation of the pairs $(w_{1,k},\, a_k)$, $(w_{2,k},\, b_k)$, and the triplets $(w_{3,k},\, c_k,\, d_k)$. This symmetry is also present in the system (3.37). When $w_{1,k}$ is eliminated, the resulting polynomials, represented by elementary symmetric polynomials $\tilde{a}_k$, remain symmetric in a_k. Similarly, eliminating $w_{2,k}$ allows the moment equations to be expressed in terms of the symmetric polynomials $\tilde{b}_k$.

Eliminating $w_{1,k}$ and $w_{2,k}$ from the moment equations (3.32) and expressing the results in terms of $\tilde{a}_k$ and $\tilde{b}_k$ results in a longer system. However, this transformation simplifies the solution when one is found.

3.4.3 Summary of new results

Table 3.7 displays the estimated lowest number of integration points and the actual lowest number achieved in known cubature rules of varying quality, for degrees up to 20. For NI or *O (i.e., PO or NO) quality cubature rules, and PI quality rules up to degree 6, existing results are detailed in (Cools, 2003). More recent findings, all of PI quality, include results from (L. Zhang et al., 2009) for degree 8, (Witherden & Vincent, 2015) for degrees 7 and 9, and (Chuluunbaatar et al., 2022) for degrees 12 to 20. Additionally, results for degrees 10 and 11 are sourced from version 0.9.7 of the PHG (Parallel Hierarchical Grid) software, available at `http://lsec.cc.ac.cn/phg/download.htm`, and were derived using an enhanced version of the methodology described in (L. Zhang et al., 2009).[1]

For degrees 1 to 6 and 8, the literature already provides rules of PI or NI quality that meet the estimated lowest number of points. For other degrees, there is an increasing gap between the estimated and actual lowest number of points as the degree increases. By applying the methods discussed in previous sections and extending the approach from (Papanicolopulos, 2015), we have obtained some new results, which are underlined in Table 3.7.

For degrees 7 and 9, we have derived rules that achieve the optimal number of points; however, these feature complex point coordinates. Although these results may not be practically useful, they validate the accuracy of the consistency predictions made for these rule structures. Given that the moment equations for both cases are zero-dimensional and all

1. L. Zhang, personal communication, 14 Sep. 2022.

Table 3.7: Estimated and known lower bounds for number of integration points in fully symmetric rules on the tetrahedron. Rules of worse quality are not shown if any rules of better quality with the same or lower number of integration points are known. Underlined results are newly obtained in this work.

degree	optimal	PI	NI	*O	*C
1	1	1			
2	4	4			
3	5	8	5		
4	11	14	11		
5	14	14			
6	24	24			
7	30	35	31	–	30
8	43	46	43		
9	52	59	55	53	52
10	71	79			
11	86	96	–	87	
12	117	123			
13	133	145			
14	163	175			
15	190	209			
16	233	248			
17	266	284			
18	318	343			
19	355	383			
20	415	441			

Table 3.8: Degree 9, 55-point NI rule generators and weights. The barycentric coordinates of the generators in terms of α and β are given in table 3.1.

Orbit type	weight	α	β
0	-4.6296861376723131		
1	1.2150353004018342	0.23962566193927949	
1	-0.22747436971238236	0.11018941963473842	
1	0.012202966891188984	0.049553558692414900	
2	0.016884869064330526	0.45159058017363670	
3	0.021517480296540043	0.39673341203779513	0.18159662632542559
3	0.10182368310953738	0.13432467380123244	0.65894809355477895
3	0.0041022810075698088	0.011921405727783134	0.71336869297190663

solutions have been computed using Gröbner bases, we confirm that there are no optimal rules of higher quality available. Additionally, for degree 9, we have developed a new 55-point NI rule with the structure $[1,3,1,3,0]$, improving upon the existing 59-point PI rule, as detailed in Table 3.8.

For degree 10, the optimal consistent rule structure, as detailed in Table 3.4, is $[0,5,2,3,0]$, which accounts for 68 points. The corresponding moment equations, derived using the non-monomial basis from Section 3.2.4, include a subsystem for type-3 orbits with 9 equations and 9 unknowns, which turns out to be inconsistent. Consequently, there are no degree-10 rules with the structure $[*,*,*,3,0]$; excluding these from consideration leads to identifying an optimal consistent structure of $[1,4,1,4,0]$ with 71 points. Similar analysis for degree 11 reveals that no rule with the structure $[*,*,*,4,0]$ exists, thereby establishing the optimal consistent structure as $[0,5,1,5,0]$ with 86 points.

3.4.4 Details on the consistency conditions for degree 10

For a degree-10 rule with structure $[*,*,*,3,0]$, table 3.2 shows that there are 9 basis elements that only involve type-3 orbits, namely

$$\begin{aligned}[g_1,\ldots,g_9] = [&p^3-pr-q^2, qr, p\left(p^3-pr-q^2\right), r\left(p^2-r\right), q\left(p^3-pr-q^2\right),\\ &pqr, p^2\left(p^3-pr-q^2\right), q^2r, pr\left(p^2-r\right)]\end{aligned} \tag{3.46}$$

Therefore the corresponding subsystem of the moment equations becomes

$$Q_i = I_i \qquad i = 1\ldots 9 \tag{3.47}$$

where

$$Q_i = 12 \sum_{j=1}^{3} w_j g_i(p_j, q_j, r_j) \tag{3.48}$$

and I_i are the exact integrals (2.4) evaluated for the elements g_i, resulting in

$$\begin{aligned}[I_1, \ldots, I_9] = [&1/22680, 1/151200, 19/4989600, 23/9979200, 1/1108800, \\ &13/19958400, 1/2620800, 29/172972800, 17/74131200]\end{aligned} \tag{3.49}$$

The quantities p_j, q_j, r_j for every orbit satisfy $\Delta_j = \Delta(p_j, q_j, r_j) = 0$, where the discriminant Δ is given by equation (3.21). Therefore the system (3.47) is essentially a system of 9 equations with nine unknowns, which we therefore assume to be consistent.

Despite assumptions, computing a Gröbner basis for the system reveals it to be inconsistent, indicating a failure of the consistency conditions to accurately predict the behavior of the moment equations in this instance. This inconsistency is not attributed to the choice of non-monomial basis described in Section 3.2.4. Rather, it stems from a more intricate relationship among the quantities Q_i, highlighting that in this case, the Q_i are not independent but are interconnected by specific equation

$$\begin{aligned}
&Q_1Q_3Q_7^2 + Q_1Q_3Q_7Q_8 - 5Q_1Q_3Q_7Q_9 + 4Q_1Q_3Q_9^2 - 4Q_1Q_4Q_7^2 - 4Q_1Q_4Q_7Q_8 \\
&\quad + 20Q_1Q_4Q_7Q_9 - 16Q_1Q_4Q_9^2 - Q_1Q_5^2Q_7 + 8Q_1Q_5Q_6Q_7 - 8Q_1Q_5Q_6Q_9 \\
&\quad - 12Q_1Q_6^2Q_7 + 4Q_1Q_6^2Q_8 + 12Q_1Q_6^2Q_9 + 4Q_2^2Q_7^2 + 4Q_2^2Q_7Q_8 - 20Q_2^2Q_7Q_9 \\
&\quad + 16Q_2^2Q_9^2 + 8Q_2Q_3Q_5Q_9 - 8Q_2Q_3Q_6Q_7 - 8Q_2Q_3Q_6Q_8 + 8Q_2Q_3Q_6Q_9 \\
&\quad - 8Q_2Q_4Q_5Q_7 + 32Q_2Q_4Q_6Q_7 - 32Q_2Q_4Q_6Q_9 - Q_3^3Q_7 - Q_3^3Q_8 + 5Q_3^3Q_9 \\
&\quad + 4Q_3^2Q_4Q_7 + 4Q_3^2Q_4Q_8 - 28Q_3^2Q_4Q_9 + Q_3^2Q_5^2 - 8Q_3^2Q_5Q_6 + 16Q_3^2Q_6^2 \\
&+ 4Q_3Q_4^2Q_7 + 32Q_3Q_4^2Q_9 + 8Q_3Q_4Q_5Q_6 - 32Q_3Q_4Q_6^2 - 16Q_4^3Q_7 + 16Q_4^2Q_6^2 = 0
\end{aligned} \tag{3.50}$$

Equation (3.50) is non-linear, and does not hold for $n_3 > 3$. It is therefore clear that deriving consistency conditions that correctly indicate that there are no degree-10 rules with structure $[*, *, *, 3, 0]$ requires a different approach to the one in this paper (and in general in the literature) which is based on linearly independent basis elements.

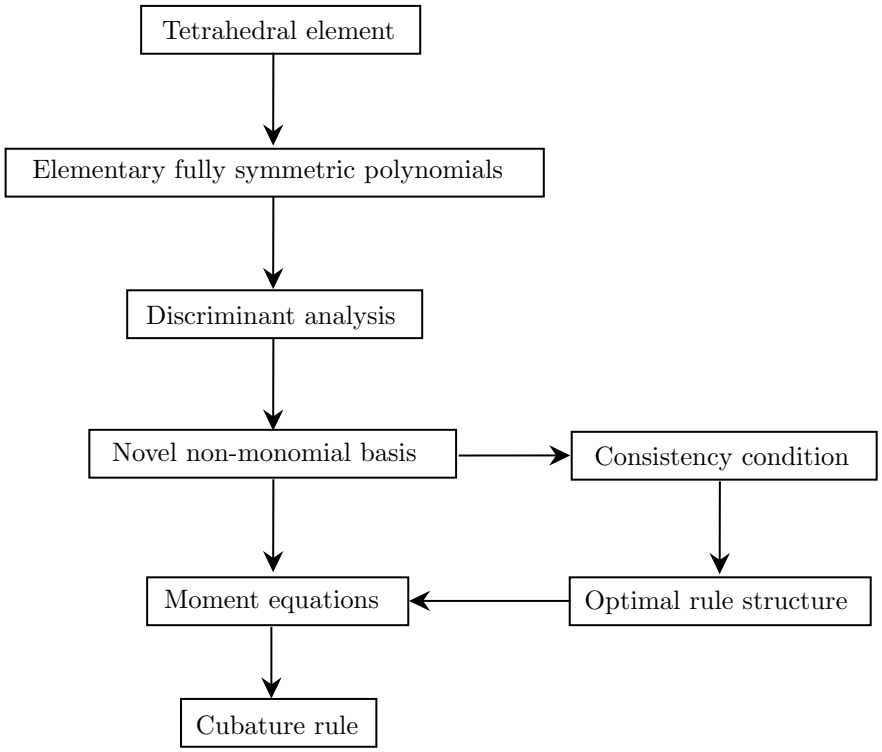


Figure 3.3: Flowchart to construct optimal fully symmetric cubature rules for tetrahedra

3.5 Summary

In this chapter, we have rigorously developed a new non-monomial fully symmetric polynomial basis for the tetrahedron, designed to maximize the number of zero elements across various orbit types. This innovative basis facilitates the direct formulation of consistency conditions. For the first time, we have derived explicit formulas for these conditions, enabling us to estimate the optimal consistent rule structures. Figure 3.3 illustrates the flowchart for constructing optimal fully symmetric cubature rules for tetrahedra using the newly derived consistency conditions. Additionally, we introduce an algorithm designed to generate optimal and quasi-optimal rule structures.

The newly developed non-monomial fully symmetric polynomial basis improves the calculation of specific cubature rules by decomposing the moment equations into smaller, independent subsystems. This decomposition enables more efficient solution methods. Using this approach, we derived a new NI cubature rule of degree 9 with 55 points, surpassing existing PI/NI quality rules. Additionally, we identified optimal rule structures for degrees 7 and 9, though these are not practically applicable due to their complex point coordinates.

Additionally, we demonstrated that the optimal rule structures predicted by our formulas for degrees 10 and 11 result in inconsistent moment equations. This inconsistency is not due to errors in the consistency conditions but arises from a intricate non-linear relationship among the moment equations that traditional assumptions fail to capture.

The quasi-optimal rule structures identified are valuable starting points for deriving additional cubature rules. For higher degrees with many equations and unknowns, advanced solving techniques, similar to those used for triangles, may be necessary. The methodologies in this chapter provide a foundation for extending these approaches to derive consistency conditions for various domains and symmetries.

In summary, while the novel basis derived from analyzing the quartic function's root distribution offers significant advantages in eliminating redundant orbits and formulating consistency conditions, it also introduces complexity due to the interdependence of the basis variables p, q, and r. This redundancy complicates solving the moment equations, necessitating further simplification and sometimes preventing a solution altogether.

Chapter 4

Other Invariance in Cubature Rules

4.1 Rotational symmetry in cubature rule

Rotationally symmetric cubature rules can be derived similarly to fully symmetric rules, as described in the previous chapter, but require a rotationally symmetric basis. This basis, unchanged under rotations around a central point or axis, ensures the configuration and properties of the basis functions or cubature points remain consistent, as described in Definition 10.

A rotationally symmetric basis, like a fully symmetric basis, reduces the number of unique points needed for precise numerical integration or function approximation in rotationally symmetric domains or functions. Figure 4.1 illustrates examples of rotationally symmetric and fully symmetric orbits.

4.1.1 Rotationally symmetric basis

The explicit consistency conditions detailed in Chapter 3 facilitate the search for new fully symmetric cubature rules for tetrahedra. For illustrative purposes, consider a type-4 orbit in such a cubature rule. This orbit includes a point defined by volume coordinates (L_1, L_2, L_3, L_4) and the 23 additional points generated the transformations in G_{FS}. Using Equation (2.36) for a polynomial $\hat{f}(L_1, L_2, L_3, L_4)$ in volume coordinates which yields only sums of the form

$$T_s = \sum_{G_{FS}(i_1,i_2,i_3,i_4)} \hat{f}(L_{i_1}, L_{i_2}, L_{i_3}, L_{i_4}). \tag{4.1}$$

T_s remains invariant under any permutations of barycentric coordinates. Using Definition 10, we introduce a type-5 orbit specific to rotational symmetry, which includes a generator with four distinct variables and comprises 12 points. Table 4.1 lists all possible orbits for rotationally symmetric cubature rules. Consequently, the sum specified in (4.1) for rotational symmetry is adjusted accordingly as

$$T_r = \sum_{G_{RS}(i_1,i_2,i_3,i_4)} \hat{f}(L_{i_1}, L_{i_2}, L_{i_3}, L_{i_4}) \tag{4.2}$$

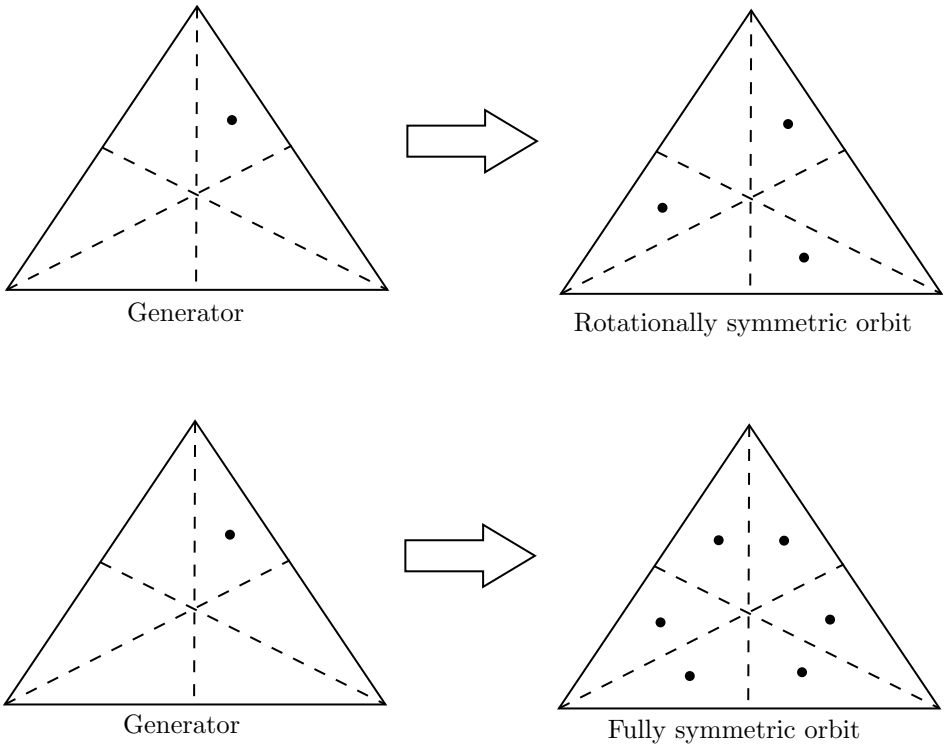


Figure 4.1: G_{RS}-invariant orbit and G_{FS}-invariant orbit for triangle

Table 4.1: Types of orbits for rotational symmetry in tetrahedron

Orbit type	Generator	Condition	points	variables
0	$(1/4,\, 1/4,\, 1/4,\, 1/4)$	—	1	1
1	$(\alpha,\, \alpha,\, \alpha,\, 1-3\alpha)$	$\alpha \neq 1/4$	4	2
2	$(\alpha,\, \alpha,\, 1/2-\alpha,\, 1/2-\alpha)$	$\alpha \neq 1/4$	6	2
3	$(\alpha,\, \alpha,\, \beta,\, 1-2\alpha-\beta)$	$\alpha \neq \beta,\, 3\alpha+\beta \neq 1,\, \alpha+\beta \neq 1/2$	12	3
4	$(\alpha,\, \beta,\, \gamma,\, 1-\alpha-\beta-\gamma)$	all coordinates distinct	24	4
5	$(\alpha,\, \beta,\, \gamma,\, 1-\alpha-\beta-\gamma)$	all coordinates distinct	12	4

which means the system polynomial equations now contain only the polynomials in barycentric coordinates in the form of T_r. Thus, we first need to find a rotationally symmetric basis for the vector space of polynomials in the form of T_r.

Compare Equation (4.1) and (4.2) and use the method mentioned in (Papanicolopulos, 2016b) we can rewrite (4.2) as

$$T_r = \frac{T_s + T_a}{2} \tag{4.3}$$

where T_a is an anti-symmetric polynomial

$$T_a = 2 \sum_{G_{RS}(i_1,i_2,i_3,i_4)} \hat{f}(L_{i_1}, L_{i_2}, L_{i_3}, L_{i_4}) - \sum_{G_{FS}(i_1,i_2,i_3,i_4)} \hat{f}(L_{i_1}, L_{i_2}, L_{i_3}, L_{i_4}) \tag{4.4}$$

which can be expressed in the product of symmetric polynomials and Vandermonde polynomial. Thus, according to Equation (4.3), T_r can be expressed as a polynomial in symmetric polynomials and Vandermonde polynomial.

A general Vandermonde polynomial is defined as

$$V_n = \sum_{1<i<j<n} (X_i - X_j) \tag{4.5}$$

where X_i are variables.

In chapter 3, we have mentioned that in the case of tetrahedra, a symmetric polynomial can be expressed in elementary symmetric polynomial $\tilde{l}_2$, $\tilde{l}_3$ and $\tilde{l}_4$ and the commonly used monomial basis consists of

$$\tilde{l}_2^i \tilde{l}_3^j \tilde{l}_4^k \quad 2i + 3j + 4k \leq d \tag{4.6}$$

In this case, consider the four barycentric coordinates for tetrahedra, Vandermonde polynomial has the form

$$v = V_4 = (L_1 - L_2)(L_1 - L_3)(L_1 - L_4)(L_2 - L_3)(L_2 - L_4)(L_3 - L_4) \tag{4.7}$$

which is a polynomial in barycentric coordinates with degree 6. The rotationally symmetric basis in the form of monomial can therefore be expressed as

$$\tilde{l}_2^i \tilde{l}_3^j \tilde{l}_4^k v^s \quad \text{with } 2i + 3j + 4k + 6s \leq d \text{ and } s \in \{0, 1\} \tag{4.8}$$

Table 4.2: Rotationally symmetric basis. The column "orbit types" lists the orbit types for which the elements are not necessarily zero.

elements	orbit types	weighted degree	number of elements	
$p^i q^j r^k \Delta$	4,5	$2i+3j+4k+12$	$m_{p3}(d-12)$	m_4
$p^i q^j r\left(p^2-r\right)$	3,4,5	$2i+3j+8$	$m_{p2}(d-8)$	$\}m_3$
$p^i q^j qr$	3,4,5	$2i+3j+7$	$m_{p2}(d-7)$	
$p^i q^j\left(p^3-pr-q^2\right)$	3,4,5	$2i+3j+6$	$m_{p2}(d-6)$	
$p^i q^j r^k v$	5	$2i+3j+4k+6$	$m_{p3}(d-6)$	m_5
$p^i\left(p^2-r\right)$	1,3,4,5	$2i+4$	$m_{p1}(d-4)$	$\}m_1$
$p^i q$	1,3,4,5	$2i+3$	$m_{p1}(d-3)$	
$p^i r$	2,3,4,5	$2i+4$	$m_{p1}(d-4)$	m_2
p	1,2,3,4,5	2	$m_{p0}(d-2)$	m_{12}
1	0,1,2,3,4,5	0	1	m_0

The reason $s \in \{0,1\}$ is that v relates to the discriminant (3.19) in the form

$$v^2 = \Delta \tag{4.9}$$

where Δ is a polynomial in elementary symmetric polynomials $\tilde{l}_i$, so if $s \geq 2$, monomials $\tilde{l}_2^i \tilde{l}_3^j \tilde{l}_4^k v^s$ becomes

$$\tilde{l}_2^{i'} \tilde{l}_3^{j'} \tilde{l}_4^{k'} v^t \quad \text{with } 2i'+3j'+4k'+6t \leq d \text{ and } t = s \mod 2. \tag{4.10}$$

which is identical to (4.8). Similarly, using the quantities p, q and r defined in (3.20) of Section 3.2.3, the refined rotationally symmetric basis is outlined in Table 4.2.

4.1.2 Moment equations for rotationally symmetric cubature rules

Comparing Table 3.2 and Table 4.2 shows that the rotationally symmetric basis is more extensive than the fully symmetric basis. All fully symmetric polynomials are rotationally symmetric, but not all rotationally symmetric polynomials are fully symmetric. Thus, the vector space of rotationally symmetric polynomials is broader. Consequently, constructing a rotationally symmetric cubature rule could involve more moment equations than a fully symmetric rule.

Table 4.3 details the number of basis elements for each orbit group under rotational symmetry, with m_5 representing the group exclusive to type-5 orbits. Since V_4 is a degree-6 polynomial non-zero only for type-5 orbits, the number of equations m_5 for these orbits can be calculated by

$$m_5 = m_{p3}(d-6). \tag{4.11}$$

Table 4.3: Number of basis elements for rotational symmetry in each orbit group

Degree	m_0	m_{12}	m_1	m_2	m_3	m_4	m_5	m_e
0	1	0	0	0	0	0	0	1
1	1	0	0	0	0	0	0	1
2	1	1	0	0	0	0	0	2
3	1	1	1	0	0	0	0	3
4	1	1	2	1	0	0	0	5
5	1	1	3	1	0	0	0	6
6	1	1	4	2	1	0	1	10
7	1	1	5	2	2	0	1	12
8	1	1	6	3	4	0	2	17
9	1	1	7	3	6	0	3	21
10	1	1	8	4	9	0	5	28
11	1	1	9	4	12	0	6	33
12	1	1	10	5	16	1	9	43
13	1	1	11	5	20	1	11	50
14	1	1	12	6	25	2	15	62
15	1	1	13	6	30	3	18	72
16	1	1	14	7	36	5	23	87
17	1	1	15	7	42	6	27	99
18	1	1	16	8	49	9	34	118
19	1	1	17	8	56	11	39	133
20	1	1	18	9	64	15	47	155

Table 4.4: Comparison of orbit efficiency ratio R_i for orbit type $i = 1..5$

Type-i orbit	$i=0$	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
R_i	1	2	3	4	6	3

While rotationally symmetric cubature rules generate more moment equations for degrees higher than five (see Table3.3 and 4.3), they can require fewer integration points than their fully symmetric counterparts for the same degree and quality.

Defining the structure of a rotationally symmetric cubature rule as $[n_0, n_1, n_2, n_3, n_4, n_5]$, with n_5 representing the number of type-5 orbits. Similar to Equation (4.12), the total number of points n_K for a rotationally symmetric rule with orbit structure $[n_0, n_1, n_2, n_3, n_4, n_5]$ is:

$$n_K = n_0 + 4n_1 + 6n_2 + 12n_3 + 24n_4 + 12n_5 \tag{4.12}$$

while the number of unknowns is

$$n_e = n_0 + 2n_1 + 2n_2 + 3n_3 + 4n_4 + 4n_5. \tag{4.13}$$

For analysis, we introduce orbit efficiency ratio R_i defined as:

$$R_i = \frac{a_i}{b_i} \tag{4.14}$$

where a_i and b_i are the number of points and variables for a type-i orbit, respectively. R_i indicates type-i orbit's contribution to the number of integration points per variable. A smaller R_i value indicates greater efficiency, as it requires fewer integration points for the same number of variables.

As shown in Table 4.1, a type-5 orbit uses 12 fewer integration points than a type-4 orbit with the same number of variables. Compared to a type-3 orbit, a type-5 orbit involves one more variable but the same number of integration points. Type-3 and type-4 orbits have higher R_i values than type-5, as shown in Table 4.4. Thus, a rule structure $[*, *, *, 0, 0, *]$ is advantageous, often yielding cubature rules with fewer points. For instance, using the same method described in the last chapter but with a different basis, a degree 7 rotationally symmetric cubature rule with the structure $[0, 2, 0, 0, 0, 2]$ results in the following moment equations:

$$1 = 4\sum_{k=1}^{2} w_{1,k} + 12\sum_{k=1}^{2} w_{5,k} \tag{4.15a}$$

$$\frac{1}{20} = 4\sum_{k=1}^{2} w_{1,k}a_k^2 + 12\sum_{k=1}^{2} w_{5,k}p_{5,k} \tag{4.15b}$$

$$\frac{1}{120} = 4\sum_{k=1}^{2} w_{1,k}a_k^3 + 12\sum_{k=1}^{2} w_{5,k}q_{5,k} \tag{4.15c}$$

$$\frac{1}{1260} = 12\sum_{k=1}^{2} w_{5,k}r_{5,k} \tag{4.15d}$$

$$\frac{1}{336} = 4\sum_{k=1}^{2} w_{1,k}a_k^4 + 12\sum_{k=1}^{2} w_{5,k}\left(p_{5,k}^2 - r_{5,k}\right) \tag{4.15e}$$

$$\frac{1}{1120} = 4\sum_{k=1}^{2} w_{1,k}a_k^5 + 12\sum_{k=1}^{2} w_{5,k}p_{5,k}q_{5,k} \tag{4.15f}$$

$$\frac{1}{19440} = 12\sum_{k=1}^{2} w_{5,k}p_{5,k}r_{5,k} \tag{4.15g}$$

$$\frac{67}{604800} = 4\sum_{k=1}^{2} w_{1,k}a_k^7 + 12\sum_{k=1}^{2} w_{5,k}p_{5,k}^2q_{5,k} \tag{4.15h}$$

$$\frac{59}{181440} = 4\sum_{k=1}^{2} w_{1,k}a_k^6 + 12\sum_{k=1}^{2} w_{5,k}p_{5,k}\left(p_{5,k}^2 - r_{5,k}\right) \tag{4.15i}$$

$$\frac{1}{22680} = 12\sum_{k=1}^{2} w_{5,k}\left(p_{5,k}^3 - p_{5,k}r_{5,k} - q_{5,k}^2\right) \tag{4.15j}$$

$$\frac{1}{151200} = 12\sum_{k=1}^{2} w_{5,k}q_{5,k}r_{5,k} \tag{4.15k}$$

$$0 = 12\sum_{k=1}^{2} w_{5,k}v_k \tag{4.15l}$$

Moment equations (4.15) form a positive-dimensional system because v is connected to the discriminant by the relationship (4.9) and is not an independent variable. To make this system zero-dimensional and obtain finite solutions, additional n_5 equations are required:

$$(v_1)^2 = 27\left(-q_{5,1}^4 + 2p_{5,1}\left(p_{5,1}^2 - 3r_{5,1}\right)q_{5,1}^2 - \left(p_{5,1}^2 - 4r_{5,1}\right)\left(p_{5,1}^2 - r_{5,1}\right)^2\right) \tag{4.16a}$$

$$(v_2)^2 = 27\left(-q_{5,2}^4 + 2p_{5,2}\left(p_{5,2}^2 - 3r_{5,2}\right)q_{5,2}^2 - \left(p_{5,2}^2 - 4r_{5,2}\right)\left(p_{5,2}^2 - r_{5,2}\right)^2\right) \tag{4.16b}$$

The system formed by Equations (4.15) and (4.16) includes 14 equations and 14 variables. The solutions provide coordinates and weights for the cubature rule generators. As discussed in Section 2.3, there are two main approaches to solve such polynomial systems: algebraic and numerical methods. Algebraic solutions enumerate all possible rules, while numerical

algorithms typically converge to a single solution. Algebraic methods are feasible for lower degree cubature rules, such as $[0,2,0,0,0,2]$ for degree 7, but for higher degrees, the moment equations become too intricate. Thus, numerical methods are recommended for higher degree cases.

4.1.3 Numerical algorithm

To obtain a numerical solution, we adapted the approach from (L. Zhang et al., 2009) to convert the algebraic system into a nonlinear least squares problem. This is solved using the PHG code, which employs the $MINPACK$ routines *lmder1* and *lmdif1*, described in (More, Garbow, & Hillstrom, n.d.). These routines implement the Levenberg-Marquardt algorithm, which efficiently finds local minima of nonlinear least squares problems.

The Levenberg–Marquardt algorithm is a popular optimization technique used to solve non-linear least squares problems. It is particularly useful in cases where the goal is to minimize the sum of squared residuals between observed data and model predictions. The algorithm combines the features of two classical optimization methods: Gauss-Newton and gradient descent methods, making it effective for a wide range of problems, especially in curve fitting and parameter estimation.

The Levenberg–Marquardt algorithm is designed to minimize a cost function of the form:

$$R(\boldsymbol{x}) = \sum_{i}^{m} (f_i(\boldsymbol{x}))^2 \tag{4.17}$$

where $f_i(\boldsymbol{x})$ represents the residuals (the difference between observed and predicted values), and $\boldsymbol{x}$ is the vector of parameters to be optimized. The objective is to find the parameter vector $\boldsymbol{x}$ that minimizes the total squared error. At each iteration, the Levenberg–Marquardt algorithm updates the parameter vector $\boldsymbol{x}$ by solving the following modified normal equation:

$$(\boldsymbol{J}^T \boldsymbol{J} + \lambda \boldsymbol{I}) \Delta \boldsymbol{x} = -\boldsymbol{J}^T \boldsymbol{r} \tag{4.18}$$

where J is the Jacobian matrix of partial derivatives of the residuals $f_i(\boldsymbol{x})$, $\boldsymbol{r}$ is the vector of residuals, $\boldsymbol{I}$ is the identity matrix, is a damping parameter that controls the step size, λ is a damping parameter that controls the step size, $\Delta \boldsymbol{x}$ is the parameter update step. The damping parameter λ governs the transition between the Gauss-Newton (small λ) and gradient descent (large λ) steps. If the update improves the solution, λ is reduced, allowing the algorithm to move toward the Gauss-Newton method. If the update worsens the solution, λ is increased, making the algorithm behave more like gradient descent.

Algorithm 2 Algorithm to compute rotationally symmetric cubature rule in tetrahedron

$d \leftarrow$ degree of the cubature rule
$n \leftarrow$ target number of points
1) Generate bases for degree d rotationally symmetric polynomials
for each decomposition $n = n_0 + 4n_1 + 6n_2 + 12n_5$ **do**
  while Number of tries $\leq$ Maximum number of initial guesses **do**
    2) Randomly choose an initial guess.
    3) Find least square solution to the algebraic equations for the given decomposition$[n_0, n_1, n_2, 0, 0, n_5]$.
    4) Break when a "good" solution is found.
  end while
end for

The outcomes depend heavily on the initial parameter estimates. The key difference between *lmder1* and *lmdif1* is their handling of Jacobians: *lmder1* requires an analytical Jacobian, while *lmdif1* calculates Jacobians using finite differences. We chose *lmdif1* for its ability to handle Jacobians automatically, making it suitable for our applications.

In the solution process, we consider $n_0 + n_1 + n_2 + n_5$ weights as independent variables (with n_3 and n_4 being zero). The other unknowns are coordinate variables in different orbits: n_1 for type-1 orbits, n_2 for type-2 orbits, and $3n_5$ for type-5 orbits. Therefore, we need to solve a nonlinear least squares problem involving $n_0 + 2n_1 + 2n_2 + 4n_5$ unknowns.

Algorithm 2 details the computation of cubature rules, defining a "good" solution as one with "zero" residual and PI quality. To expedite the search, the algorithm uses MPI for parallelization, testing varied initial guesses across different processes. Computation stops across all processes once a "good" solution is found.

For the step 1) in Algorithm 2, the fully symmetric basis generated in (L. Zhang et al., 2009) is

$$B_{FS} = \{\langle i, j, k, l\rangle | i + j + k + l = d, i \geq j \geq k \geq l \geq 0\} \tag{4.19}$$

where i, j, k and l are the integer exponents of monomial $L_1^i L_2^j L_3^k L_4^l$. Let

$$B_d = \{\langle i, j, k, l\rangle | i + j + k + l = d\} \tag{4.20}$$

be the asymmetric basis of degree d, fully symmetric bases B_{FS} satisfy

$$G_{FS}(B_{FS}) = B_d. \tag{4.21}$$

Similarly, in the case of rotational symmetry, we need to find a rotationally symmetric basis B_{RS} such that

$$G_{RS}(B_{RS}) = B_d. \tag{4.22}$$

Because fully symmetric bases B_{FS} is G_{RS}-invariant, we can divide B_{RS} into two parts

$$B_{RS} = B'_{RS} + B_{FS}. \tag{4.23}$$

where B'_{RS} is the discrepancy between fully symmetric basis B_{FS} and rotationally symmetric basis B_{RS}. Substitute (4.23) back into (4.22) we can get

$$G_{RS}\left(B'_{RS}\right) = B_d - G_{RS}(B_{FS}) \tag{4.24}$$

which can be used to calculate B'_{RS}. Then, the complete rotationally symmetric basis B_{RS} can be achieved by Equation (4.23).

Let $|B|$ be the cardinality of set B. For example, consider a degree 7 cubature rule for tetrahedron, fully symmetric basis B_{FS} is

$$\begin{aligned}\{&\langle 2,2,2,1\rangle, \langle 3,2,1,1\rangle, \langle 3,2,2,0\rangle, \langle 3,3,1,0\rangle,\\ &\langle 4,1,1,1\rangle, \langle 4,2,1,0\rangle, \langle 4,3,0,0\rangle, \langle 5,1,1,0\rangle,\\ &\langle 5,2,0,0\rangle, \langle 6,1,0,0\rangle, \langle 7,0,0,0\rangle\}.\end{aligned} \tag{4.25}$$

By addressing a linear integer programming problem in (4.20), we establish that the cardinality of B_d is 120. The rotationally symmetric linear transformations applied to the fully symmetric bases yield $|G_{RS}(B_{FS})| = 108$. Consequently, this leads to the following relation for the rotationally symmetric basis:

$$|G_{RS}\left(B'_{RS}\right)| = |B_d| - |G_{RS}(B_{FS})| = 12. \tag{4.26}$$

In this example, $G_{RS}\left(B'_{RS}\right)$ represents a single complete set containing twelve elements generated by all transformations in G_{RS}:

$$\begin{aligned}\{&\langle 0,1,4,2\rangle, \langle 0,2,1,4\rangle, \langle 0,4,2,1\rangle, \langle 1,0,2,4\rangle,\\ &\langle 1,2,4,0\rangle, \langle 1,4,0,2\rangle, \langle 2,0,4,1\rangle, \langle 2,1,0,4\rangle,\\ &\langle 2,4,1,0\rangle, \langle 4,0,1,2\rangle, \langle 4,1,2,0\rangle, \langle 4,2,0,1\rangle\}.\end{aligned} \tag{4.27}$$

The generator could be any element from this set, such as $\langle 0,1,4,2\rangle$, which means

Table 4.5: Comparison between the fewest number of integration points in fully symmetric PI rules and rotationally symmetric PI rules on the tetrahedron. Underlined results are newly obtained in this work.

degree	Fully symmetric	Rotationally symmetric
7	35	<u>32</u>
8	46	
9	59	<u>58</u>
10	79	<u>78</u>
11	96	<u>96</u>

$$B'_{RS} = \langle 0,1,4,2 \rangle, \tag{4.28}$$

then the complete rotationally symmetric basis B_{RS} is constructed using Equation (4.23) and the final result is:

$$\begin{aligned} \{&\langle 2,2,2,1 \rangle, \langle 3,2,1,1 \rangle, \langle 3,2,2,0 \rangle, \langle 3,3,1,0 \rangle, \\ &\langle 4,1,1,1 \rangle, \langle 4,2,1,0 \rangle, \langle 4,3,0,0 \rangle, \langle 5,1,1,0 \rangle, \\ &\langle 5,2,0,0 \rangle, \langle 6,1,0,0 \rangle, \langle 7,0,0,0 \rangle, \langle 0,1,4,2 \rangle \}. \end{aligned} \tag{4.29}$$

4.1.4 Rotationally symmetric cubature rules

Using the algorithm in Subsection 4.1.3, we obtained several new rotationally symmetric cubature rules. Table 4.5 lists the lowest number of integration points for fully symmetric and rotationally symmetric PI rules for degrees 7 to 11, with data for fully symmetric PI rules sourced from Table 3.7.

For degree 7, we found a new 32-point PI rule with the structure $[0,2,0,0,0,2]$, detailed in Table 4.6. This rule uses 3 fewer points than the optimal fully symmetric PI rule. No PI rules with fewer points than the existing fully symmetric PI rule were found for degree 8. For degree 9, we developed a 58-point PI rule with the structure $[0,4,1,0,0,3]$, detailed in Table 4.7, using one fewer point than the previous optimal fully symmetric PI rule. For degree 10, we introduced a 78-point PI rule with the structure $[0,3,1,0,0,5]$, detailed in Table 4.8. For degree 11, we found a unique 96-point PI rule with the structure $[0,3,0,0,0,7]$, matching the point count of the optimal fully symmetric PI rule, detailed in Table 4.9.

Table 4.6: Degree 7, 32-point PI rule generators and weights. The barycentric coordinates of the generators in terms of α, β and γ are given in table 4.1.

Orbit type	weight	α	β	γ
1	0.0203319531327237	0.0593471101264938		
1	0.0455596863797766	0.19804620211073		
5	0.0343803492023788	0.0414352322990194	0.48378638391719	0.234996960850308
5	0.0268880910017775	0.606335249960351	0.0509373110576675	0.0459752056658163

Table 4.7: Degree 9, 58-point PI rule generators and weights.

Orbit type	weight	α	β	γ
1	0.00653162827530929	0.0405197694131099		
1	0.0413745817197838	0.152919308605651		
1	0.0483444012894103	0.297459101293375		
1	0.00648356477307691	0.33332365529637		
2	0.00326830253568522	0.00667361004635429		
5	0.0225528607623228	0.123529961995743	0.447935263469537	0.0410746122583638
5	0.0126206504251977	0.718246812066329	0.0352339035614949	0.0434849786603688
5	0.0122809455254433	0.224036097221082	0.014754144096144	0.167637727995568

Table 4.8: Degree 10, 78-point PI rule generators and weights.

Orbit type	weight	α	β	γ
1	0.0281951930587875	0.159857819792385		
1	0.000388086245773334	0.00471408676386377		
1	0.0303619819792753	0.291403856648543		
2	0.0224810411428727	0.116713796273249		
5	0.0144377307043309	0.416108442130229	0.3589881550926	0.20137211643434
5	0.0103036281440898	0.0269407779999434	0.562063809098776	0.321320801565786
5	0.00556999803005424	0.35011918560089	0.598422574481169	0.013360955835962
5	0.00665292857714934	0.810671262448602	0.127299261998876	0.0281593952301471
5	0.0154801068783271	0.630674524119898	0.181920694630335	0.153397891101455

Table 4.9: Degree 11, 96-point PI rule generators and weights.

Orbit type	weight	α	β	γ
1	0.0202494603204022	0.154850739615501		
1	0.0278120469783064	0.288179247566595		
1	0.0140720316875435	0.325172326555607		
5	0.00618849446361804	0.379139457038439	0.561756035433476	0.0207803335884142
5	0.00323105817314915	0.757357255478599	0.200766966848501	0.00970488192494772
5	0.0149757360808016	0.133480561919289	0.537526571473883	0.0299885993542045
5	0.0111006954610384	0.713934127575933	0.101131083868683	0.0334947401906106
5	0.00158107598710846	0.891842704603017	0.00903623899783386	0.0598626558204338
5	0.00643025000626648	0.161678676429346	0.516845922370527	0.311345181128638
5	0.019114843499267	0.151211470673164	0.305701661532563	0.0872889549382812

4.2 Multisymmetric polynomials in cubature formulation

Multisymmetric polynomials extend symmetric polynomials to multiple sets of variables, reflecting symmetry in algebraic expressions. They explore the invariance and patterns when multiple variable sets undergo permutation symmetries. Advances in multisymmetric polynomials have been driven by invariant theory and group theory, notably through Briand's work, which examines them as invariants under the symmetric group S_n. His research provides computational tools and theoretical insights into their structure and applications (Briand, 2002). Briand also discusses whether the ring of multisymmetric polynomials can be generated by elementary multisymmetric polynomials or multisymmetric power sums (Briand, 2004).

The study of the minimal generating set for the multisymmetric polynomial ring has been pivotal in advancing algebraic theory. Fleischmann (1998) provided foundational insights, which Vaccarino expanded into a general framework for generating multisymmetric functions within the polynomial ring $A_R(n,m)$ under the action of the symmetric group S_n (Vaccarino, 2005). Domokos contributed fundamental theorems on polynomial invariants of pseudo-reflection groups like Weyl groups of type B_n, offering a detailed presentation of multisymmetric polynomials and proving the reducedness of the invariant commuting scheme for any base ring (Domokos, 2007). He also demonstrated that the ideal of relations among a minimal generating system is mainly generated by polarizations of specific relations in both three-dimensional and n-dimensional vector variables (Domokos & Puskás, 2012). Lopatin and Reimers (2021) recently examined the differences between generating and separating invariants in multisymmetric polynomials.

However, despite extensive research, there has been no connection to the formulation of cubature rules. In Section 3.4.2, invariance under the permutation of orbits of the same type was introduced to simplify algebraic solutions. Extending this concept to general cubature rules and treating each point as an orbit allows the formulation to inherently exhibit multisymmetry, remaining unchanged under any permutation of points. Incorporating multisymmetric polynomials into the moment equations can simplify them, making the nonlinear systems more manageable for algebraic solutions. Additionally, using multisymmetric polynomials eliminates permutation invariance within each orbit type, potentially simplifying numerical solutions.

4.2.1 Multisymmetric polynomials in moment equations

To develop new Gaussian-type cubature rules, deriving and solving moment equations is crucial. For a cubature rule with degree d, the general form of the moment equations for non-negative integers i is:

$$I_{i_1,i_2,\ldots,i_n} = \sum_{k=1}^{N} w_k x_{1,k}^{i_1} x_{2,k}^{i_2} \ldots x_{n,k}^{i_n}. \tag{4.30}$$

Here, $I_{i_1,i_2,\ldots,i_n}$ represents the exact integral of the monomial $x_{1,k}^{i_1} x_{2,k}^{i_2} \ldots x_{n,k}^{i_n}$ over the domain satisfying $i_1 + i_2 + \cdots + i_n \leq d$, n is the dimension of the domain, N is the number of integration points, w_k is the weight of the kth point, and $(x_{1,k}, x_{2,k}, \ldots, x_{n,k})$ are the coordinates of the kth point.

Consider a degree two asymmetric cubature rule for a triangle using three integration points. The corresponding moment equations involving the weights w_k and coordinates (x_k, y_k) with $k = 1 \ldots 3$ are structured as follows:

$$w_1 + w_2 + w_3 = I_{0,0} \tag{4.31a}$$
$$w_1 x_1 + w_2 x_2 + w_3 x_3 = I_{1,0} \tag{4.31b}$$
$$w_1 x_1^2 + w_2 x_2^2 + w_3 x_3^2 = I_{2,0} \tag{4.31c}$$
$$w_1 y_1 + w_2 y_2 + w_3 y_3 = I_{0,1} \tag{4.31d}$$
$$w_1 x_1 y_1 + w_2 x_2 y_2 + w_3 x_3 y_3 = I_{1,1} \tag{4.31e}$$
$$w_1 y_1^2 + w_2 y_2^2 + w_3 y_3^2 = I_{0,2} \tag{4.31f}$$

In analyzing the moment equations (4.31), the right-hand side remains invariant under permutations of integration points and their weights, $\{w_k, x_k, y_k\}$. Consequently, the left-hand side forms multisymmetric polynomials. This study emphasizes the practical application of these polynomials in numerical methods and computational engineering, rather than their purely mathematical properties.

Definition of multisymmetric polynomials

Multisymmetric polynomials extend the concept of symmetric polynomials to multiple sets of variables. Instead of dealing with symmetric polynomials in a single set of variables x_1, x_2,...,x_n, multisymmetric polynomials handle tuples of these variables across $m \geq 2$ groups. In the context of invariant theory, considering a field $\mathbb{K}$, a symmetric group S_n, and its standard representation on $\Omega = \mathbb{K}^n$, symmetric polynomials form the elements of the ring $\mathbb{K}[\Omega]^{S_n}$. Similarly, multisymmetric polynomials belong to the invariant ring $\mathbb{K}[\Omega^m]^{S_n}$, which captures the action of S_n diagonally across the m groups of Ω.

Let m and n be positive integers defining the dimensions of a matrix X, which organizes the arguments of a polynomial P as follows:

$$X = \begin{bmatrix} x_{11} & \ldots & x_{n1} \\ \vdots & \ddots & \vdots \\ x_{1m} & \ldots & x_{nm} \end{bmatrix} \tag{4.32}$$

where x_{ij} with $i = 1..n$ and $j = 1..m$ are complex numbers. The polynomial P is defined as multisymmetric if it remains unchanged under any permutation of the rows or columns of X. For instance, in the scenario of a degree 2 cubature rule in a triangle with three integration points, the matrix X might look like:

$$X = \begin{bmatrix} w_1 & x_1 & y_1 \\ w_2 & x_2 & y_2 \\ w_3 & x_3 & y_3 \end{bmatrix} \tag{4.33}$$

and the permutation of points is equivalent to the permutation of rows.

Elementary multisymmetric polynomials

Elementary multisymmetric polynomials form the generating set for the invariant ring $\mathbb{K}[\Omega^m]^{S_n}$. This means multisymmetric polynomials with rational coefficients can be represented using elementary multisymmetric polynomials. This theoretical foundation, first established by Schlafli (1852) and further developed by MacMahon (1916); Noether (1915); Weyl (1946), allows the multisymmetric polynomials in the moment equations to be expressed using elementary multisymmetric polynomials. Considering $1 \leq t \leq n$,

$$t = \sum_{i=1}^{m} \alpha_i, \tag{4.34}$$

the elementary multisymmetric polynomials $e_{\alpha_1,\ldots,\alpha_m}$ used in our work are defined as

$$e_{\alpha_1,\ldots,\alpha_m} = \sum_{1 \leq i_1 \leq i_2 \cdots \leq i_t \leq n} \boldsymbol{x}_{i_1}^{\tilde{k}_1} \boldsymbol{x}_{i_2}^{\tilde{k}_2} \ldots \boldsymbol{x}_{i_t}^{\tilde{k}_t} \tag{4.35}$$

where

$$\boldsymbol{x}_{i_v}^{\tilde{k}_v} = \prod_{u=1}^{m} x_{u,v}^{k_{u,v}} \quad \text{for } v = 1 \ldots t \tag{4.36}$$

and $k_{u,v}$ satisfy

$$k_{u,v} = \begin{cases} 1 & \text{if } \sum_{l=1}^{v} k_{u,l} < \alpha_u \text{ and } \sum_{w=1}^{m} k_{w,v} = 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.37}$$

Elementary multisymmetric polynomials can be computed using the formula provided in (Vaccarino, 2005):

$$\sum_{\sum_{j=1}^{m}\alpha_j\leq n} v_1^{\alpha_1}\dots v_m^{\alpha_m} e_{\alpha_1\dots\alpha_m} = \prod_{i=1}^{n}\left(1+\sum_{j=1}^{m} v_j x_{j,i}\right) \tag{4.38}$$

In this equation, v represents commuting independent variables. Specifically, when $m=1$, the equation simplifies to the elementary symmetric polynomials:

$$\sum_{k=0}^{n} v^k e_k = \prod_{i=1}^{n}(1+vx_i) \tag{4.39}$$

The multisymmetric polynomials in the moment equations for the three-point cubature rule on a triangle belong to the invariant ring $\mathbb{R}[\Omega^3]^{S_3}$. Here are the formulas calculated by (4.38) for the complete elementary multisymmetric polynomials $e_{\alpha_1,\alpha_2,\alpha_3}(w,x,y)$:

$$e_{0,0,0} = 1 \tag{4.40a}$$
$$e_{0,0,1} = y_1+y_2+y_3 \tag{4.40b}$$
$$e_{0,0,2} = y_1y_2+y_1y_3+y_2y_3 \tag{4.40c}$$
$$e_{0,0,3} = y_1y_2y_3 \tag{4.40d}$$
$$e_{0,1,0} = x_1+x_2+x_3 \tag{4.40e}$$
$$e_{0,1,1} = x_1y_2+x_1y_3+x_2y_1+x_2y_3+x_3y_1+x_3y_2 \tag{4.40f}$$
$$e_{0,1,2} = x_1y_2y_3+x_2y_1y_3+x_3y_1y_2 \tag{4.40g}$$
$$e_{0,2,0} = x_1x_2+x_1x_3+x_2x_3 \tag{4.40h}$$
$$e_{0,2,1} = x_1x_2y_3+x_1x_3y_2+x_2x_3y_1 \tag{4.40i}$$
$$e_{0,3,0} = x_1x_2x_3 \tag{4.40j}$$
$$e_{1,0,0} = w_1+w_2+w_3 \tag{4.40k}$$
$$e_{1,0,1} = w_1y_2+w_1y_3+w_2y_1+w_2y_3+w_3y_1+w_3y_2 \tag{4.40l}$$
$$e_{1,0,2} = w_1y_2y_3+w_2y_1y_3+w_3y_1y_2 \tag{4.40m}$$
$$e_{1,1,0} = w_1x_2+w_1x_3+w_2x_1+w_2x_3+w_3x_1+w_3x_2 \tag{4.40n}$$
$$e_{1,1,1} = w_1x_2y_3+w_1x_3y_2+w_2x_1y_3+w_2x_3y_1+w_3x_1y_2+w_3x_2y_1 \tag{4.40o}$$
$$e_{1,2,0} = w_1x_2x_3+w_2x_1x_3+w_3x_1x_2 \tag{4.40p}$$
$$e_{2,0,0} = w_1w_2+w_1w_3+w_2w_3 \tag{4.40q}$$
$$e_{2,0,1} = w_1w_2y_3+w_1w_3y_2+w_2w_3y_1 \tag{4.40r}$$
$$e_{2,1,0} = w_1w_2x_3+w_1w_3x_2+w_2w_3x_1 \tag{4.40s}$$
$$e_{3,0,0} = w_1w_2w_3 \tag{4.40t}$$

These polynomials encapsulating intricate relationships in a structured manner suitable for algebraic manipulation and analysis.

4.2.2 Simplifying moment equations

For the three-point cubature rule in a triangle, only a subset of the elementary multisymmetric polynomials from (4.40) is needed. Specifically, we select polynomials where the index α_1 is 0 or 1, as only one weight is required for each set of variables. Additionally, the sum of indices α_2 and α_3 does not exceed 2, reflecting the need for two independent coordinate variables in a triangle. The relevant elementary multisymmetric polynomials for this scenario are:

$$e_{0,0,1} = y_1 + y_2 + y_3 \tag{4.41a}$$
$$e_{0,0,2} = y_1y_2 + y_1y_3 + y_2y_3 \tag{4.41b}$$
$$e_{0,1,0} = x_1 + x_2 + x_3 \tag{4.41c}$$
$$e_{0,1,1} = x_1y_2 + x_1y_3 + x_2y_1 + x_2y_3 + x_3y_1 + x_3y_2 \tag{4.41d}$$
$$e_{0,2,0} = x_1x_2 + x_1x_3 + x_2x_3 \tag{4.41e}$$
$$e_{1,0,0} = w_1 + w_2 + w_3 \tag{4.41f}$$
$$e_{1,0,1} = w_1y_2 + w_1y_3 + w_2y_1 + w_2y_3 + w_3y_1 + w_3y_2 \tag{4.41g}$$
$$e_{1,0,2} = w_1y_2y_3 + w_2y_1y_3 + w_3y_1y_2 \tag{4.41h}$$
$$e_{1,1,0} = w_1x_2 + w_1x_3 + w_2x_1 + w_2x_3 + w_3x_1 + w_3x_2 \tag{4.41i}$$
$$e_{1,1,1} = w_1x_2y_3 + w_1x_3y_2 + w_2x_1y_3 + w_2x_3y_1 + w_3x_1y_2 + w_3x_2y_1 \tag{4.41j}$$
$$e_{1,2,0} = w_1x_2x_3 + w_2x_1x_3 + w_3x_1x_2 \tag{4.41k}$$

As discussed, moment equations like (4.31) can be interpreted as multisymmetric polynomials. To represent these equations using elementary multisymmetric polynomials, we apply the product rule detailed in (Vaccarino, 2005). For $k, h \in \mathbb{N}$, the product rule is given by:

$$e_{\alpha_1,\ldots,\alpha_k}(f_1,\ldots,f_k)e_{\beta_1,\ldots,\beta_h}(g_1,\ldots,g_h) = \sum_{\gamma} e_{\gamma}(f_1,\ldots,f_k,g_1,\ldots,g_h,f_1g_1,f_1g_2,\ldots,f_kg_1,\ldots,f_kg_h) \tag{4.42}$$

where

$$\gamma = (\gamma_{10},\ldots,\gamma_{k0},\gamma_{01},\ldots,\gamma_{0h},\gamma_{11},\gamma_{12},\ldots,\gamma_{k1},\ldots,\gamma_{kh}). \tag{4.43}$$

The indices γ satisfy the conditions:

$$\begin{cases} \gamma_{ij} \in \mathbb{N} \\ |\gamma| \leq n \\ \sum_{j=0}^{h} \gamma_{ij} = \alpha_i, \quad \text{for } i = 1..k \\ \sum_{i=0}^{k} \gamma_{ij} = \beta_j, \quad \text{for } j = 1..h \end{cases} \tag{4.44}$$

The product rule facilitates expressing moment Equations (4.31) using elementary multisymmetric polynomials (4.41). For instance, Equation (4.31b) can be transformed to:

$$I_{1,0} = e_{0,0,1}(w,x,wx) = e_{1,0,0}(w,x,wx)e_{0,1,0}(w,x,wx) - e_{1,1,0}(w,x,wx) \tag{4.45}$$

where according to the definition of elementary multisymmetric polynomial we have

$$e_{1,0,0}(w,x,wx) = e_{1,0,0} \tag{4.46a}$$
$$e_{0,1,0}(w,x,wx) = e_{0,1,0} \tag{4.46b}$$
$$e_{1,1,0}(w,x,wx) = e_{1,1,0} \tag{4.46c}$$

Therefore, Equation (4.45) rewrites as:

$$I_{1,0} = e_{1,0,0}e_{0,1,0} - e_{1,1,0} \tag{4.47}$$

Similar transformations allow the moment equations (4.31) to be restated as:

$$e_{1,0,0} = I_{0,0} \tag{4.48a}$$
$$e_{0,1,0}e_{1,0,0} - e_{1,1,0} = I_{1,0} \tag{4.48b}$$
$$e_{0,1,0}^2 e_{1,0,0} - e_{0,1,0}e_{1,1,0} - e_{0,2,0}e_{1,0,0} + e_{1,2,0} = I_{2,0} \tag{4.48c}$$
$$e_{0,0,1}e_{1,0,0} - e_{1,0,1} = I_{0,1} \tag{4.48d}$$
$$e_{0,0,1}e_{0,1,0}e_{1,0,0} - \frac{1}{2}e_{0,0,1}e_{1,1,0} - \frac{1}{2}e_{0,1,0}e_{1,0,1} - \frac{1}{2}e_{0,1,1}e_{1,0,0} + \frac{1}{2}e_{1,1,1} = I_{1,1} \tag{4.48e}$$
$$e_{0,0,1}^2 e_{1,0,0} - e_{0,0,1}e_{1,0,1} - e_{0,0,2}e_{1,0,0} + e_{1,0,2} = I_{0,2} \tag{4.48f}$$

The revised system of equations (4.48) features simpler polynomials of the same order as those in the original system (4.31), making them easier to solve. This streamlined structure suits algebraic solvers better and may facilitate the development of new cubature rules. Computing the Gröbner basis for system (4.48) yields 6 equations, compared to 21 from the original system (4.31). This significant reduction in complexity in this very simple example potentially suggests an even larger gap with more points or higher dimensions.

While the reformulated moment equations reduce complexity, they introduce a new challenge: the system now has 6 equations with 11 unknowns, compared to the original's 9 unknowns. This implies that the 11 elementary multisymmetric polynomials are not independent, requiring two additional equations to define their relationships.

To identify these supplementary equations, one can use the Gröbner basis method on the system (4.41) to isolate independent equations excluding the variables x, y and w. The resulting relationships between elementary multisymmetric polynomials are shown in Equations (4.49), (4.50) and (4.51).

$$\begin{aligned}
&e_{0,0,1}^2 e_{1,1,0}^2 - 2e_{0,0,1}e_{0,1,0}e_{1,0,1}e_{1,1,0} - 2e_{0,0,1}e_{0,1,1}e_{1,0,0}e_{1,1,0} + 4e_{0,0,1}e_{0,2,0}e_{1,0,0}e_{1,0,1} \\
&+ 4e_{0,0,2}e_{0,1,0}e_{1,0,0}e_{1,1,0} - 4e_{0,0,2}e_{0,2,0}e_{1,0,0}^2 + e_{0,1,0}^2 e_{1,0,1}^2 - 2e_{0,1,0}e_{0,1,1}e_{1,0,0}e_{1,0,1} + e_{0,1,1}^2 e_{1,0,0}^2 \\
&- 4e_{0,0,1}e_{1,0,1}e_{1,2,0} + 2e_{0,0,1}e_{1,1,0}e_{1,1,1} + 4e_{0,0,2}e_{1,0,0}e_{1,2,0} - 4e_{0,0,2}e_{1,1,0}^2 + 2e_{0,1,0}e_{1,0,1}e_{1,1,1} \\
&- 4e_{0,1,0}e_{1,0,2}e_{1,1,0} - 2e_{0,1,1}e_{1,0,0}e_{1,1,1} + 4e_{0,1,1}e_{1,0,1}e_{1,1,0} + 4e_{0,2,0}e_{1,0,0}e_{1,0,2} - 4e_{0,2,0}e_{1,0,1}^2 \\
&+ 12e_{1,0,2}e_{1,2,0} - 3e_{1,1,1}^2 = 0
\end{aligned} \tag{4.49}$$

$$\begin{aligned}
&2e_{0,0,1}^2 e_{0,2,0}e_{1,0,0} - 2e_{0,0,1}e_{0,1,0}e_{0,1,1}e_{1,0,0} + 2e_{0,0,2}e_{0,1,0}^2 e_{1,0,0} - 2e_{0,0,1}^2 e_{1,2,0} + 2e_{0,0,1}e_{0,1,0}e_{1,1,1} \\
&+ e_{0,0,1}e_{0,1,1}e_{1,1,0} - 2e_{0,0,1}e_{0,2,0}e_{1,0,1} - 2e_{0,0,2}e_{0,1,0}e_{1,1,0} - 4e_{0,0,2}e_{0,2,0}e_{1,0,0} - 2e_{0,1,0}^2 e_{1,0,2} \\
&+ e_{0,1,0}e_{0,1,1}e_{1,0,1} + e_{0,1,1}^2 e_{1,0,0} + 6e_{0,0,2}e_{1,2,0} - 3e_{0,1,1}e_{1,1,1} + 6e_{0,2,0}e_{1,0,2} = 0
\end{aligned} \tag{4.50}$$

$$\begin{aligned}
&2e_{0,0,1}^4 e_{0,2,0}e_{1,1,0}^2 - 2e_{0,0,1}^3 e_{0,1,0}e_{0,1,1}e_{1,1,0}^2 - 4e_{0,0,1}^3 e_{0,1,0}e_{0,2,0}e_{1,0,1}e_{1,1,0} + 2e_{0,0,1}^2 e_{0,0,2}e_{0,1,0}^2 e_{1,1,0} \\
&+ 4e_{0,0,1}^2 e_{0,1,0}^2 e_{0,1,1}e_{1,0,1}e_{1,1,0} + 2e_{0,0,1}^2 e_{0,1,0}^2 e_{0,2,0}e_{1,0,1}^2 - 4e_{0,0,1}e_{0,0,2}e_{0,1,0}^3 e_{1,0,1}e_{1,1,0} \\
&- 2e_{0,0,1}e_{0,1,0}^3 e_{0,1,1}e_{1,0,1}^2 + 2e_{0,0,2}e_{0,1,0}^4 e_{1,0,1}^2 - 4e_{0,0,1}^3 e_{0,1,1}e_{1,1,0}e_{1,2,0} + 4e_{0,0,1}^3 e_{0,2,0}e_{1,1,0}e_{1,1,1} \\
&+ 8e_{0,0,1}^2 e_{0,0,2}e_{0,1,0}e_{1,1,0}e_{1,2,0} - 8e_{0,0,1}^2 e_{0,0,2}e_{0,2,0}e_{1,0,0}e_{1,2,0} - 12e_{0,0,1}^2 e_{0,0,2}e_{0,2,0}e_{1,1,0}^2 \\
&+ 4e_{0,0,1}^2 e_{0,1,0}e_{0,1,1}e_{1,0,1}e_{1,2,0} - 4e_{0,0,1}^2 e_{0,1,0}e_{0,2,0}e_{1,0,1}e_{1,1,1} - 8e_{0,0,1}^2 e_{0,1,0}e_{0,2,0}e_{1,0,2}e_{1,1,0}
\end{aligned}$$

$$
\begin{aligned}
&+2e_{0,0,1}^2e_{0,1,1}^2e_{1,0,0}e_{1,2,0}+3e_{0,0,1}^2e_{0,1,1}^2e_{1,1,0}^2+8e_{0,0,1}^2e_{0,2,0}^2e_{1,0,0}e_{1,0,2}-8e_{0,0,1}e_{0,0,2}e_{0,1,0}^2e_{1,0,1}e_{1,2,0}\\
&-4e_{0,0,1}e_{0,0,2}e_{0,1,0}^2e_{1,1,0}e_{1,1,1}+8e_{0,0,1}e_{0,0,2}e_{0,1,0}e_{0,2,0}e_{1,0,0}e_{1,1,1}\\
&+24e_{0,0,1}e_{0,0,2}e_{0,1,0}e_{0,2,0}e_{1,0,1}e_{1,1,0}+4e_{0,0,1}e_{0,0,2}e_{0,1,1}e_{0,2,0}e_{1,0,0}e_{1,1,0}\\
&-8e_{0,0,1}e_{0,0,2}e_{0,2,0}^2e_{1,0,0}e_{1,0,1}+4e_{0,0,1}e_{0,1,0}^2e_{0,1,1}e_{1,0,2}e_{1,1,0}+8e_{0,0,1}e_{0,1,0}^2e_{0,2,0}e_{1,0,1}e_{1,0,2}\\
&-2e_{0,0,1}e_{0,1,0}e_{0,1,1}^2e_{1,0,0}e_{1,1,1}-6e_{0,0,1}e_{0,1,0}e_{0,1,1}^2e_{1,0,1}e_{1,1,0}-8e_{0,0,1}e_{0,1,0}e_{0,1,1}e_{0,2,0}e_{1,0,0}e_{1,0,2}\\
&-e_{0,0,1}e_{0,1,1}^3e_{1,0,0}e_{1,1,0}+2e_{0,0,1}e_{0,1,1}^2e_{0,2,0}e_{1,0,0}e_{1,0,1}-8e_{0,0,2}^2e_{0,1,0}e_{0,2,0}e_{1,0,0}e_{1,1,0}\\
&+4e_{0,0,2}e_{0,1,0}^3e_{1,0,1}e_{1,1,1}-12e_{0,0,2}e_{0,1,0}^2e_{0,2,0}e_{1,0,1}^2+2e_{0,0,2}e_{0,1,0}e_{0,1,1}^2e_{1,0,0}e_{1,1,0}\\
&+4e_{0,0,2}e_{0,1,0}e_{0,1,1}e_{0,2,0}e_{1,0,0}e_{1,0,1}-4e_{0,1,0}^3e_{0,1,1}e_{1,0,1}e_{1,0,2}+2e_{0,1,0}^2e_{0,1,1}^2e_{1,0,0}e_{1,0,2}\\
&+3e_{0,1,0}^2e_{0,1,1}^2e_{1,0,1}^2-e_{0,1,0}e_{0,1,1}^3e_{1,0,0}e_{1,0,1}+8e_{0,0,1}^2e_{0,0,2}e_{1,2,0}^2-4e_{0,0,1}^2e_{0,1,1}e_{1,1,1}e_{1,2,0}\\
&+24e_{0,0,1}^2e_{0,2,0}e_{1,0,2}e_{1,2,0}-6e_{0,0,1}^2e_{0,2,0}e_{1,1,1}^2-8e_{0,0,1}e_{0,0,2}e_{0,1,0}e_{1,1,1}e_{1,2,0}\\
&+8e_{0,0,1}e_{0,0,2}e_{0,1,1}e_{1,1,0}e_{1,2,0}-8e_{0,0,1}e_{0,0,2}e_{0,2,0}e_{1,1,0}e_{1,1,1}-24e_{0,0,1}e_{0,1,0}e_{0,1,1}e_{1,0,2}e_{1,2,0}\\
&+10e_{0,0,1}e_{0,1,0}e_{0,1,1}e_{1,1,1}^2-4e_{0,0,1}e_{0,1,1}^2e_{1,0,1}e_{1,2,0}-2e_{0,0,1}e_{0,1,1}^2e_{1,1,0}e_{1,1,1}\\
&+8e_{0,0,1}e_{0,1,1}e_{0,2,0}e_{1,0,1}e_{1,1,1}+12e_{0,0,1}e_{0,1,1}e_{0,2,0}e_{1,0,2}e_{1,1,0}-24e_{0,0,1}e_{0,2,0}^2e_{1,0,1}e_{1,0,2}\\
&-16e_{0,0,2}^2e_{0,1,0}e_{1,1,0}e_{1,2,0}+24e_{0,0,2}^2e_{0,2,0}e_{1,0,0}e_{1,2,0}+16e_{0,0,2}^2e_{0,2,0}e_{1,1,0}^2+32e_{0,0,2}e_{0,1,0}^2e_{1,0,2}e_{1,2,0}\\
&-6e_{0,0,2}e_{0,1,0}^2e_{1,1,1}^2+8e_{0,0,2}e_{0,1,0}e_{0,1,1}e_{1,0,1}e_{1,2,0}+8e_{0,0,2}e_{0,1,0}e_{0,1,1}e_{1,1,0}e_{1,1,1}\\
&-8e_{0,0,2}e_{0,1,0}e_{0,2,0}e_{1,0,1}e_{1,1,1}-8e_{0,0,2}e_{0,1,0}e_{0,2,0}e_{1,0,2}e_{1,1,0}-6e_{0,0,2}e_{0,1,1}^2e_{1,0,0}e_{1,2,0}\\
&-4e_{0,0,2}e_{0,1,1}^2e_{1,1,0}^2-12e_{0,0,2}e_{0,1,1}e_{0,2,0}e_{1,0,0}e_{1,1,1}-16e_{0,0,2}e_{0,1,1}e_{0,2,0}e_{1,0,1}e_{1,1,0}\\
&+8e_{0,0,2}e_{0,2,0}^2e_{1,0,0}e_{1,0,2}+16e_{0,0,2}e_{0,2,0}^2e_{1,0,1}^2-4e_{0,1,0}^2e_{0,1,1}e_{1,0,2}e_{1,1,1}-2e_{0,1,0}e_{0,1,1}^2e_{1,0,1}e_{1,1,1}\\
&-4e_{0,1,0}e_{0,1,1}^2e_{1,0,2}e_{1,1,0}+12e_{0,1,0}e_{0,1,1}e_{0,2,0}e_{1,0,1}e_{1,0,2}+3e_{0,1,1}^3e_{1,0,0}e_{1,1,1}+4e_{0,1,1}^3e_{1,0,1}e_{1,1,0}\\
&-2e_{0,1,1}^2e_{0,2,0}e_{1,0,0}e_{1,0,2}-4e_{0,1,1}^2e_{0,2,0}e_{1,0,1}^2-24e_{0,0,2}^2e_{1,2,0}^2+24e_{0,0,2}e_{0,1,1}e_{1,1,1}e_{1,2,0}\\
&-72e_{0,0,2}e_{0,2,0}e_{1,0,2}e_{1,2,0}+12e_{0,0,2}e_{0,2,0}e_{1,1,1}^2+12e_{0,1,1}^2e_{1,0,2}e_{1,2,0}-9e_{0,1,1}^2e_{1,1,1}^2\\
&+12e_{0,1,1}e_{0,2,0}e_{1,0,2}e_{1,1,1}=0 \quad (4.51)
\end{aligned}
$$

While feasible for a degree-two cubature rule in a triangle with three integration points, the computational cost of this method increases sharply with the cubature rule's degree. Thus, finding a direct formula to describe the relationships among elementary multisymmetric polynomials would be valuable.

Current literature, including John Paul Dalbec's 1995 PhD thesis (Dalbec, 1995), does not offer such a formula. Dalbec proposed an algorithm to calculate a Gröbner basis, similar to the method used here. This gap highlights a significant challenge: although using elementary multisymmetric polynomials to simplify moment equations is theoretically appealing, it is impractical without a method to maintain the system's dimensionality. Maintaining dimensionality ensures no critical information is added or lost, which is essential for accurately solving the system. A positive-dimensional system often has infinitely many solutions, making it difficult to pinpoint a specific solution without additional constraints. In polynomial algebra, especially

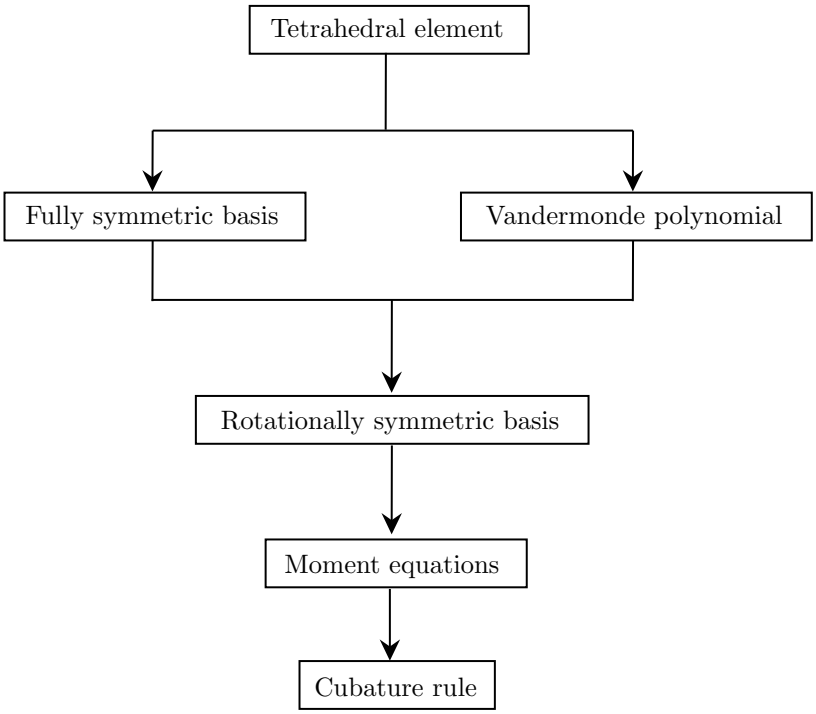


Figure 4.2: Flowchart to construct rotationally symmetric cubature rules for tetrahedra

when using Gröbner bases, the system's dimensionality affects the computation and complexity of the bases, impacting both feasibility and computation time. If an explicit formula or technique to express moment equations in elementary multisymmetric polynomials without increasing system dimension emerges, it could revolutionize solving moment equations and discovering new cubature rules.

4.3 Summary

In this chapter, we explored rotational symmetry in cubature formulation and the role of multisymmetric polynomials in the moment equations for asymmetric cubature rules. We developed a new rotationally symmetric monomial basis by incorporating a Vandermonde polynomial within a non-monomial, fully symmetric polynomial basis. This led to a comprehensive system of moment equations, augmented by additional equations from the relationship between the Vandermonde polynomial and the discriminant, to achieve zero-dimensionality. The process for constructing rotationally symmetric cubature rules for tetrahedra is outlined in Figure 4.2.

Using the Gröbner Bases method, we derived a novel degree 7 PI rule with three fewer points than the optimal fully symmetric PI rule. For higher degrees, numerical solutions were achieved using the PHG code and the Levenberg-Marquardt nonlinear least-squares algorithm. This approach successfully identified three PI rules for degrees 9 to 11, improving point efficiency with a 58-point rule for degree 9 and a 78-point rule for degree 10, and matching the best fully symmetric rule with a 96-point rule for degree 11.

Solving moment equations for degrees above 11 is challenging due to the increased number of equations and variables, requiring enhanced computational capabilities. This adaptable methodology can extend to derive rotationally symmetric cubature rules for various symmetric domains, providing a robust framework for future higher-dimensional cubature rule formulation.

While rotationally symmetric cubature rules offer advantages by reducing the number of integration points and aligning with FEM vertex numbering in specific cases, their complexity and reliance on non-optimal basis structures pose challenges for establishing consistent and accurate rule structures. Fully symmetric cubature rules, despite requiring more integration points, remain preferable for most applications due to their robustness and independence from vertex numbering.

This chapter also examines the use of multisymmetric polynomials in three-point cubature for triangles, showing they can be systematically expressed in elementary multisymmetric polynomials. However, this introduces redundancy, as these polynomials are not mutually independent. Currently, no comprehensive formula describes this interdependency.

Analyzing and computing multisymmetric polynomials is challenging, especially as variable sets become more intricate. Ongoing research aims to develop algorithms to simplify these expressions and identify their canonical forms. Exploring their algebraic properties, such as factorization patterns and zero structures, offers significant mathematical insights. Despite the complexities, studying multisymmetric polynomials in cubature formulations holds substantial potential for advancing numerical integration methods.

Chapter 5

Framework of evaluation for cubature rules

The finite element method involves approximation errors, so various tests are designed to validate new finite elements. The proposed problem set aims to help users and developers of finite element programs assess the accuracy of specific elements in different applications. These problems are not intended as benchmarks for cost comparisons, as they are generally too small to be meaningful for that purpose (O. Zienkiewicz & Taylor, 1997).

New finite elements are tested with small problems, and the results are usually published in literature or FEM program documentation (Hughes, 2012). These test results are intended to help users evaluate elements before use, but they are often inadequate because:

a) They test an insufficient number of conditions.
b) Few, if any, bad results are reported (not by design, but because developers fix only the bugs they discover).
c) They cannot usually be compared with results for other elements, especially those in different programs.

These defects would be absent in a well-designed set of standard test problems applied to various finite elements and widely circulated (Chapelle, Bathe, et al., 2011).

The importance of verifying finite element methods through independent testing and comparative assessments is increasingly recognized. This is underscored by actions such as the Nuclear Regulatory Commission's mandates for validating structural analysis software in the U.S. and the establishment of the National Agency for Finite Element Methods and Standards (NAFEMS) in the UK (Hinton et al., 1992). While the Nuclear Regulatory Commission typically uses verification tests from software developers, NAFEMS is moving towards conducting its own independent tests. This shift emphasizes the need for rigorous, unbiased evaluations in finite element analysis.

The most critical accuracy issues in modern finite elements are spurious mechanisms (rank deficiencies) and locking (excessive stiffness under certain loads or shapes). Most elements exhibit one of these issues, but not both. A major challenge is designing elements free from both problems. While elementary defects like violating rigid body properties and node renumbering invariance are rarer today, they remain problematic when they occur.

Developing a comprehensive set of test problems for finite elements involves considering parameters like loading, element and problem geometry, and material properties. The test suite should challenge each element type under different deformation scenarios. Typically, elements are tested in standard geometric configurations (squares for quadrilaterals, cubes for hexahedra, and isosceles right triangles for triangular elements). However, accuracy is also influenced by common geometric features like curvature. To effectively assess element performance, a diverse array of tests, including patch tests, beams, plates, and shell problems, is essential to cover a wide range of conditions.

5.1 Introduction to patch test

The patch test, devised by Irons (Bazeley, Cheung, Irons, & Zienkiewicz, 1966; B. Irons, 1966; B. M. Irons & Razzaque, 1972), was initially created to validate elements that breached continuity requirements and evolved into a mandatory assessment for all elements. It often identifies simple coding errors and can determine the convergence order of any finite element type. The test involves applying simple load or displacement conditions to a small cluster of adjacent elements and checking if the model accurately replicates the expected response. Figures5.1 and 5.2 illustrate common two- and three-dimensional patch tests. The element shapes and load orientations in the patch test lack specific symmetries, indicating that an element type passing this test will likely yield correct results in any scenario involving uniform stress states.

In theory, finite elements are repeatedly subdivided within an idealized model, achieving more uniform strain distribution as smoothness criteria are met. As subdivision progresses, strain nonuniformity within each element becomes negligible. The patch test ensures precise outcomes for any constant strain state across various element shapes, minimizing errors due to nonuniform strain. Passing the constant strain patch test is a fundamental indicator of an element's convergence to accurate simulation results. The sufficiency and necessity of this test are discussed in (de Arantes e Oliveira, 1977; Strang, 1972; Taylor, Simo, Zienkiewicz, & Chan, 1986). Table 5.1 lists part of key works in the development of the patch test.

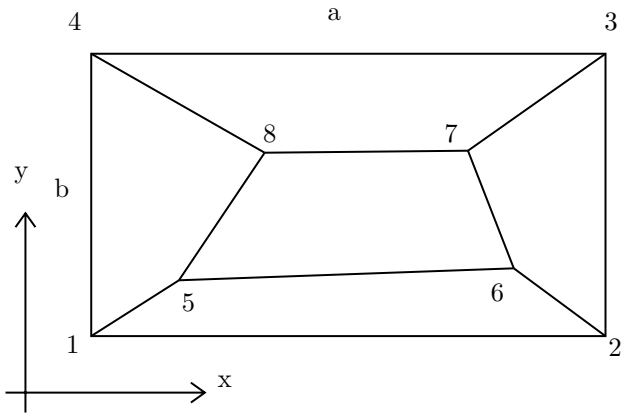


Figure 5.1: Patch test for plane elements. a = 0.24; b = 0.12; thickness t = 0.001; Young's modulus $E = 1.0 \times 10^6$; Poisson ratio $\nu = 0.25$. Boundary conditions: $u = 10^{-3}(x + y/2)$, $v = 10^{-3}(x/2 + y)$. Theoretical solution: $\sigma_x = \sigma_y = 1600$, $\tau_{xy} = 400$, $\varepsilon_x = \varepsilon_y = \gamma = 10^{-3}$.

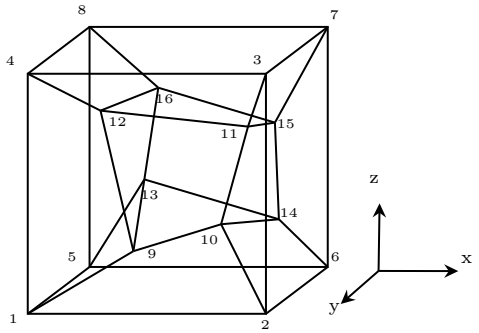


Figure 5.2: Patch test for solid elements. Outer dimension: unit cube. Young's modulus $E = 1.0 \times 10^6$; Poisson ratio $\nu = 0.25$. Boundary conditions: $u = 10^{-3}(2x + y + z)/2$, $v = 10^{-3}(x + 2y + z)/2$, $w = 10^{-3}(x + y + 2z)/2$. Theoretical solution: $\sigma_x = \sigma_y = \sigma_z = 2000$, $\tau_{xy} = \tau_{yz} = \tau_{xz} = 400$, $\varepsilon_x = \varepsilon_y = \varepsilon_z = \gamma_{xy} = \gamma_{yz} = \gamma_{xz} = 10^{-3}$.

Table 5.1: Part of important work in the development of patch test.

1966	Irons: Introduction of patch test.
1971	Babuska: Inf-sup condition for mixed formulation.
1974	Veubeke: Variational interpretation of patch test.
1980	Stummel: Limitations of patch test.
1986	Taylor: Patch test and convergence.
1989	White & Abel: Shell finite elements testing.
1993	Chapelle & Bathe: Inf-sup test.
1995	Felippa & Haugen: Individual element tests.
1997	Zienkiewicz & Taylor: Patch test revisited.
1998	Zhang: Patch test and FEM convergence.

The patch test's evolution from a basic validation tool to a pivotal element in finite element methods underscores its crucial role in computational mechanics. Initially emphasized by De Veubeke (1974) for developing non-conforming elements, Taylor, Beresford, and Wilson (1976) and de Arantes e Oliveira (1977) extended its use to improve stress analysis and ensure convergence. These studies demonstrated its effectiveness in reducing errors in complex element configurations.

Further advancements by B. M. Irons and Razzaque (1972) and B. Irons and Loikkanen (1983) addressed limitations by introducing alternative approaches for reduced integration and derivative smoothing. The patch test was also applied to Babuska-Brezzi conditions by O. C. Zienkiewicz, Qu, Taylor, and Nakazawa (1986), handling instabilities in incompressible and mixed displacement-strain formulations.

In summary, the patch test has significantly enhanced theoretical and practical understanding in finite element analysis, confirming its indispensable role in ensuring stability, consistency, and convergence of solutions. Research into its application, especially concerning inf-sup conditions and mixed formulations, remains active (Bathe, 2001; Bertrand & Boffi, 2024; W. Chen, Wang, & Zhao, 2009; X. Chen & Hisada, 2006; Ortner, 2012; Srinivasan, Dattaguru, & Singh, 2020). The tool continues to drive advances in element design and system behavior simulation. Ongoing research aims to improve its implementation in three-dimensional problems and refine its capabilities for modern computational demands. This will be further explored in Chapter 6, focusing on the synergy between numerical integration and patch test effectiveness in finite element simulations.

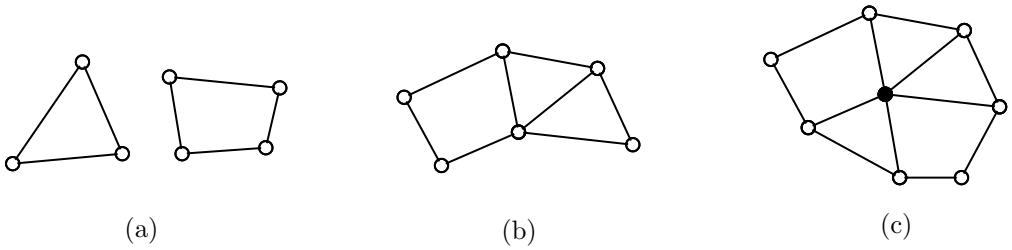


Figure 5.3: Different types of patch: (a) single element patches; (b) multi-element patch; (c) fully connected patch

5.2 Consistency and stability assessment

A patch consists of one or more finite elements sharing common interfaces. A fully connected patch includes at least one fully connected vertex node. Even a single element qualifies as a patch. Figure 5.3 illustrates various types of patches, including common triangular and quadrilateral elements.

To guarantee convergence, the approximation must satisfy both *consistency* and *stability* conditions (Ralston, 1965):

5.2.1 Consistency assessment

The initial phase of the patch test evaluates consistency. This involves ensuring the approximation accurately reproduces fundamental responses under all relevant boundary conditions across all patches. While it's impractical to validate every solution and patch, the selected test should convincingly demonstrate compliance across all scenarios.

For a small region about twice the size of element h, the unknown function $\boldsymbol{\phi}$ can be expanded into a Taylor series around a point a at the origin. To ensure convergence of the function and its first derivatives in typical second-order equations in two dimensions, the series must include terms up to $p \geq 2$:

$$\boldsymbol{\phi} = \boldsymbol{\phi}_a + \frac{\partial \boldsymbol{\phi}}{\partial x}\bigg|_a x + \frac{\partial \boldsymbol{\phi}}{\partial y}\bigg|_a y + \cdots + O\left(h^p\right) \tag{5.1a}$$

$$\frac{\partial \boldsymbol{\phi}}{\partial x} = \frac{\partial \boldsymbol{\phi}}{\partial x}\bigg|_a + \cdots + O\left(h^{p-1}\right) \tag{5.1b}$$

$$\frac{\partial \boldsymbol{\phi}}{\partial y} = \frac{\partial \boldsymbol{\phi}}{\partial y}\bigg|_a + \cdots + O\left(h^{p-1}\right) \tag{5.1c}$$

As the mesh size *h* approaches zero, the finite element approximation should exactly reproduce the posed problem for any linear forms of $\boldsymbol{\phi}$. Therefore, the consistency requirement ensures that the approximation in Equation (A.7) converges to the exact differential equation as the element size approaches zero:

$$\mathcal{L}\boldsymbol{u} + \boldsymbol{b} = 0 \tag{5.2}$$

where $\mathcal{L}$ is a differential operator and $\boldsymbol{b}$ is the vector of external forces or source terms. The corresponding boundary conditions are also satisfied, as given by:

$$\mathcal{B}\boldsymbol{u} = 0 \tag{5.3}$$

where $\mathcal{B}$ is a boundary operator.

5.2.2 Stability assessment

The stability test ensures that the solution to the discrete equation system (A.7) is unique and free from spurious mechanisms, which can corrupt the solution regardless of the element size. For any given patch, the matrix of Equation (A.7) simplifies to:

$$\boldsymbol{K}^p \tilde{\boldsymbol{u}}^p = \boldsymbol{P}^p. \tag{5.4}$$

This process includes considering relevant boundary conditions. Assuming these conditions are sufficient (e.g., eliminating rigid body modes), the patch equations must be solvable. Solvability is verified by computing the eigenvalues of $\boldsymbol{K}^p$ to ensure there are no spurious zero eigenvalues. It's unnecessary to perform a complete eigenvalue test for every assembly. For structural or irreducible displacement elements, it is enough to ensure that any single element does not have excess zero eigenvalues beyond those associated with rigid body modes. Thus, zero eigenvalues will not appear in assemblies of two or more elements. This procedure applies to any type of element, including mixed ones.

Consider a patch of elements as shown in Figures 5.1 and 5.2. By fixing the minimum necessary parameters in $\tilde{\boldsymbol{u}}^p$ to ensure a physically valid solution—such as eliminating rigid body motion in elasticity or setting a constant temperature in heat conduction—we solve for the remaining $\tilde{\boldsymbol{u}}$ values. These solutions are then compared with a presumed exact base solution. This approach detects any singularity in the $\boldsymbol{K}$ matrix, ensuring the validity of the finite element model under test conditions.

5.3 Remarks on patch tests

Before discussing specific applications of the patch test, note that passing all components of the patch test is necessary and sufficient for model convergence. However, in many practical applications, convergence can be slow, requiring many elements for precise results. Satisfying the consistency condition ensures that each primary variable ϕ_i satisfies

$$|\phi_i - \hat{\phi}_i| \leq C_i h^{p+1} \tag{5.5}$$

as h approaches zero, with p representing the polynomial order used in the solution. Higher order polynomials in element construction often enhance convergence rates, quantifiable through higher-order patch tests. Additional insights are discussed in (O. C. Zienkiewicz et al., 2013).

5.4 Standard tests

To evaluate the cubature rules and ensure accurate finite element (FE) simulations, we used standard tests based on widely recognized benchmarks from (Macneal & Harder, 1985) and (Rao & Shrinivasa, 2001). These benchmarks are crucial for validating new element formulations.

The precision of cubature rules in each test is quantified by the relative error, defined as:

$$e_r = \frac{d_t - d_e}{d_e} \times 100\% \tag{5.6}$$

where d_e is the expected displacement at a characteristic point, and d_t is the test-calculated displacement. The expected displacement is obtained using a highly refined mesh, subdividing each original element into 10,000 smaller elements, with Abaqus quadratic serendipity elements and full integration for enhanced accuracy.

To streamline the presentation of loading and boundary conditions, a notation A/B is used, where A specifies the loading conditions and B specifies the boundary conditions. The following abbreviations are used for loading conditions:

- S: Shear load
- IS: In-plane shear
- OS: Out-of-plane shear
- T: Tension
- C: Concentrated force
- IC: In-plane concentrated force

Table 5.2: Summary of standard tests for quadrilateral elements

Test number	Name	Load	Boundary
1	Regular beam	IS	F
3	Regular beam	T	F
4	Parallelogram beam	IS	F
6	Parallelogram beam	T	F
7	Trapezoidal beam	IS	F
9	Trapezoidal beam	T	F
10	Axially loaded plate	IC	F
11	Curved beam	IS	F

- OC: Out-of-plane concentrated force
- U: Uniformly distributed load
- SW: Self-weighted load

For boundary conditions, the following abbreviations are used:

- F: Fixed end
- C: Clamped boundary
- S: Simply supported boundary
- FB: Free boundary
- SYM: Symmetric boundary

In this notation, a curved beam subjected to in-plane shear load with fixed ends is represented as IS/F. Here, "IS" indicates the in-plane shear loading condition, and "F" denotes fixed boundary conditions at the ends of the beam. This concise representation simplifies the description of complex loading and support scenarios in structural analysis.

5.4.1 Tests for quadrilateral elements

A series of tests have been selected to evaluate the performance of cubature rules for quadrilateral elements in real finite element (FE) simulations. The details of boundary conditions and load conditions for these tests are summarized in Table 5.2. The geometrical and material properties, such as dimensions, Young's Modulus, and Poisson's ratio for each test, illustrated in Figures 5.4 to 5.6. The tests include three types of geometries: straight cantilever beams (Tests 1, 3, 4, 6, 7, 9), plates (Test 10), and curved beams (Test 11). All tests were performed using ABAQUS, utilizing the user-defined subroutine UEL to implement the behavior of the newly developed elements.

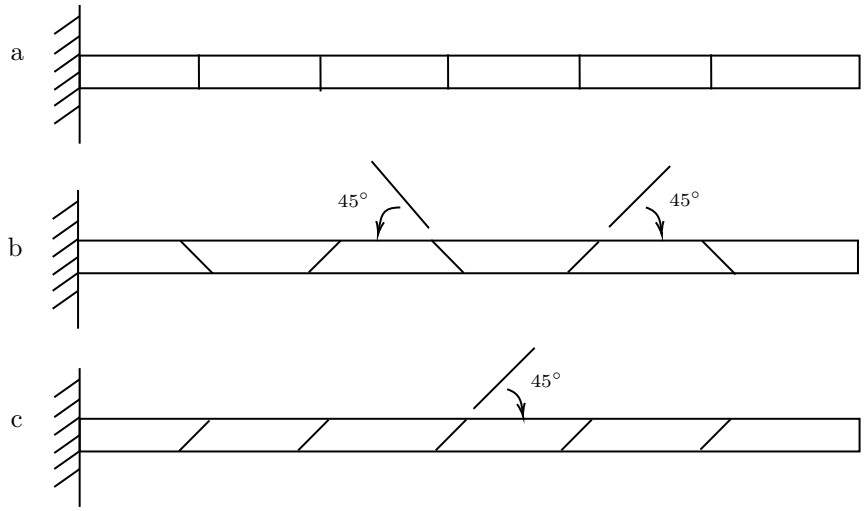


Figure 5.4: Test 1-9: straight cantilever beam with regular, trapezoidal and parallelogram elements. Length =6.0; width =0.2; depth=0.1; Young's modulus $E = 1.0 \times 10^7$; Poisson ratio $\nu = 0.3$; mesh=6×1. Load is unit force at free end.

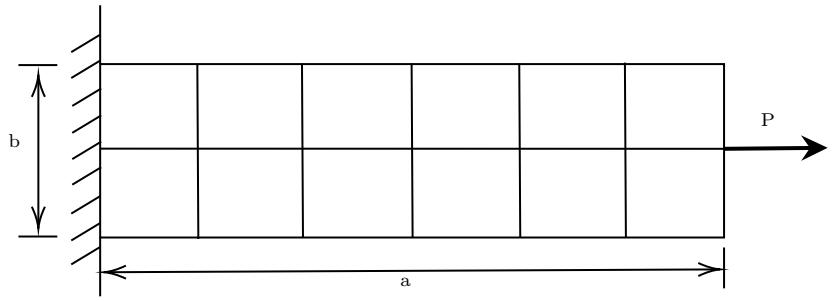


Figure 5.5: Test 10: axially loaded plate. Load $P = 1$, thickness t = 0.1, Young's modulus $E = 1 \times 10^7$, Poisson ratio $\nu = 0.2$.

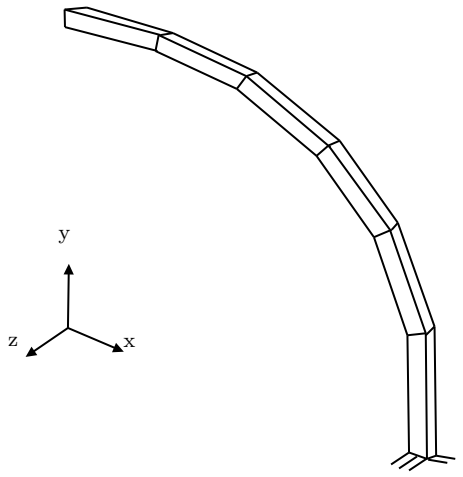


Figure 5.6: Test 11-12: curved beam. Inner radius = 4.12; outer radius = 4.32; arc = $90°$; thickness = 0.1; Young's modulus $E = 1.0 \times 10^7$; Poisson ratio $\nu = 0.25$; mesh=6×1. Load is unit force at tip.

Table 5.3: Summary of standard tests for hexahedral element

Test number	Name	Load	Boundary
1	Regular beam	IS	F
2	Regular beam	OS	F
3	Regular beam	T	F
4	Parallelogram beam	IS	F
5	Parallelogram beam	OS	F
6	Parallelogram beam	T	F
7	Trapezoidal beam	IS	F
8	Trapezoidal beam	OS	F
9	Trapezoidal beam	T	F
11	Curved beam	IS	F
12	Curved beam	OS	F
13	Twisted beam	IS	F
14	Twisted beam	OS	F
15	Rectangular plate aa	OC	C
16	Rectangular plate aa	OC	S
17	Rectangular plate aa	U	C
18	Rectangular plate aa	U	S
19	Rectangular plate ab	OC	C
20	Rectangular plate ab	OC	S
21	Rectangular plate ab	U	C
22	Rectangular plate ab	U	S
23	Scordelis-Lo roof	SW	S
24	Pinched cylinder	C	C
25	Holed semi-sphere	C	FB

5.4.2 Tests for hexahedral elements

To assess the effectiveness of cubature rules for hexahedral elements, a series of standard tests has been selected. Table 5.3 summarizes these tests, while Figures 5.4 to 5.11 illustrate the geometrical configurations and material properties, such as size, Young's Modulus, and Poisson's ratio, for each test case. These tests encompass a range of structures including straight cantilever beams (Tests 1-9), curved beams (Tests 11-12), twisted beams (Tests 13-14), plates with varying dimensions and meshes (Tests 15-22), the Scordelis-Lo roof (Test 23), a pinched cylinder (Test 24), and a holed semi-sphere (Test 25).

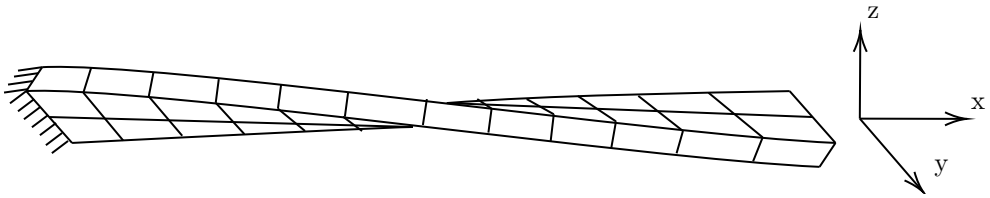


Figure 5.7: Test 13-14: twisted beam. Length = 12.0; width = 1.1; depth = 0.32; twist = 90° from root to tip; Young's modulus $E = 29.0 \times 10^6$; Poisson ratio $\nu = 0.22$; mesh=12×2. Load is unit force at tip.

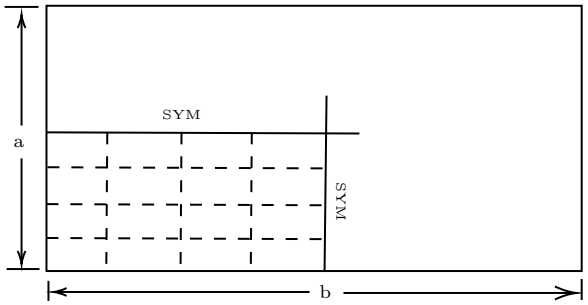


Figure 5.8: Test 15-22: rectangular plate. a = 2.0; b = 2.0 or 10.0; thickness t = 0.01; Young's modulus $E = 1.7472 \times 10^7$; Poisson ratio $\nu = 0.3$; boundary conditions = simply supported or clamped; mesh=$N \times N$. Load is uniform pressure $q = 10^{-4}$ or central force $P = 4.0 \times 10^{-4}$.

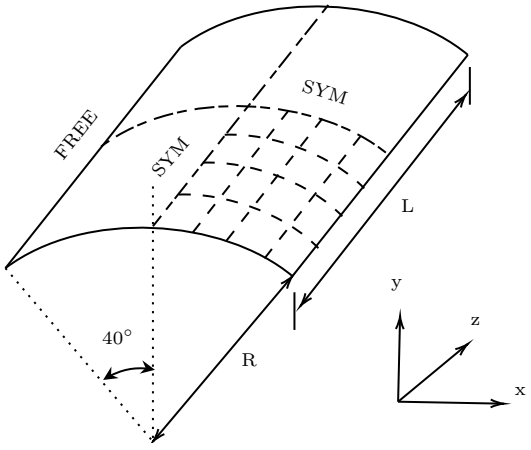


Figure 5.9: Test 23: Scordelis-Lo roof. Radius $R = 25$; Length $L = 50.0$; thickness $t = 0.25$; Young's modulus $E = 4.32 \times 10^8$; Poisson ratio $\nu = 0.01$; boundary conditions: $u = v = 0$ for curved sides and free for straight sides. Load is self-weight: 90/area.

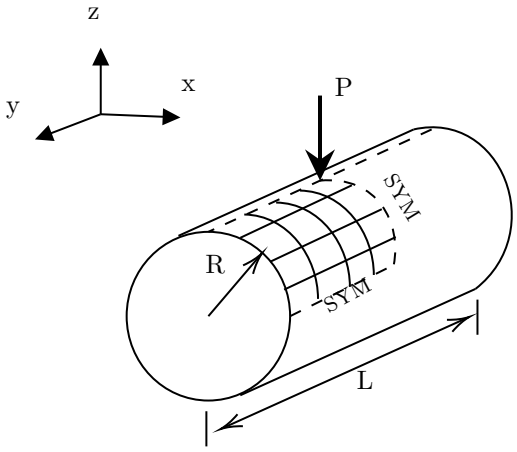


Figure 5.10: Test 24: pinched cylinder. Radius $R = 300$; Length $L = 600.0$; thickness $t = 3.0$; Young's modulus $E = 3.0 \times 10^7$; Poisson ratio $\nu = 0.3$; boundary conditions: rigid end. Load is central load $P = 1$.

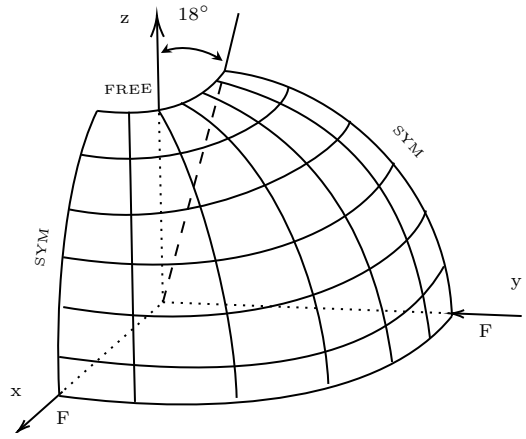


Figure 5.11: Test 25: holed semi-sphere. Radius $R = 10.0$; thickness $t = 0.04$; Young's modulus $E = 6.825 \times 10^7$; Poisson ratio $\nu = 0.3$; mesh = 6×6 (on quadrant). Load is concentrated force $F = 2$.

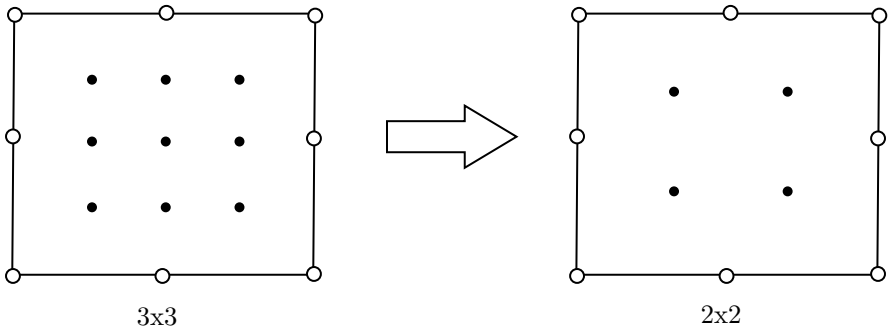


Figure 5.12: Full and reduced integration for 8-node quadrilateral element.

5.5 Reduced integration and spurious modes

5.5.1 Full and reduced integration

To understand reduced integration, it is essential to first define full integration. Bathe (1996) describes full numerical integration in displacement-based or mixed finite element formulations as using an integration order that accurately reproduces exact matrices for geometrically undistorted elements. For Gaussian-type product rules, an integration order lower than that of full integration in each direction is termed reduced integration. Figure 5.12 illustrates a reduced integration product rule for an 8-node quadrilateral element. When a lower integration order is applied selectively to specific terms or directions, it is known as selective reduced integration. Reduced integration is not applicable to triangular elements, as it does not enhance accuracy in these cases.

The primary advantage of reduced integration is the decreased computation time due to fewer multiplications needed to compute an element's stiffness matrix. For commonly used product rules, this efficiency is quantified by the formula:

$$r = \left(\frac{p+1}{p} \right)^d \tag{5.7}$$

where p represents the number of integration points per direction, and d is the problem's dimensionality. These efficiency improvements can be substantial. In nonlinear analysis, evaluating stress components at integration points consumes significant time. Reducing from four integration points to one would be a major accomplishment. Reduced integration can also address locking problems, such as shear and dilation locking, caused by interpolation failure (Barlow, 1989). Since Barlow's work (Barlow, 1976), it has been recognized that strains computed with reduced integration are often more accurate than those computed with full integration.

5.5.2 Spurious zero energy modes

A notable disadvantage of reduced integration is the potential decrease in integration accuracy. As Barlow discusses, reduced integration can yield more accurate strains in configurations like eight-node quadrilaterals and twenty-node bricks, but this may be offset by the overall reduction in integration accuracy. Additionally, a significant drawback is the emergence of spurious zero energy modes (SZEM), which can severely impact the solution's reliability.

SZEMs are particular solutions or eigenvectors of the stiffness matrix, represented as:

$$\boldsymbol{K}\boldsymbol{\phi}_S = 0 \tag{5.8}$$

where $\boldsymbol{K}$ is the stiffness matrix and $\boldsymbol{\phi}_S$ denotes a SZEM. Differing from a rigid body mode $\boldsymbol{\phi}_0$, the modal strain state for a SZEM is given by

$$\boldsymbol{\varepsilon}_S = \boldsymbol{B}\boldsymbol{\phi}_S. \tag{5.9}$$

This strain state does not vanish across the entire domain but evaluates to zero at integration points, satisfying Equation (5.8). SZEM arises when the number of strain evaluations at integration points is fewer than the independent strain states determined by the nodal displacements, as shown in inequality:

$$N_s \times N_p \leq N_d \times N_n - N_r \tag{5.10}$$

where N_s is the number of strain state, N_p is the number of integration points, N_d is the number of dimension, N_n is the number of nodes and N_r is the number of rigid body modes.

In this scenario, specific linear combinations of strain states result in zero strains at all integration points. The mismatch between independent strain states and strain evaluations at integration points sets a lower bound for the emergence of SZEMs. While additional spurious modes may occur due to redundant strain evaluations, this fundamental mismatch is a key predictor. Figure 5.13 illustrates a rigid body mode compared with a SZEM, highlighting these concepts visually.

SZEMs occurring within the fields of elements are often termed "communicating" modes. Figure 5.14 provides an illustration of a rigid body mode alongside an SZEM within these element fields.

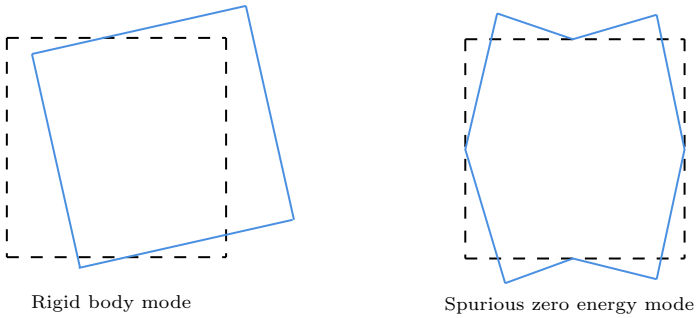


Figure 5.13: A rigid body mode and a SZEM. Black dash line: before deformation; Blue line: after deformation.

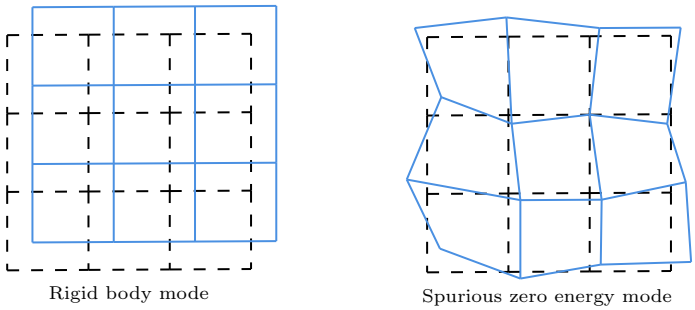


Figure 5.14: Rigid body mode and SZEM in element fields.

A spurious mode that propagates uniquely across the entire field is defined as a Global Spurious Zero Energy Mode (GSZEM). Figure 5.15 depicts an example. The mode shape in one element sets the mode shapes of adjoining elements to ensure displacement continuity, causing a chain reaction that sequentially determines the mode shapes of all elements in the system.

Conversely, multiple instances of a spurious mode are designated as Local Spurious Zero Energy Modes (LSZEMs). SZEMs that remain isolated within element fields and do not propagate are termed non-communicating modes. Figure 5.16 illustrates an example of an LSZEM.

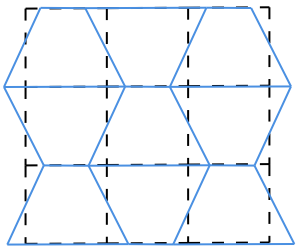


Figure 5.15: A global spurious zero energy mode.

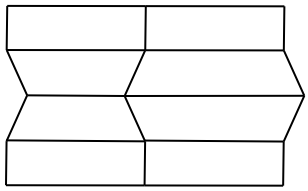


Figure 5.16: A local spurious zero energy mode.

The phenomenon of "near mechanism," attributed to LSZEMs, was investigated by (Bićcanić & Hinton, 1979). This study demonstrated that reduced integration can significantly impair the performance of four-node membrane elements, even when restrained. Figures 5.17 to 5.20 display the lowest six eigenvectors for a cantilever beam using both four- and eight-node elements, illustrating that reduced integration in four-node elements results in inferior performance.

A spurious mode that exists within a single element but disappears in a field of elements is categorized as a non-communicating mode. For example, the SZEM in Figure 5.13 is non-communicable because two adjacent elements cannot simultaneously exhibit this mode, causing it to vanish in the assembled configuration. An assembly of two or more elements will not exhibit this spurious mode. Proper boundary conditions can also eliminate SZEMs.

The presence or absence of spurious modes depends on factors like element formulations, numerical integration methods, mesh quality, and problem setup. GSZEMs often arise during the assembly of the global stiffness matrix, which integrates contributions from multiple elements. Even if individual elements perform well, their assembly can produce global spurious modes.

5.6 Summary

In this chapter, a comprehensive framework is presented for evaluating the performance of new elements in FE simulations. The patch test is emphasized as an essential tool for assessing the consistency and stability of finite elements under various boundary conditions. This introduction explains the nature of patches, the execution of patch tests, and their role in verifying element formulation correctness. Additionally, standard tests for quadrilateral and hexahedral elements are detailed, validating finite element performance across intricate geometries and conditions.

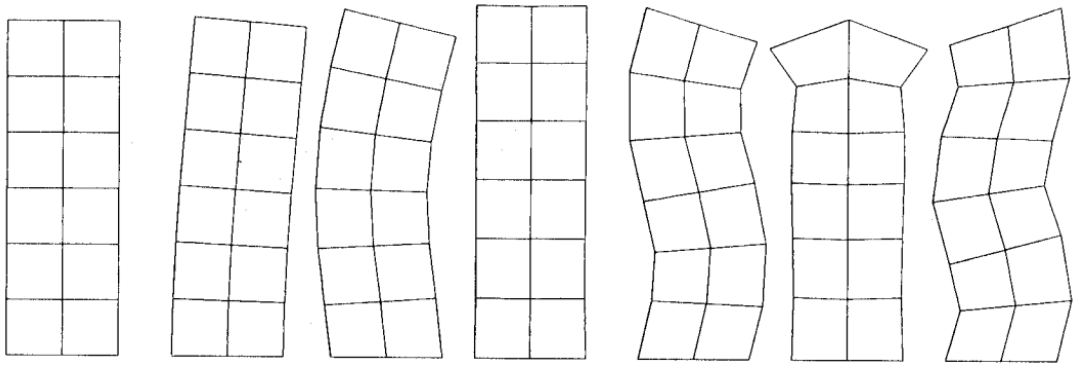


Figure 5.17: Lowest six eigenvectors for cantilever beam with four-node elements and full integration.

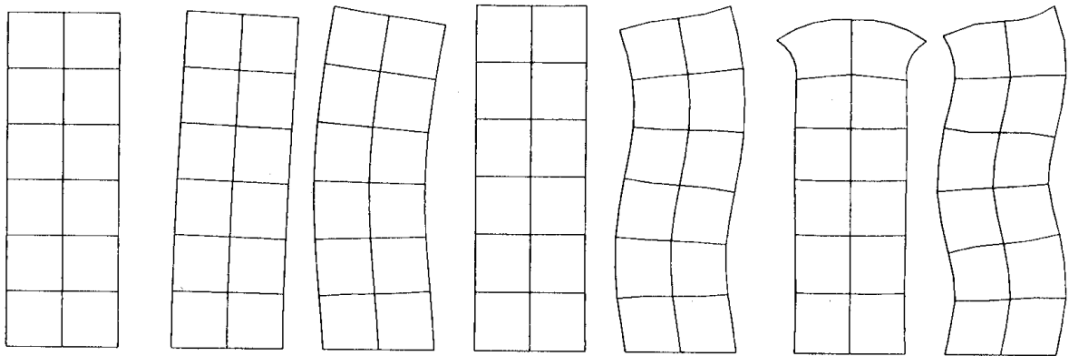


Figure 5.18: Lowest six eigenvectors for cantilever beam with eight-node elements and full integration.

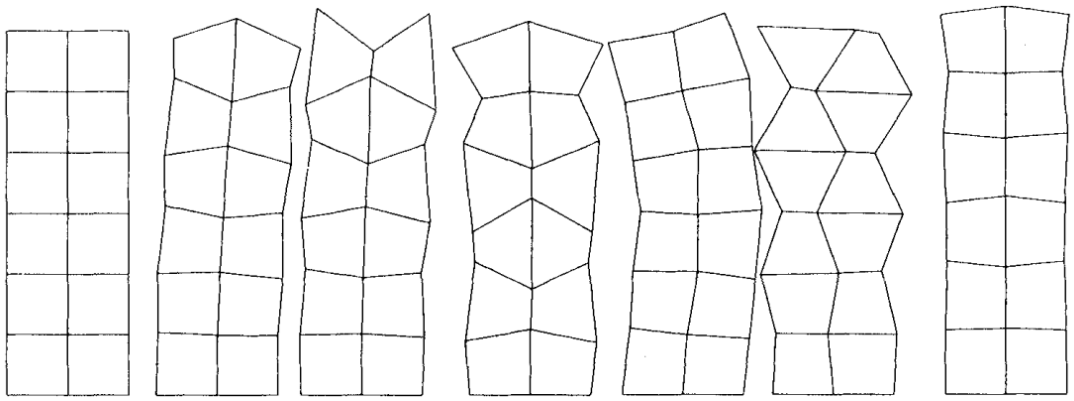


Figure 5.19: Lowest six eigenvectors for cantilever beam with four-node elements and reduced integration.

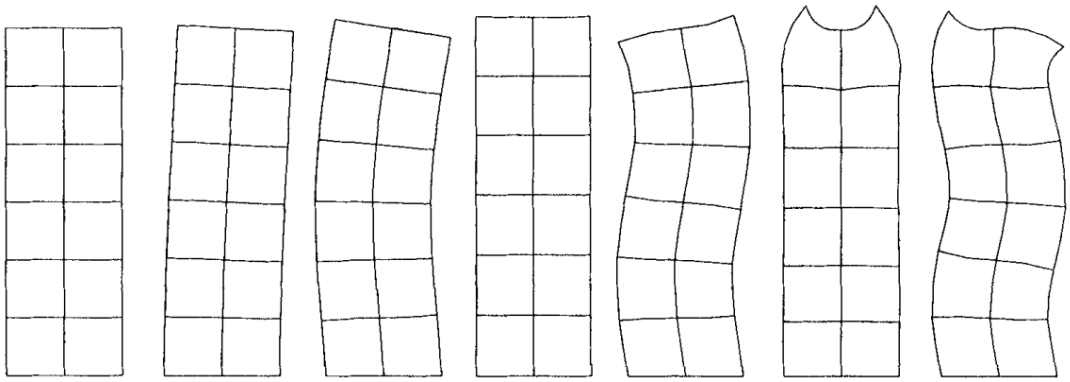


Figure 5.20: Lowest six eigenvectors for cantilever beam with eight-node elements and reduced integration.

In addition, we discuss the accuracy of integration methods in finite elements, contrasting full and reduced integration techniques and their impact on computational efficiency and error control. We also explore spurious zero energy modes—undesirable artifacts of reduced integration that compromise solution accuracy. This section underscores the balance between computational efficiency and result precision in engineering simulations, highlighting the nuanced decisions involved in finite element method applications.

Chapter 6

Cubature Rules for Finite Element Method passing the Patch Test

6.1 Criteria to pass patch test

The essential requirement in a patch test is that the theoretical solution is precisely replicated within the patch, as detailed in Section 5.2.2. This involves accurately calculating strains and stresses from nodal displacements and ensuring that forces on interior nodes are in equilibrium. Specifically, the forces due to displacements imposed on interior nodes must cancel out, satisfying the condition:

$$\sum_{e} \boldsymbol{F}_{\boldsymbol{i}}^{\boldsymbol{e}} = 0. \tag{6.1}$$

Here, $\boldsymbol{F}_{\boldsymbol{i}}^{\boldsymbol{e}}$ represents the vector of generalized forces exerted by the element on a node i. Thus, fulfilling Equation (6.1) necessitates exact computation of the integral in Equation (A.23).

According to (MacNeal, 1994), the necessary condition for conforming elements derived from a presumed displacement field to satisfy the patch test includes:

1) Strains computed from nodal displacements are correct to the desired order in Equations (5.1b) and (5.1c).
2) The integral
$$\boldsymbol{f} = \int_{V_e} \boldsymbol{B}^T \sigma dV \tag{6.2}$$
relating the generalized forces on nodes to stresses is exact for the desired order of σ.

6.2 Integration requirement

In the context of a constant-stress patch test, the equilibrium condition (6.2) can be expressed as:

$$\boldsymbol{f} = \int_{V_e} \boldsymbol{B}^T dV \boldsymbol{\sigma} \tag{6.3}$$

where the matrix $\boldsymbol{B}$ comprises elements from the gradient of displacement shape functions $\nabla_{\boldsymbol{x}} \boldsymbol{N}$. For instance, in a two-dimensional setting, $\boldsymbol{B}^T$ takes the form:

$$\boldsymbol{B}^T = \begin{bmatrix} \partial_x \boldsymbol{N} & \cdot & \partial_y \boldsymbol{N} \\ \cdot & \partial_y \boldsymbol{N} & \partial_x \boldsymbol{N} \end{bmatrix}. \tag{6.4}$$

The accuracy of this integration, crucial for the integrity of the solution, depends on the exact integration of:

$$\boldsymbol{g} = \int_{V_e} \nabla_{\boldsymbol{x}} \boldsymbol{N} dV \tag{6.5}$$

by the selected cubature rule. Typically, finite element analyses employ polynomial cubature rules, which are designed to integrate a specific set of polynomials exactly. These rules are often characterized as either total-degree or product rules, where a total-degree rule of degree d precisely integrates all polynomials up to a total degree d, and a product rule of degree d does so for each variable independently. MacNeal (1994) provides guidelines for the minimum number of integration points required by product rules for different elements.

6.2.1 Parametric elements

In the framework of finite element analysis, parametric elements are described in a parametric space, designated by parent elements. Here, the displacement shape functions are functions of parametric coordinates $\boldsymbol{\xi}$. Concurrently, the geometric representation of the element is provided in a parametric format where Cartesian coordinates are interpolated through the Equation (A.25) for example. In this context, $\boldsymbol{N}'$ represents the geometric shape functions that are also dependent on the natural coordinates $\boldsymbol{\xi}$.

Transforming the domain of integral (6.5) into parametric space V', we can express it as:

$$\boldsymbol{g} = \int_{V_e'} \det(\boldsymbol{J}) \nabla_{\boldsymbol{x}} \boldsymbol{N} dV'. \tag{6.6}$$

Here, $\det(\boldsymbol{J})$ represents the determinant of the Jacobian matrix $\boldsymbol{J} = \nabla_{\boldsymbol{\xi}} \boldsymbol{x}$, which is the gradient of the Cartesian coordinates with respect to the natural coordinates. By employing the chain rule, the integrand in (6.6) simplifies to:

$$\bar{\boldsymbol{g}} = \det(\boldsymbol{J}) \nabla_{\boldsymbol{x}} \boldsymbol{N} = \det(\boldsymbol{J}) \left(\nabla_{\boldsymbol{\xi}} \boldsymbol{\phi}\right) \boldsymbol{J}^{-1} = \nabla_{\boldsymbol{\xi}} \boldsymbol{N} \operatorname{adj}(\boldsymbol{J}) = \nabla_{\boldsymbol{\xi}} \boldsymbol{N} \operatorname{adj}\left(\boldsymbol{X} \nabla_{\boldsymbol{\xi}} \boldsymbol{N}'\right) \tag{6.7}$$

where $\operatorname{adj}(\,)$ denotes the adjoint of a matrix. This formulation aids in accurately transforming derivatives from the parametric space to the Cartesian coordinate system.

6.2.2 Polynomial interpolation

In the common scenario where both the displacement shape functions $\boldsymbol{N}$ and the geometry shape functions $\boldsymbol{N}'$ are polynomials of the natural coordinates $\boldsymbol{\xi}$, the expression derived in Equation (6.7) indicates that the integrand $\bar{\boldsymbol{g}}$ is also polynomial in $\boldsymbol{\xi}$.

The shape functions $\boldsymbol{N}$ and $\boldsymbol{N}'$ can be expressed as linear combinations of monomials through the equations:

$$\boldsymbol{N} = \boldsymbol{M} \boldsymbol{v} \tag{6.8}$$

$$\boldsymbol{N}' = \hat{\boldsymbol{M}} \hat{\boldsymbol{v}} \tag{6.9}$$

where $\boldsymbol{v}$ and $\hat{\boldsymbol{v}}$ represent vectors of monomials, and $\boldsymbol{M}$ and $\hat{\boldsymbol{M}}$ are matrices of coefficients that are typically square and invertible in the context of finite element applications. Consequently, Equation (6.7) can be reformulated as:

$$\bar{\boldsymbol{g}} = \boldsymbol{M} \nabla_{\boldsymbol{\xi}} \boldsymbol{v} \operatorname{adj}\left(\boldsymbol{X} \hat{\boldsymbol{M}} \nabla_{\boldsymbol{\xi}} \hat{\boldsymbol{v}}\right). \tag{6.10}$$

In two-dimensional parametric space, equation (6.10) can be represented using index notation as follows:

$$\bar{g}^{(2)}_{pj} = \sum_{q,s,t} \left(\varepsilon_{kl} \frac{\partial v_s}{\partial \xi_k} \frac{\partial \hat{v}_t}{\partial \xi_l} \right) \varepsilon_{jd} M_{ps} \hat{M}_{qt} X_{dq}. \tag{6.11}$$

Here, ε_{kl} denotes the Levi-Civita symbol. Given the arbitrariness of the nodal coordinates $\boldsymbol{X}$ and the invertibility of the matrices $\boldsymbol{M}$ and $\hat{\boldsymbol{M}}$, it is essential to ensure that the polynomial in the parentheses is integrated precisely for the finite element integration to be accurate.

Using multi-index notation $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$, we express monomials as follows:

$$v_s = \prod_i \xi_i^{\alpha_{si}} = \boldsymbol{\xi}^{\boldsymbol{\alpha}_s}, \qquad \hat{v}_t = \prod_i \xi_i^{\hat{\alpha}_{ti}} = \boldsymbol{\xi}^{\hat{\boldsymbol{\alpha}}_s}. \tag{6.12}$$

The integrand then simplifies after some algebraic manipulations to:

$$h_{st}^{(2)} = \varepsilon_{kl} \frac{\partial v_s}{\partial \xi_k} \frac{\partial \hat{v}_t}{\partial \xi_l} = \cdots = \begin{vmatrix} \boldsymbol{\alpha}_s & \hat{\boldsymbol{\alpha}}_t \end{vmatrix} \boldsymbol{\xi}^{\boldsymbol{\alpha}_s + \hat{\boldsymbol{\alpha}}_t - \mathbf{1}} \tag{6.13}$$

where $\mathbf{1} = (1,1)$ represents a vector of ones and $\begin{vmatrix} \boldsymbol{\alpha}s & \hat{\boldsymbol{\alpha}}t \end{vmatrix}$ denotes the determinant of a 2×2 matrix formed from the multi-indices $\boldsymbol{\alpha}s$ and $\hat{\boldsymbol{\alpha}}t$. The term $\boldsymbol{\xi}^{\boldsymbol{\alpha}s + \hat{\boldsymbol{\alpha}}t - \mathbf{1}}$ indicates the monomials that must be exactly integrated, applicable for all combinations of $\boldsymbol{\alpha}s$ and $\hat{\boldsymbol{\alpha}}t$ where these indices are linearly independent. This results in a straightforward method for deriving the necessary monomials for integration without the need for complex symbolic operations.

6.2.3 Monomial sets

Based on the discussions in section 6.2.2, constructing monomial sets required for specific interpolations of displacement and geometry becomes straightforward. This paper focuses specifically on quadrilateral elements employing serendipity interpolation. These monomial sets are denoted using the notation $m_{i,j}^2$, where 2 represents the spatial dimension. Here, i indicates the order of the displacement interpolation, and $j \leq i$ represents the order of the geometry interpolation. This notation systematically captures the relationship between the interpolation orders and their dimensional context.

In cases of higher-order interpolation, the monomial sets expand considerably. To succinctly describe these sets, we introduce a shorthand notation $\langle m \rangle$ for each monomial set m. This notation allows for a more compact representation, facilitating easier reference and discussion within the text.

1. Each monomial is indicated by its multi-index, as the variables are always the natural coordinates $\boldsymbol{\xi}$.
2. Only multi-indices with indices in non-increasing order are given, as symmetry ensures all other permutations of indices are also included in the set.
3. Only the largest multi-indices (according to the partial ordering $\leq$) are given. For any multi-index $\boldsymbol{\alpha}$, all multi-indices $\boldsymbol{\beta} \neq \boldsymbol{\alpha}$ for which $\boldsymbol{\alpha} \leq \boldsymbol{\beta}$ are included in the set, and are thus omitted from the short form.

As an example, the monomial set

$$m_{2,2}^2 = \{1, \xi, \eta, \xi^2, \xi\eta, \eta^2, \xi^3, \xi^2\eta, \xi\eta^2, \eta^3, \xi^2\eta^2\} \tag{6.14}$$

has the short form

$$\langle m^2_{2,2} \rangle = \{(3,0),(2,2)\}. \tag{6.15}$$

For quadrilateral elements with serendipity interpolation, the first few sets are

$$\langle m^2_{1,1} \rangle = \{(1,0)\} \tag{6.16a}$$
$$\langle m^2_{2,1} \rangle = \{(2,1)\} \tag{6.16b}$$
$$\langle m^2_{2,2} \rangle = \{(3,0),(2,2)\} \tag{6.16c}$$
$$\langle m^2_{3,1} \rangle = \{(3,1)\} \tag{6.16d}$$
$$\langle m^2_{3,2} \rangle = \{(4,1),(3,2)\} \tag{6.16e}$$
$$\langle m^2_{3,3} \rangle = \{(5,0),(4,1),(3,3)\}. \tag{6.16f}$$

6.3 Cubature rules for quadrilaterals

6.3.1 Summary of theoretical background

Gaussian cubature rules approximate the scaled integral of a function $f(\boldsymbol{x})$ over a domain Ω as a weighted sum of function values at specific integration points $\boldsymbol{x}_i$:

$$\frac{\int_V f(\boldsymbol{x}) d\Omega}{\int_V d\Omega} \approx \sum_{i=1}^{N} w_i f(\boldsymbol{x}_i). \tag{6.17}$$

Requiring that equation (6.17) is exact for a specific set of functions f and number of integration points N, leads to a system of so-called moment equations, which can be solved to obtain the coordinates $\boldsymbol{x}_i$ and corresponding weights w_i of individual cubature rules.

The requirement for equation (6.17) to be exact for a specific set of functions f and a defined number of integration points N results in a system of moment equations. Solving these equations yields the coordinates $\boldsymbol{x}_i$ and corresponding weights w_i necessary for constructing precise cubature rules.

Most cubature rules described in the literature are total-degree rules, designed to exactly integrate all monomials up to a specified total degree. An alternative approach involves product rules that integrate monomials up to a given degree in each variable precisely. In this chapter, we compute cubature rules tailored to the monomial sets identified as necessary for passing the patch test, as detailed in Section 6.2.3. These specifically designed rules are expected to be more efficient, requiring fewer integration points compared to the total-degree and product rules that integrate the same sets of monomials exactly.

Table 6.1: Types of orbits for full symmetry in square

Orbit type	Generator	Condition	points	variables
0	$(0, 0)$	—	1	1
1	$(\alpha, 0)$	$\alpha \neq 0$	4	2
2	(α, α)	$\alpha \neq 0$	4	2
3	(α, β)	$\alpha \neq 0, \beta \neq 0$	8	3

To achieve exact integration of a specific monomial set, we solve a system of polynomial moment equations. This system can be inconsistent (offering no solutions), zero-dimensional (yielding a finite number of solutions), or positive-dimensional (providing an infinite array of solutions). Within finite element analysis, we focus on real solutions where all integration points $\boldsymbol{x}_i$ are within the domain Ω. While we prioritize cubature rules with all positive weights (PI quality), it is possible to use rules with some negative weights (NI quality). In this study, we employ Gröbner bases to solve the moment equations. Although this method is computationally demanding compared to iterative numerical techniques, it has the advantage of identifying all possible solutions.

In this chapter, we focus exclusively on fully symmetric cubature rules, which align the integration points and their weights with the symmetry of the domain, whether a square or a cube. These rules accurately integrate all even monomials, which are monomials where all powers are even. Consequently, we consider only the sets of even monomials, denoted as $m_{i,j}^{d*}$, derived from the complete monomial sets $m_{i,j}^{d}$. The corresponding short forms are represented as $\langle m_{i,j}^{d*} \rangle$.

In our analysis of fully symmetric cubature rules on the square, we identify integration points according to four distinct types of orbits. Table 6.1 details the number of points for each orbit type and the corresponding number of variables present in the moment equations for each orbit. The number of variables in these equations aligns with those needed to define the generator point, from which all other points in the orbit are derived through the transformations in G_{CFS} defined in Section 2.1.2. Additionally, a separate variable is included to represent the weight, which is consistent across all points within the same orbit.

6.3.2 Results for quadrilaterals

The even monomial sets corresponding to the sets in equation (6.16) are

$$\langle m_{1,1}^{2*} \rangle = \{(0,0)\} \tag{6.18a}$$

$$\langle m_{2,1}^{2*} \rangle = \{(2,0)\} \tag{6.18b}$$

$$\langle m_{2,2}^{2*} \rangle = \{(2,2)\} \tag{6.18c}$$

$$\langle m_{3,1}^{2*} \rangle = \{(2,0)\} \tag{6.18d}$$

$$\langle m_{3,2}^{2*} \rangle = \{(4,0),(2,2)\} \tag{6.18e}$$

$$\langle m_{3,3}^{2*} \rangle = \{(4,0),(2,2)\} \tag{6.18f}$$

The monomial sets $m_{1,1}^{2*}$, $m_{2,1}^{2*} = m_{3,1}^{2*}$ and $m_{3,2}^{2*} = m_{3,3}^{2*}$ include all even monomials of total degrees up to 1, 3, and 5, respectively. These sets are precisely integrated by the fully symmetric cubature rules of total degrees 1, 3, and 5, as outlined by Stroud (1971). Additionally, the set $m_{2,2}^{2*}$, encompassing all even monomials of individual degree up to 3, is accurately integrated by a 2×2 product rule of degree 3. It is notable that the set $m_{3,1}^{2}$ reaches a total degree of 4, yet contains no even terms at degree 4; similarly, $m_{3,3}^{2}$ reaches a total degree of 6 without even terms at this level. For these sets, existing cubature rules suffice, and we have identified the minimal necessary rules for each case. Table 6.2 provides the recommended integration orders for stiffness matrix computations and patch tests for quadrilateral and serendipity elements based on different orders of element geometry. For interpolation orders of 4 and higher, the monomial sets become more complex, rendering the minimal rules beyond simple total-degree or product rules.

The monomial set $m_{1,1}^{2}$ is precisely integrated using a 1-point 1×1 product rule, which corresponds to the widely recognized reduced-integration bi-linear quadrilateral element. This element exhibits two spurious zero-energy modes. Given our focus on fully symmetric cubature rules, the subsequent viable rules utilize 4 points. Consequently, employing the 2×2 product rule is logical, resulting in the conventional full-integration bi-linear element.

The most efficient cubature rule for the set $m_{2,2}^{2}$ is the 4-point 2×2 product rule, which corresponds to the well-recognized reduced-integration serendipity quadratic element. This element exhibits one spurious zero-energy mode; however, it remains practical and is commonly used because this mode does not communicate within the element field. Similarly, the set $m_{2,1}^{2}$ necessitates a 4-point rule. Although a fully symmetric 4-point rule of total degree 3 is documented by Stroud (1971), it does not provide advantages over the 2×2 product rule in terms of efficiency. For both $m_{2,1}^{2}$ and $m_{2,2}^{2}$, a 5-point rule will remove the spurious zero-energy mode.

For the cubic serendipity element, the set $m_{3,3}^{2}$ requires at least an 8-point rule of total-degree 3, denoted as $m_{3,3}^{2}|8$, which is detailed in Table 6.3 and depicted in Fig. 6.3 alongside the 9-point 3×3 product rule $m_{3,3}^{2}|9^{*}$. Product rules will henceforth be marked with the symbol * like $m_{3,3}^{2}|9^{*}$. The rule $m_{2,1}^{2}|4$ and $m_{3,1}^{2}|4$ are both four-point rules of degree 3 found by Stroud but are unsuitable for these elements as they fail to adequately model normal deformation during testing. The $m_{2,1}^{2}|4$ rule leads to GSZEM that result in abnormal deformations as illustrated in Figure 6.2. The rule $m_{3,1}^{2}|4$ induces SZEM because it satisfies the inequality shown in (5.10), causing singularity issues in the stiffness matrix.

Table 6.2: Recommended integration order of product rule for stiffness matrix and patch test in quadrilateral serendipity element with different element order and geometry

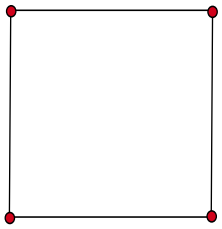
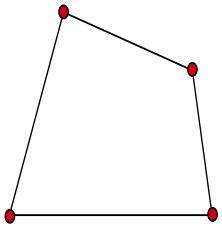
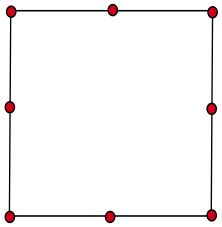
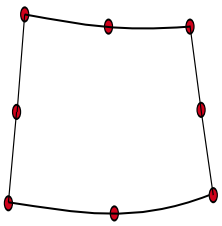
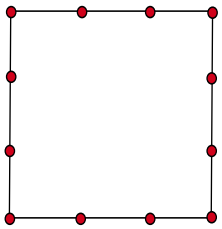
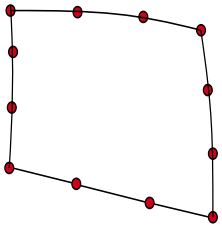
Element order	Geometry	Stiffness matrix	Patch test	Example
Linear	Undistorted	2x2	1x1	
Linear	Distorted	2x2	1x1	
Quadratic	Undistorted	3x3	2x2	
Quadratic	Distorted	3x3	2x2	
Cubic	Undistorted	4x4	2x2	
Cubic	Distorted	4x4	3x3	

Figure 6.1: Original geometry of cantilever beam

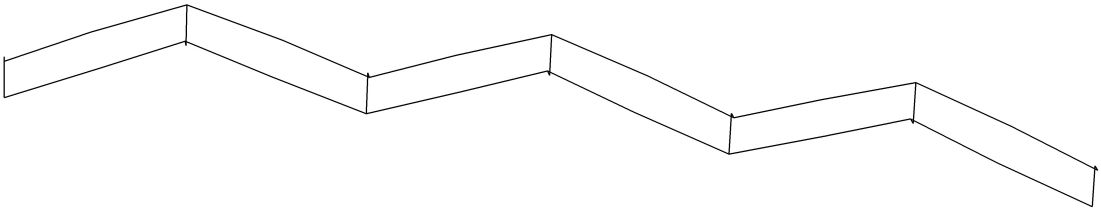


Figure 6.2: Abnormal deformation in test 1 caused by GSZEM with $m^2_{2,1}|4$.

Table 6.3: Eight-point symmetric cubature rule $m^2_{3,3}|8$ in (Stroud, 1971)

Orbit type	w	α^2
1	$40/49$	$7/15$
2	$9/49$	$7/9$

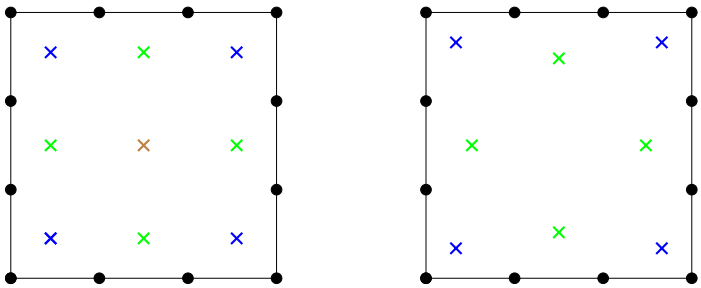


Figure 6.3: Location of integration points for cubature rules for $m^2_{3,3}|9^*$ and $m^2_{3,3}|8$ (left: 9-point product rule, right: 8-point rule)

Table 6.4: Summary of the cubature rules for quadrilateral elements and the corresponding properties.

Cubature rule	Has SZEM?	Stable?
$m^2_{1,1}\|1^*$	YES	YES
$m^2_{1,1}\|4^*$	NO	YES
$m^2_{2,1}\|4$	YES	NO
$m^2_{2,1}\|4^*$	YES	YES
$m^2_{2,2}\|4^*$	YES	YES
$m^2_{3,1}\|4$	YES	NO
$m^2_{3,3}\|8$	NO	YES
$m^2_{3,3}\|9^*$	NO	YES

The monomial set $m^2_{3,2}$ is equivalent to $m^2_{3,3}$, leading to the same cubature rules and corresponding finite elements. Similarly, $m^2_{3,1}$ mirrors the requirements of $m^2_{2,1}$; however, employing 4-point and 5-point rules for these sets results in elements burdened by an excessive number of spurious zero-energy modes, rendering them impractical. Consequently, at least an 8-point rule, specifically the $m^2_{3,3}|8$, is necessary to produce viable elements for the cubic serendipity element, regardless of the geometry interpolation used.

Table 6.4 provides a summary of the cubature rules and their properties for various monomial sets. The second column identifies whether each rule is associated with SZEM. The third column indicates the stability of each rule in finite element simulation, regardless of the rule's accuracy in integration.

6.4 Performance evaluation for quadrilaterals

While satisfying the integration requirement is necessary for passing the patch test, it alone does not guarantee the optimal numerical behavior of finite elements. To evaluate the behavior of the cubature rules detailed in Section 6.3, a selection of standard tests from section 5.4.1 has been employed to assess the performance of quadrilateral elements. Figure 6.4 presents a flowchart that outlines the process for evaluating a cubature rule.

Table 6.5 compares the relative error of the $m^2_{3,3}|8$ and $m^2_{3,3}|9^*$ elements. The $m^2_{3,3}|8$ rule shows performance similar to the $m^2_{3,3}|9^*$ product rule but with one fewer integration point. This can make the $m^2_{3,3}|8$ rule more preferable in scenarios where reducing computational cost is crucial.

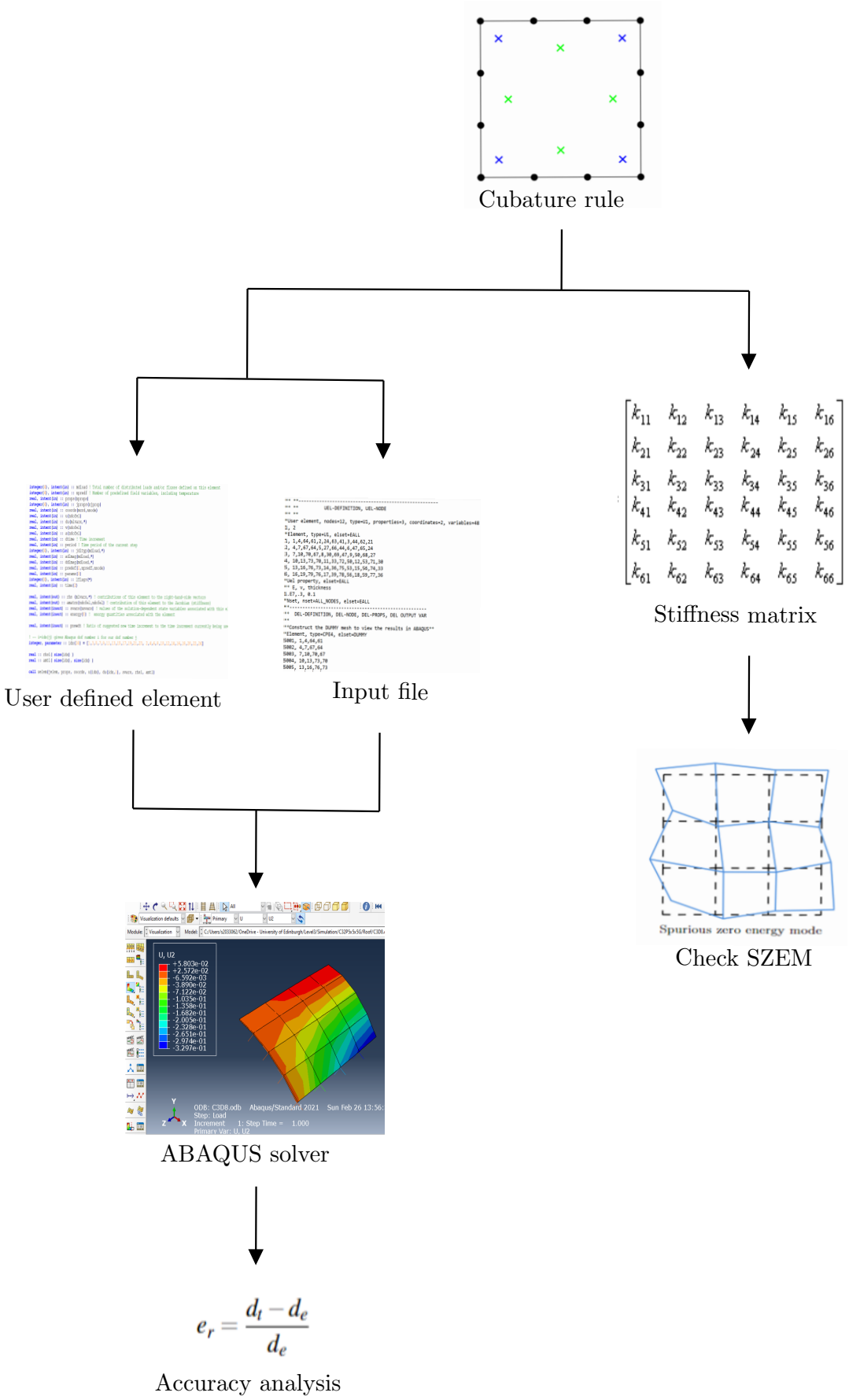


Figure 6.4: Flowchart of cubature rule analysis.

Table 6.5: Summary of standard tests and relative error of results for the 12-node cubic serendipity quadrilateral element with general geometry for rules $m^2_{3,3}|9^*$ and $m^2_{3,3}|8$

Test	Mesh	Load	d_e	$m^2_{3,3}\|9^*$	$m^2_{3,3}\|8$
1	Regular beam	S	1.06E-01	-7.57%	-7.56%
3	Regular beam	T	2.73E-05	0.00%	0.00%
4	Parallelogram beam	S	1.06E-01	-7.23%	-7.21%
6	Parallelogram beam	T	3.06E-05	-10.84%	-10.84%
7	Trapezoidal beam	S	1.07E-01	-14.32%	-14.19%
9	Trapezoidal beam	T	3.06E-05	-10.81%	-10.84%
10	Axially loaded plate	C	4.16E-06	-4.06%	-2.67%
11	Curved beam	S	8.31E-02	-0.11%	-0.09%

6.5 Hexahedral elements

6.5.1 Integration requirement

The calculations from Section 6.2.2 extend to three dimensions with a modified expression for the integrand:

$$\bar{g}^{(3)}_{pj} = \sum_{q,r,s,t,u} h^{(3)}_{stu} \varepsilon_{jde} M_{ps} \hat{M}_{qt} \hat{M}_{ru} x_{dq} x_{er} \tag{6.19}$$

where the quantity to be integrated exactly is

$$h^{(3)}_{stu} = \varepsilon_{klm} \frac{\partial v_s}{\partial \xi_k} \frac{\partial \hat{v}_t}{\partial \xi_l} \frac{\partial \hat{v}_u}{\partial \xi_m} = \cdots = \left| \boldsymbol{\alpha}_s \quad \hat{\boldsymbol{\alpha}}_t \quad \hat{\boldsymbol{\alpha}}_u \right| \boldsymbol{\xi}^{\boldsymbol{\alpha}_s + \hat{\boldsymbol{\alpha}}_t + \hat{\boldsymbol{\alpha}}_u - \mathbf{1}} \tag{6.20}$$

with $\mathbf{1} = (1,1,1)$. Therefore, in three dimensions, there is a qualitative difference in the contribution of the displacement and geometry terms compared to the two-dimensional case, where the two terms contribute equally. This difference makes elements with lower interpolation order for the geometry more attractive in three dimensions.

For hexahedral elements with serendipity interpolation, the first few monomial sets are

$$\langle m^3_{1,1} \rangle = \{(2,1,1)\} \tag{6.21a}$$
$$\langle m^3_{2,1} \rangle = \{(3,2,1)\} \tag{6.21b}$$
$$\langle m^3_{2,2} \rangle = \{(5,1,1),(4,3,1),(4,2,2),(3,3,3)\} \tag{6.21c}$$
$$\langle m^3_{3,1} \rangle = \{(4,2,1)\} \tag{6.21d}$$
$$\langle m^3_{3,2} \rangle = \{(6,2,1),(4,4,1),(5,3,2),(4,3,3)\} \tag{6.21e}$$
$$\langle m^3_{3,3} \rangle = \{(8,1,1),(7,2,1),(6,4,1),(6,3,2),(5,4,2),(4,4,4)\}. \tag{6.21f}$$

Table 6.6: Types of orbits for full symmetry in cube

Orbit type	Generator	Condition	points	variables
0	$(0, 0, 0)$	—	1	1
1	$(\alpha, 0, 0)$	$\alpha \neq 0$	6	2
2	$(\alpha, \alpha, 0)$	$\alpha \neq 0$	12	2
3	$(\alpha, \beta, 0)$	$\alpha \neq 0, \beta \neq 0$	24	3
4	(α, α, α)	$\alpha \neq 0$	8	2
5	(α, α, β)	$\alpha \neq 0, \beta \neq 0$	24	3
6	(α, β, γ)	$\alpha \neq 0, \beta \neq 0, \gamma \neq 0$	48	4

6.5.2 Cubature rules for hexahedrals

In the context of cubature on a cube, there are seven distinct types of orbits, each described in detail in Table 6.6.

The even generators corresponding to those specified in equation (6.21) are detailed as follows:

$$\langle m^{3*}_{1,1} \rangle = \{(2,0,0)\} \tag{6.22a}$$
$$\langle m^{3*}_{2,1} \rangle = \{(2,2,0)\} \tag{6.22b}$$
$$\langle m^{3*}_{2,2} \rangle = \{(4,2,2)\} \tag{6.22c}$$
$$\langle m^{3*}_{3,1} \rangle = \{(4,2,0)\} \tag{6.22d}$$
$$\langle m^{3*}_{3,2} \rangle = \{(6,2,0),(4,4,0),(4,2,2)\} \tag{6.22e}$$
$$\langle m^{3*}_{3,3} \rangle = \{(8,0,0),(6,4,0),(6,2,2),(4,4,4)\}. \tag{6.22f}$$

The multi-indices in each short form are listed in decreasing lexicographical order, with the first index, $\alpha_{1,1}$, representing the highest even power in the monomial set. Consequently, the minimal product rule necessary would have $\alpha_{1,1}/2+1$ integration points per spatial dimension. Given this, equation (6.22) suggests that product rules may be suboptimal because they integrate more monomials than strictly necessary. Therefore, our goal is to identify non-product symmetric rules, of PI quality or possibly NI quality, that require fewer points than the least extensive product rule capable of achieving the same accuracy.

The monomial set $m^3_{1,1}$ necessitates an 8-point $2\times2\times2$ product rule, unlike its two-dimensional equivalent, which only requires a 1-point 1×1 product rule. For this set, the only rules with fewer points are a 6-point PB rule, which is the minimal rule of total degree 3 with integration points at the face centers, and a one-parameter family of 7-point rules of at most NI quality.

Table 6.7: Symmetric cubature rules for $m^3_{1,1}$

Name	Points	Quality	Dimension	Orbit	w	α^2
$m^3_{1,1}\|6$	6	PB	0	1	4/3	1
$m^3_{1,1}\|7$	7	NI	1	0	-1	—
				1	3/2	8/9

These rules are summarized in Table 6.7. In cases of positive-dimensional rule families, which encompass an infinite number of solutions, only a representative example is provided. Both the 6-point and 7-point rules are known to introduce spurious zero-energy modes and typically exhibit poor performance in practical applications.

For the monomial sets $m^3_{2,1}$ and $m^3_{2,2}$, the minimal product rules required are $2 \times 2 \times 2$ and $3 \times 3 \times 3$, respectively. In both instances, there are no alternative fully symmetric cubature rules that require fewer points than these minimum product rules.

The minimal product rule for the set $m^3_{3,1}$ is a $3 \times 3 \times 3$ rule, utilizing 27 points, characterized by the structure $[1,1,1,0,1,0,0]$, where all nonzero coordinates have $\alpha^2 = 3/5$. Table 6.8 presents the recommended integration order of preduct rule for the stiffness matrix and patch test for hexahedral serendipity elements across various orders of element and geometry.

Eliminating one of the four orbits results in four different rules that have identical coordinates but different weights for the remaining orbits, as shown in Table 6.9 and Figure 6.5. These rules were initially calculated by Franke (1971) as total degree 5 rules, but here they are derived independently based on their ability to precisely integrate the set $m^3_{3,1}$, which includes the degree-6 even monomial $(4,2,0)$.

The 15-point and 19-point NI rules presented in Table 6.9 represent the minimal rules for the set $m^3_{3,1}$. Additionally, the 21-point rule is the minimal rule of PI quality, and is not unique; a second PI rule $m^3_{3,1}|21b$ also exists, with details shown in Table 6.10 and Figure 6.6. Rules with 20 to 23 points, categorized as positive-dimensional, are also listed in Table 6.10; these achieve NI quality at 20 and 21 points, and PI quality at 22 and 23 points. The 15, 19, 20, and 21c rules are known to introduce spurious zero-energy modes, leading to poor performance in elements. Conversely, the 22-point and 23-point rules, while free from spurious zero-energy modes in a mesh of elements, exhibit singularity when tested in a single element scenario.

Table 6.8: Recommended integration order of product rule for stiffness matrix and patch test in hexahedral serendipity element with different element order and geometry

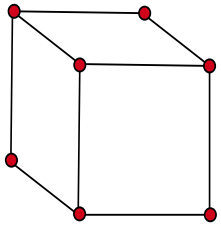
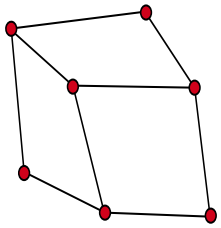
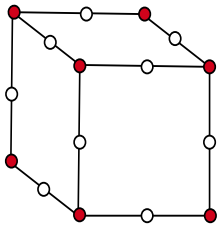
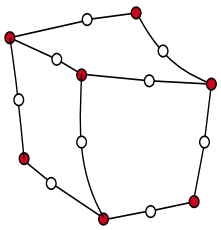
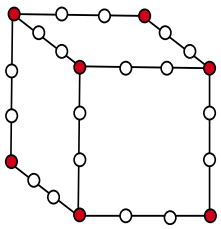
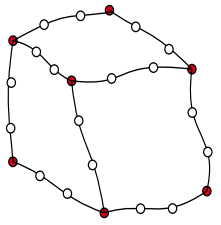
Element order	Geometry	Stiffness matrix	Patch test	Example
Linear	Undistorted	2x2x2	2x2x2	
Linear	Distorted	2x2x2	2x2x2	
Quadratic	Undistorted	3x3x3	2x2x2	
Quadratic	Distorted	3x3x3	3x3x3	
Cubic	Undistorted	4x4x4	3x3x3	
Cubic	Distorted	4x4x4	5x5x5	

Table 6.9: Summary of cubature rules for $m_{3,1}^3$ with all non-zero coordinates $\alpha^2 = 3/5$

Name	Points	w_0	w_1	w_2	w_4	Quality
$m_{3,1}^3\|15$	15	-32/81	80/81	—	25/81	NI
$m_{3,1}^3\|19$	19	168/81	-20/81	50/81	—	NI
$m_{3,1}^3\|21a$	21	128/81	—	40/81	5/81	PI
$m_{3,1}^3\|26$	26	—	64/81	8/81	21/81	PI
$m_{3,1}^3\|27^*$	27	512/729	320/729	200/729	125/729	PI

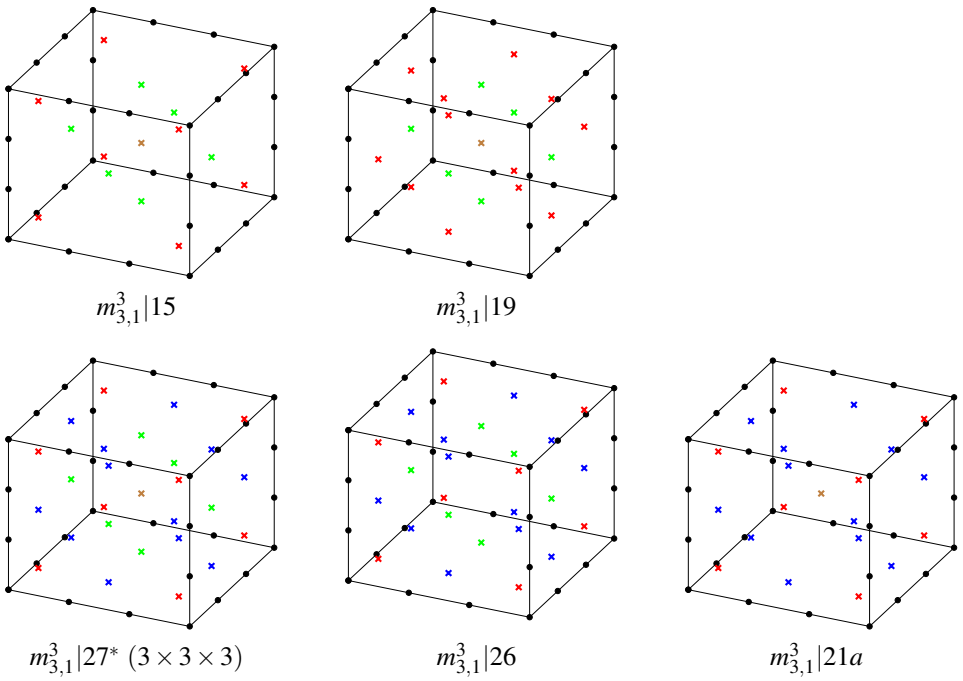


Figure 6.5: Location of integration points for the cubature rules for $m_{3,1}^3$ in Table 6.9

Table 6.10: Symmetric cubature rules for $m_{3,1}^3$. Does not include rules from Table 6.9.

Name	Points	Quality	Dimension	Orbit	w	α^2
$m_{3,1}^3\|20$	20	NI	1	1	128/81	3/4
				1	-160/243	9/10
				4	25/81	3/5
$m_{3,1}^3\|21b$	21	PI	0	0	8/5	—
				2	2/5	2/3
				4	1/5	1/3
$m_{3,1}^3\|21c$	21	NI	2	0	-38/81	—
				1	5/9	8/15
				1	4/9	2/3
				4	25/81	3/5
$m_{3,1}^3\|22$	22	PI	1	1	4/5	2/3
				4	1/5	1/3
				4	1/5	2/3
$m_{3,1}^3\|23$	23	PI	2	0	4/9	—
				1	5/9	4/5
				4	3/7	1/3
				4	25/252	4/5

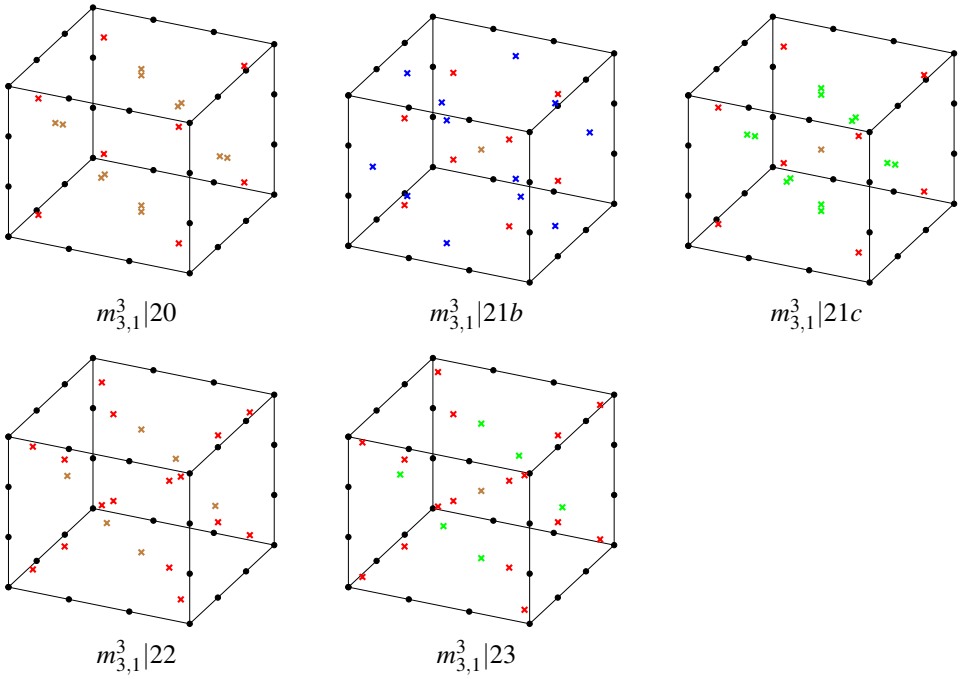


Figure 6.6: Location of integration points for the cubature rules for $m_{3,1}^3$ in Table 6.10

Table 6.11: Symmetric cubature rules for $m^3_{3,3}$

Name	Points	Quality	Dim.	Orbit	w	α^2	β^2
$m^3_{3,3}\|83$	83	PI	0	0	0.3526408581	—	—
				1	0.2237202525	0.4229513246	—
				2	0.1425196417	0.3679763044	—
				4	0.1376479994	0.2985164130	—
				4	0.0417939413	0.7414771754	—
				5	0.0698745156	0.1333132496	0.9161547004
				5	0.0617615842	0.7712307871	0.1172769993
$m^3_{3,3}\|87$	87	PI	0	0	0.3008925476	—	—
				1	0.1369148181	0.7066924737	—
				2	0.0846840676	0.7398384443	—
				2	0.2186465681	0.2580917388	—
				4	0.1130100578	0.4296380890	—
				5	0.0214047877	0.8798335790	0.4292351773
				5	0.0758273144	0.1832825313	0.8817232250

For the monomial set $m^3_{3,2}$, the minimal product rule required is a $4 \times 4 \times 4$ configuration, totaling 64 points. As the order of interpolation increases, solving the corresponding moment equations becomes progressively challenging, particularly when dealing with positive-dimensional cases where deriving PI or NI quality rules is notably difficult. In the specific case of $m^3_{3,2}$, although the moment equations are technically solvable for the 47-point structures $[1,1,0,1,2,0,0]$ and $[1,1,0,0,2,1,0]$, all obtained solutions are complex. The simplest NI rule that could be determined is a 51-point rule with the structure $[1,1,1,0,1,1,0]$.

For the monomial set $m^3_{3,3}$, the minimum product rule requires a substantial grid of $5 \times 5 \times 5$, totaling 125 points. Addressing the moment equations for this set demands considerably more computational resources than for lesser sets, although the developed rules significantly reduce the number of points required compared to the baseline product rule. Notably, the best performing PI rules identified, containing 83 and 87 points, are detailed in Table 6.11 and illustrated in Figure 6.7. It is feasible to solve the moment equations for configurations with even fewer points, such as a 76-point rule with the structure $[0,2,2,0,2,1,0]$, but these solutions often involve complex numbers, rendering them impractical. Table 6.12 provides a comprehensive summary of the cubature rules and their attributes for various monomial sets.

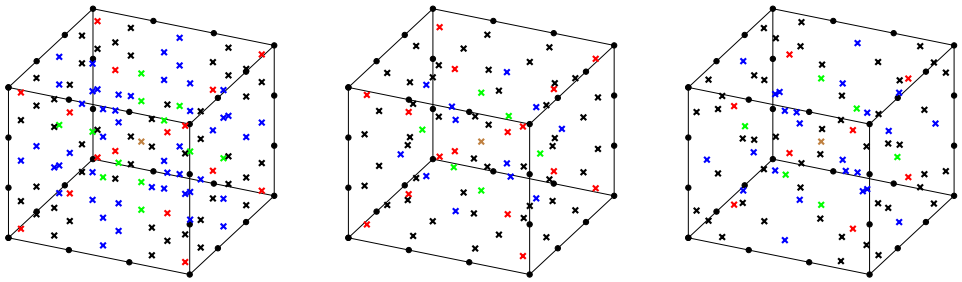


Figure 6.7: Location of integration points for cubature rules for $m^3_{3,3}$

Table 6.12: Summary of the cubature rules for hexahedral elements and the corresponding properties.

Cubature rule	Has SZEM?	Stable?
$m^3_{1,1}\|6$	YES	NO
$m^3_{1,1}\|7$	YES	NO
$m^3_{1,1}\|8^*$	NO	YES
$m^3_{2,1}\|8^*$	YES	YES
$m^3_{2,2}\|27^*$	NO	YES
$m^3_{3,1}\|15$	YES	NO
$m^3_{3,1}\|19$	YES	NO
$m^3_{3,1}\|20$	YES	NO
$m^3_{3,1}\|21a$	NO	YES
$m^3_{3,1}\|21b$	NO	YES
$m^3_{3,1}\|21c$	YES	NO
$m^3_{3,1}\|22$	NO	NO
$m^3_{3,1}\|23$	NO	NO
$m^3_{3,1}\|26$	NO	YES
$m^3_{3,1}\|27^*$	NO	YES
$m^3_{3,2}\|64^*$	NO	YES
$m^3_{3,3}\|83$	NO	YES
$m^3_{3,3}\|87$	NO	YES
$m^3_{3,3}\|125^*$	NO	YES

Table 6.13: Relative error of test results for cubature rules for $m^3_{3,1}$

Test	d_e	$m^3_{3,1}\|27^*$	$m^3_{3,1}\|26$	$m^3_{3,1}\|23$	$m^3_{3,1}\|22$	$m^3_{3,1}\|21b$	$m^3_{3,1}\|21a$
1	1.08E-01	0.00%	-0.09%	0.00%	0.00%	0.00%	0.00%
2	4.32E-01	0.05%	-0.39%	-0.19%	0.07%	0.23%	0.05%
3	3.00E-05	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
4	1.08E-01	-0.28%	-0.37%	-0.37%	-0.37%	-0.37%	-0.28%
5	4.32E-01	-0.09%	-0.86%	-0.46%	0.05%	0.16%	-0.02%
6	3.00E-05	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
7	1.08E-01	-6.75%	-6.75%	-7.12%	-6.94%	-6.94%	-6.66%
8	4.32E-01	-2.99%	-2.20%	-1.02%	-0.56%	-0.72%	-1.92%
9	3.00E-05	0.00%	0.00%	0.00%	0.00%	-0.23%	0.00%
11	8.73E-02	-1.87%	-1.88%	-11.26%	-1.96%	-1.90%	-1.85%
12	5.02E-01	-2.97%	-0.91%	19.98%	81.19%	61.76%	3.10%
13	5.42E-03	-0.12%	-0.13%	-0.12%	-0.11%	-0.07%	-0.08%
14	1.75E-03	0.49%	0.51%	0.57%	0.63%	0.74%	0.49%
15	5.60E-06	-7.74%	-8.29%	-9.98%	-8.80%	-7.30%	-6.28%
16	1.16E-05	-0.71%	-0.95%	-1.55%	-1.47%	-0.68%	-0.07%
17	1.26E-06	-9.68%	-9.68%	-9.76%	-9.76%	-9.55%	-9.47%
18	4.06E-06	0.11%	0.12%	-0.27%	-0.22%	-0.15%	0.14%
19	7.23E-06	-19.89%	-21.13%	-32.31%	-29.97%	-21.56%	-11.59%
20	1.70E-05	-7.60%	-7.43%	-14.27%	-13.15%	-9.73%	-3.40%
21	2.56E-06	-2.33%	-1.29%	-2.07%	-2.38%	-2.26%	-2.31%
22	1.30E-05	1.52%	1.46%	2.00%	1.85%	1.76%	1.40%
23	3.09E-01	-6.30%	-6.32%	-6.50%	-6.47%	-6.90%	-6.28%
24	1.82E-05	-45.42%	-46.22%	-48.19%	-47.62%	-45.68%	-17.12%
25	9.24E-02	-7.43%	-1.34%	-21.14%	-20.22%	-19.90%	-9.37%

6.5.3 Performance evaluation for hexahedrals

To evaluate the performance of the cubature rules detailed in Section 6.5.2 for hexahedral elements, we have selected a series of standard tests in section 5.4.2. Table 6.13 presents the performance results for rules used with tri-linear geometry interpolation, focusing on those that do not introduce spurious zero-energy modes. The cubature rules $m^3_{3,1}|26$ and $m^3_{3,1}|21a$ show comparable performance to the standard product rule $m^3_{3,1}|27^*$ across most tests, though some variations are observed. Notably, $m^3_{3,1}|21a$ excels in tests 19 and 20, while $m^3_{3,1}|26$ shows superior results in tests 12 and 25. These observations underscore the variable effectiveness of these rules in different testing scenarios.

Table 6.14 presents performance data for cubic-serendipity geometry interpolation, highlighting the new rules $m^3_{3,3}|83$ and $m^3_{3,3}|87$. These rules perform comparably to the traditional product rule $m^3_{3,3}|125^*$ across all tested scenarios. This similarity in performance, achieved with significantly fewer integration points, demonstrates the efficiency and potential utility of the new rules in practical applications.

Table 6.14: Relative error of test results for cubature rules for $m^3_{3,3}$

Test	d_e	$m^3_{3,3}\|125^*$	$m^3_{3,3}\|83$	$m^3_{3,3}\|87$
1	1.08E-01	0.00%	0.00%	0.00%
2	4.32E-01	0.05%	0.05%	0.05%
3	3.00E-05	0.00%	0.00%	0.00%
4	1.08E-01	-0.37%	-0.37%	-0.37%
5	4.32E-01	0.16%	-0.09%	-0.09%
6	3.00E-05	0.00%	0.00%	0.00%
7	1.08E-01	-7.22%	-7.22%	-7.22%
8	4.32E-01	-3.10%	-3.10%	-3.10%
9	3.00E-05	0.00%	0.00%	0.00%
11	8.86E-02	-0.97%	-0.97%	-0.97%
12	5.08E-01	-2.75%	-2.75%	-2.76%
13	5.42E-03	-57.66%	-57.73%	-57.65%
14	1.75E-03	-49.46%	-49.45%	-49.11%
15	5.60E-03	-9.11%	-9.11%	-9.11%
16	1.16E-05	-1.56%	-1.56%	-1.55%
17	1.26E-06	-9.88%	-9.88%	-9.92%
18	4.06E-06	-0.30%	-0.30%	-0.30%
19	7.23E-06	-36.79%	-36.79%	-36.79%
20	1.70E-05	-16.09%	-16.09%	-16.10%
21	2.56E-06	-2.21%	-2.21%	-2.19%
22	1.30E-05	1.96%	1.96%	2.00%
23	3.23E-01	2.20%	2.20%	2.20%
24	4.73E-05	-21.08%	-21.08%	-21.08%
25	1.85E-01	-13.89%	-13.89%	-13.87%

The tables offer a detailed comparison of relative error performance across various cubature rules. The findings reveal that the rules $m^3_{3,3}|26$, $m^3_{3,3}|21a$, $m^3_{3,3}|83$, and $m^3_{3,3}|87$ match or exceed the accuracy of their respective product rules. This consistent performance highlights their viability for practical use in finite element applications.

6.6 Contribution analysis of deformation modes

Despite having no SZEMs, the rule $m^3_{3,1}|21b$ displayed relatively large displacement errors, as noted in the preceding section. This discrepancy is linked to imprecise integration of the stiffness matrix, affecting the accuracy of the eigenvectors and altering expected deformation patterns. Consequently, these integration inaccuracies likely contribute to the increased displacement errors observed with this rule.

In test 12, rule $m^3_{3,1}|21b$ demonstrated a relative error of 61.76%, significantly higher than the errors for other rules in Table 6.13. This discrepancy can be examined through FEM-related eigenproblems, as outlined in (Bathe, 1996, Sec. 10.2). Analyzing the real symmetric global stiffness matrix $\boldsymbol{K}$ using eigendecomposition helps understand the issues causing the observed errors in deformation patterns and displacement calculations.

The global stiffness matrix $\boldsymbol{K}$ can be decomposed as follows:

$$\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^T. \tag{6.23}$$

In this representation, $\boldsymbol{\Phi}$ is the matrix where each column i represents the eigenvector $\boldsymbol{\phi}_i$ of $\boldsymbol{K}$, often referred to as a deformation mode. The matrix $\boldsymbol{\Lambda}$ is diagonal

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & \lambda_n \end{bmatrix}, \tag{6.24}$$

with its diagonal elements $\Lambda_{ii} = \lambda_i$ representing the corresponding eigenvalues.

Eigenvectors are orthogonal, meaning $\boldsymbol{\phi}_i^T \boldsymbol{\phi}_j = 0$ for $i \neq j$. Figure 6.8 illustrates the orthogonality of eigenvectors for rules $m^3_{3,1}|21b$ and $m^3_{3,1}|22$, compared to $m^3_{3,1}|27^*$, focusing on the fifteen smallest eigenvalues of the global stiffness matrix in test 12. When the stiffness matrices derived from two different cubature rule a and b are equivalent or very similar, the matrix plot $\boldsymbol{\Phi}_a^T \boldsymbol{\Phi}_b$ approaches an identity matrix. This analysis shows that the stiffness matrix derived from rule $m^3_{3,1}|22$ more closely aligns with that from rule $m^3_{3,1}|27^*$ than does $m^3_{3,1}|21b$.

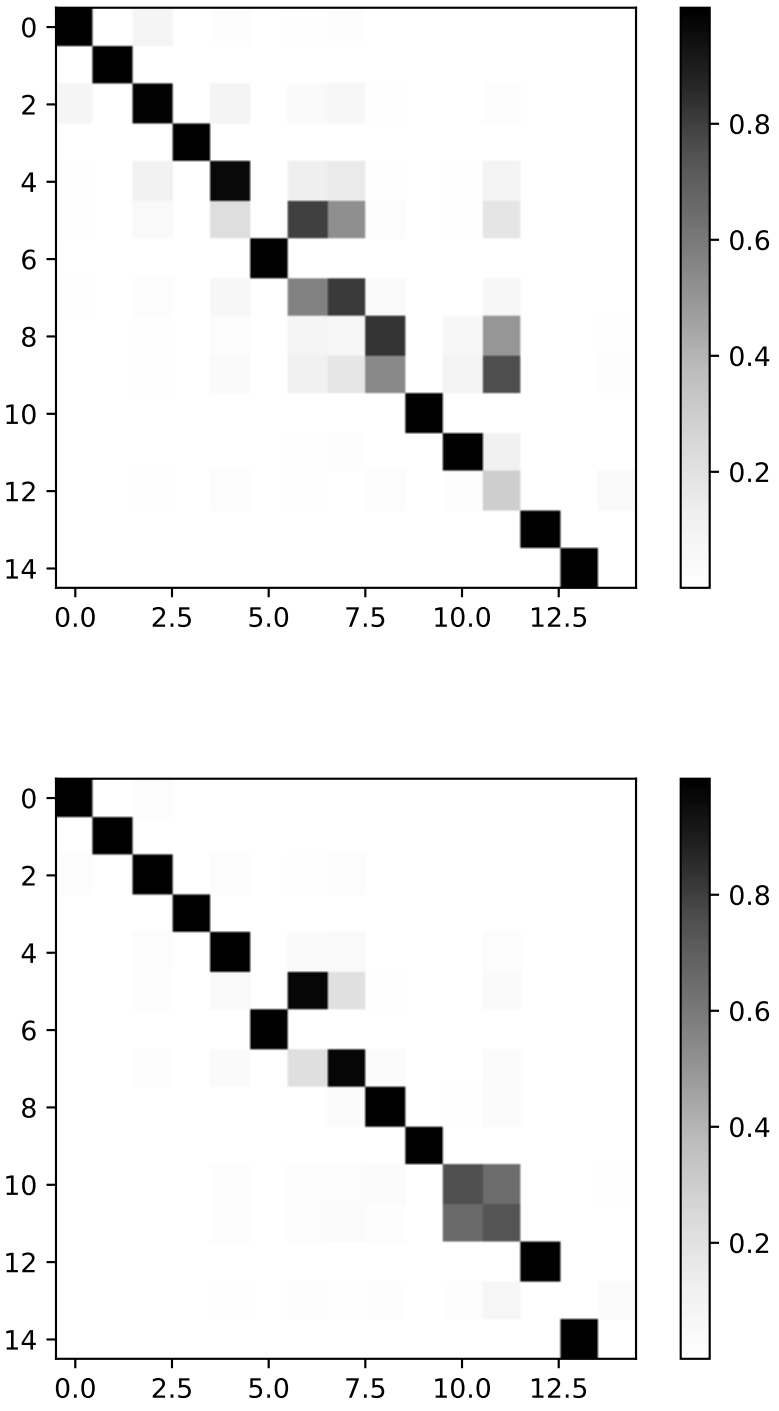


Figure 6.8: The orthogonality of $m^3_{3,1}|21b$ (above), $m^3_{3,1}|22$ (below) w.r.t. $m^3_{3,1}|27^*$ for the eigenvectors correspond to the smallest fifteen eigenvalues of global stiffness matrix of Test 12.

In structural analysis, $\boldsymbol{P}$ denotes the load vector and $\boldsymbol{\Gamma}_i$ represents the contribution factor of the ith eigenvector to the applied loading. The contribution factor $\boldsymbol{\Gamma}_i$ is calculated using the formula:

$$\boldsymbol{\Gamma}_i = \boldsymbol{\phi}_i^T \boldsymbol{\phi}_i \boldsymbol{P}. \tag{6.25}$$

Table 6.15: The contribution factors at the nodes $j = 1,..,4$ for the smallest five eigenvalues for $m^3_{3,1}|21b$ in test 12

	λ_1	λ_2	λ_3	λ_4	λ_5
γ_{i1}	0.0313854444	2.6413433685E-9	0.0220778308	-5.1768138903E-10	0.0138996821
γ_{i2}	0.0313011979	2.0901770466E-11	0.0171853314	5.3207441970E-10	0.0039146494
γ_{i3}	0.0313574672	1.5499251701E-9	0.0204473622	-8.0404556359E-11	0.0105712994
γ_{i4}	0.0313293288	1.1052794173E-9	0.0188163267	9.6221738310E-11	0.0072414211

Table 6.16: The contribution factors at the nodes $j = 1,..,4$ for the smallest five eigenvalues for $m^3_{3,1}|22$ in test 12

	λ_1	λ_2	λ_3	λ_4	λ_5
γ_{i1}	0.0277925559	2.1319633186E-10	0.0221341008	-2.0567802866E-14	0.0141975165
γ_{i2}	0.0284020258	2.1152474008E-10	0.0197350014	6.8161413449E-12	0.0089660648
γ_{i3}	0.0279953737	2.1196662286E-10	0.0213348132	1.1903022679E-12	0.0124485456
γ_{i4}	0.0281984990	2.1030212577E-10	0.0205349936	3.5661152137E-12	0.0107053234

Table 6.17: The contribution factors at the nodes $j = 1,..,4$ for the smallest five eigenvalues for $m^3_{3,1}|27^*$ in test 12

	λ_1	λ_2	λ_3	λ_4	λ_5
γ_{i1}	0.0265527498	1.3002892468E-9	0.0219823044	5.3578558295E-11	0.0141567783
γ_{i2}	0.0273392757	1.6541760715E-9	0.0205459558	-4.4773983391E-11	0.0107119388
γ_{i3}	0.0268149113	1.4171232358E-9	0.0215032697	2.1272414496E-11	0.0130073584
γ_{i4}	0.0270770830	1.5346425531E-9	0.0210244641	-1.2229631388E-11	0.0118589069

Tables 6.15 to 6.17 detail the contribution factors γ_{ij} for the ith eigenvalue λ_i at the Degrees of Freedom (DoFs) $j = 1,..,4$ as illustrated in Fig. 6.9. Here, the loads $\boldsymbol{P}$ are applied using equivalent nodal forces at the edge with four nodes for cubic elements. These contributions are calculated for the smallest five eigenvalues from the rules $m^3_{3,1}|21b$, $m^3_{3,1}|22$, and $m^3_{3,1}|27^*$. The nodal forces, denoted as P_i, are determined by integrating the pressure p over the area A, weighted by the shape functions N_i, as shown in the following equation:

$$P_i = \int_A N_i p dA. \tag{6.26}$$

Table 6.18 lists the inverses of the five smallest eigenvalues for the cubature rules $m^3_{3,1}|21b$, $m^3_{3,1}|22$, and $m^3_{3,1}|27^*$, which are critical for evaluating the structural response under the applied load.

Table 6.18: The inverse of the five smallest eigenvalues of global stiffness matrix in test 12 for $m^3_{3,1}|21b$, $m^3_{3,1}|22$ and $m^3_{3,1}|27^*$.

	$m^3_{3,1}\|21b$	$m^3_{3,1}\|22$	$m^3_{3,1}\|27^*$
$1/\lambda_1$	25.18346483	19.07142081	17.84936528
$1/\lambda_2$	4.570843825	4.564883442	4.566799369
$1/\lambda_3$	1.245101291	0.907700153	0.791815246
$1/\lambda_4$	0.203563427	0.202937535	0.203715568
$1/\lambda_5$	0.105064473	0.085353467	0.079301036

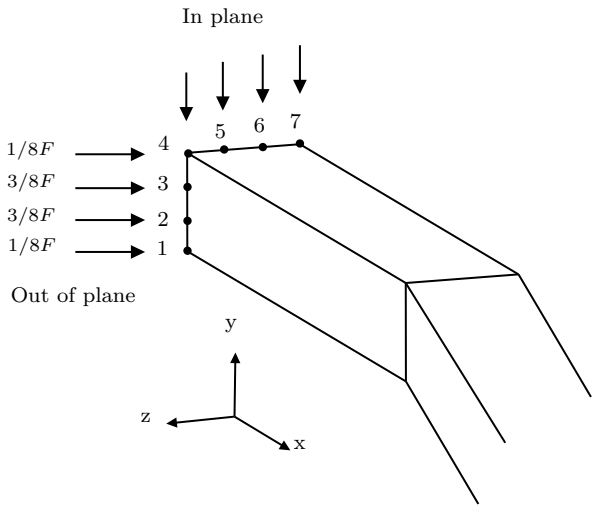


Figure 6.9: In plane and out of plane load for curved beam.

In Tables 6.15 to 6.17, we note that the second and fourth smallest eigenvalues contribute minimally to the response under the specified loading conditions. Therefore, the closeness of these particular eigenvalues across the three integration rules does not significantly impact their performance. However, as shown in Table 6.18, there are notable differences in the first and third eigenvalues between $m^3_{3,1}|21b$, $m^3_{3,1}|22$, and $m^3_{3,1}|27^*$. These eigenvalues are more critical under the given loading scenario.

As mentioned in Section A.1, the general algebraic form of the problem to be solved is

$$\boldsymbol{K}\tilde{\boldsymbol{u}} = \boldsymbol{P}. \tag{6.27}$$

Substitute Equation (6.23) into Equation (6.27) returns

$$\boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^T\tilde{\boldsymbol{u}} = \boldsymbol{P}. \tag{6.28}$$

The deformation modes $\tilde{\boldsymbol{u}}_i$ are computed using Equation (6.28), yielding:

$$\tilde{\boldsymbol{u}}_i = \frac{1}{\lambda_i}\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i\boldsymbol{P} = \frac{1}{\lambda_i}\boldsymbol{\Gamma}_i \tag{6.29}$$

which satisfy

$$\sum_i \tilde{\boldsymbol{u}}_i = \tilde{\boldsymbol{u}}. \tag{6.30}$$

This equation strategically adjusts each mode's contribution based on its eigenvalue. The calculated displacements ($\tilde{\boldsymbol{u}}$) at DoFs $j = 1,..,4$ are detailed in Table 6.19.

Table 6.19: The final response at DoFs j=1,..,4 for $m^3_{3,1}|21b$, $m^3_{3,1}|22$ and $m^3_{3,1}|27^*$ in test 12.

	$m^3_{3,1}\|21b$	$m^3_{3,1}\|22$	$m^3_{3,1}\|27^*$
u_1	0.8193624707	0.5513728033	0.4926617678
u_2	0.8103823282	0.5603891473	0.5052557046
u_3	0.8163724063	0.5543726104	0.4968597330
u_4	0.8133776758	0.5573774003	0.5010575958

Table 6.20 shows the relative error in the contribution of the first deformation mode compared to the final responses at DoFs $j = 1,..,4$. Furthermore, Table 6.21 displays the relative errors for the combined contributions of the first and third deformation modes.

Table 6.20: The relative error of first deformation mode i=1 w.r.t. final response at DoFs j=1,..,4 for $m^3_{3,1}|21b$, $m^3_{3,1}|22$ and $m^3_{3,1}|27^*$ in test 12.

	$m^3_{3,1}\|21b$	$m^3_{3,1}\|22$	$m^3_{3,1}\|27^*$
u_{11}	-3.53%	-3.86%	-3.79%
u_{12}	-2.72%	-3.34%	-3.41%
u_{13}	-3.26%	-3.69%	-3.66%
u_{14}	-2.99%	-3.51%	-3.54%

Table 6.21: The relative error of the summation of first and third deformation modes i=1,3 w.r.t. final response at DoFs j=1,..,4 for $m^3_{3,1}|21b$, $m^3_{3,1}|22$ and $m^3_{3,1}|27^*$ in test 12.

	$m^3_{3,1}\|21b$	$m^3_{3,1}\|22$	$m^3_{3,1}\|27^*$
$u_{11}+u_{31}$	-0.18%	-0.22%	-0.26%
$u_{12}+u_{32}$	-0.08%	-0.14%	-0.19%
$u_{13}+u_{33}$	-0.15%	-0.19%	-0.24%
$u_{14}+u_{34}$	-0.12%	-0.17%	-0.22%

The analysis reveals that accounting solely for the smallest eigenvalue of $\boldsymbol{K}$ results in a relative error close to 4% in each case when compared to the numerical solution. However, incorporating both the smallest and the third smallest eigenvalues of $\boldsymbol{K}$ substantially lowers the relative error to about 0.3% in each instance. This suggests that the deformation modes corresponding to these specific eigenvalues are critical for precisely predicting the final responses.

On the other hand, the performance of $m^3_{3,1}|21b$ and $m^3_{3,1}|22$ in Test 11 is much better than in Test 12. If we look at the contribution factors γ_{ij} at the DoFs $j = 4,..,7$ as shown in the Tables 6.22 to 6.24, it is interesting to observe that for the given loading, the contribution of the first, third and fifth (smallest) eigenvalues is very small compare to the second and fourth which means the proximity of the corresponding eigenvalues among the three cases is not important. In Table 6.18, three integration schemes have similar value for the second and fourth eigenvalues which are the most relevant for this loading and Table 6.25 and 6.26 shows that the deformation modes correspond to the second and fourth smallest eigenvalues have significant influence on the final responses.

Table 6.22: The contribution factors at the nodes $j = 4,..,7$ for the smallest five eigenvalues for $m^3_{3,1}|21b$ in test 11

	λ_1	λ_2	λ_3	λ_4	λ_5
γ_{i4}	2.6093050331E-9	-0.0180379649	6.0817642631E-8	-0.0249883158	2.1390289690E-8
γ_{i5}	-3.7769473042E-9	-0.0180378863	-6.0915651021E-8	-0.0249882078	-2.1391594731E-8
γ_{i6}	4.3484867280E-10	-0.0180378683	2.0283406787E-8	-0.0249885982	7.1216148473E-9
γ_{i7}	-1.6024854930E-9	-0.0180378421	-2.0381415394E-8	-0.0249885622	-7.1229199150E-9

Table 6.23: The contribution factors at the nodes $j = 4,..,7$ for the smallest five eigenvalues for $m^3_{3,1}|22$ in test 11

	λ_1	λ_2	λ_3	λ_4	λ_5
γ_{i4}	-5.0810842732E-9	-0.0180570701	3.5024211607E-8	-0.0249951189	4.1651733651E-9
γ_{i5}	5.0193836087E-9	-0.0180570378	-3.5167333660E-8	-0.0249950812	-4.1654077149E-9
γ_{i6}	-1.7118160177E-9	-0.0180570439	1.1629903496E-8	-0.0249949800	1.3920215794E-9
γ_{i7}	1.6501153000E-9	-0.0180570331	-1.1773025170E-8	-0.0249949674	-1.3922559259E-9

Table 6.24: The contribution factors at the nodes $j = 4,..,7$ for the smallest five eigenvalues for $m^3_{3,1}|27^*$ in test 11

	λ_1	λ_2	λ_3	λ_4	λ_5
γ_{i4}	-2.8792896300E-8	-0.0180362685	2.7740594588E-8	-0.0249836482	8.4111343957E-9
γ_{i5}	2.7683459209E-8	-0.0180362956	-2.7997131757E-8	-0.0249836056	-8.4135267366E-9
γ_{i6}	-9.9673720669E-9	-0.0180362720	9.1620321818E-9	-0.0249835445	2.8035816427E-9
γ_{i7}	8.8579351902E-9	-0.0180362810	-9.4185691315E-9	-0.0249835303	-2.8059739660E-9

In Test 11, the performance of the integration rules $m^3_{3,1}|21b$ and $m^3_{3,1}|22$ is markedly better than in Test 12. Analyzing the contribution factors γ_{ij} for degrees of freedom (DoFs) $j = 4..7$, as presented in Tables 6.22 to 6.24, it is notable that the contributions of the first, third, and fifth smallest eigenvalues are minimal compared to those of the second and fourth. This indicates that the closeness of the corresponding eigenvalues among the three integration schemes is less crucial for this specific loading. Tables 6.18 confirm that the three schemes yield similar values for the second and fourth eigenvalues, which are more significant for this loading scenario. Furthermore, Tables 6.25 and 6.26 illustrate that the deformation modes associated with the second and fourth smallest eigenvalues significantly impact the final responses, underlining their importance in this context.

Table 6.25: The relative error of second deformation mode i=2 w.r.t. final response at DoFs j=4,..,7 for $m^3_{3,1}|21b$, $m^3_{3,1}|22$ and $m^3_{3,1}|27^*$ in test 11.

	$m^3_{3,1}\|21b$	$m^3_{3,1}\|22$	$m^3_{3,1}\|27^*$
u_{24}	-6.18%	-6.14%	-6.19%
u_{25}	-6.18%	-6.07%	-6.19%
u_{26}	-6.18%	-6.12%	-6.19%
u_{27}	-6.18%	-6.09%	-6.19%

Table 6.26: The relative error of the summation of second and fourth deformation modes i=2,4 w.r.t. final response at DoFs j=4,..,7 for $m^3_{3,1}|21b$, $m^3_{3,1}|22$ and $m^3_{3,1}|27^*$ in test 11.

	$m^3_{3,1}\|21b$	$m^3_{3,1}\|22$	$m^3_{3,1}\|27^*$
$u_{24}+u_{44}$	-0.39%	-0.36%	-0.39%
$u_{25}+u_{45}$	-0.39%	-0.29%	-0.39%
$u_{26}+u_{46}$	-0.39%	-0.34%	-0.39%
$u_{27}+u_{47}$	-0.39%	-0.31%	-0.39%

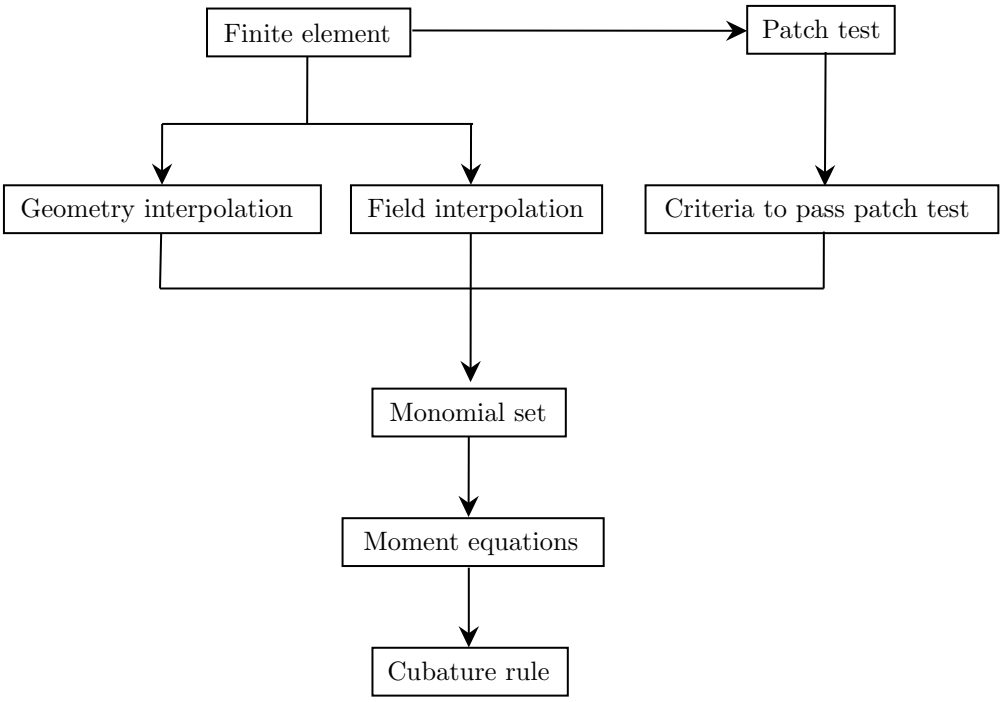


Figure 6.10: Framework to find minimum cubature rules passing the patch test

These findings explain the larger deformation errors observed in Test 12 for $m^3_{3,1}|21b$ and $m^3_{3,1}|22$ when compared to $m^3_{3,1}|27^*$. The difference between the outcomes of Tests 11 and 12 suggests that specific loading conditions can activate inaccurate deformation modes, which significantly impact the final response, primarily due to inadequate integration in the stiffness matrix.

6.7 Summary

This chapter introduces a new methodology, depicted in Figure 6.10, for generating cubature rules that ensure finite elements pass the patch test. This process efficiently identifies the exact monomial sets needed for integration, forming the basis for deriving the moment equations for specific cubature rules. Our versatile framework accommodates different element shapes and various interpolations of displacement and geometry, extending to patch tests with non-constant stress interpolation. We focus on quadrilaterals and hexahedra with serendipity interpolation. Initially, we check the cubature rules for the absence of spurious zero-energy modes, then evaluate their performance through classic finite element tests.

Many new rules are successfully developed, especially for cubic serendipity interpolation of displacements in hexahedral elements. These rules use up to 34% fewer integration points than standard product rules when geometry is also interpolated with cubic serendipity shape functions. This reduction demonstrates the efficiency of our approach in reducing the computational demand of finite element analyses.

For interpolation orders greater than three, the proposed method produces rules that outperform product rules in point count while maintaining similar accuracy. However, benchmark tests are essential to evaluate each proposed rule, as performance can vary even among rules that pass the patch test. The suboptimal performance of some rules is explained through deformation mode contribution analyses, relying on eigendecomposition of the global stiffness matrix.

In conclusion, our method offers clear advantages for serendipity interpolation by utilizing fewer basis functions than Lagrange interpolation. However, its current application is limited to linear problems, and further investigation is needed to assess its effectiveness in higher-order and nonlinear contexts.

Chapter 7

Conclusions

Numerical integration approximates integrals of functions that are difficult to integrate analytically, especially over intricate or irregular domains where exact solutions are unattainable. It is essential for precise calculations of areas, volumes, and other integral properties, supporting accurate and reliable results in various scientific and engineering applications.

In FEM, numerical integration is crucial for computing each element's contribution to the overall system equations, affecting the model's behavior. Proper integration ensures the stability and accuracy of FEM solutions by accurately calculating element properties like the stiffness matrix. Therefore, studying numerical integration and its application in FE simulations is essential.

In this thesis, we present significant contributions to Gaussian-type cubature rules and their application to finite element simulation. First, we establish an explicit consistency condition for fully symmetric cubature rules in tetrahedra using a novel basis. Second, we identify new rotationally symmetric PI rules for tetrahedra and investigate multisymmetric polynomials in cubature rule moment equations. Third, we develop a framework to identify cubature rules with the minimum integration points needed to pass the patch test by exactly integrating only the necessary monomial sets.

In Chapter 3, we developed a new non-monomial fully symmetric polynomial basis for the tetrahedron. This basis maximizes the number of zero elements across various orbit types, allowing for explicit formulas and identification of optimal and quasi-optimal consistent rule structures for the first time. This approach simplifies moment equations into smaller, independent subsystems, improving solution efficiency. Using this method, we derived a new NI rule of degree 9 with 55 points, reducing the number of points compared to existing PI/NI rules. However, optimal rule structures for degrees 7 and 9 were impractical due to complex point coordinates, and degrees 10 and 11 resulted in inconsistent moment equations. Quasi-optimal rule structures serve as a foundation for computing additional cubature rules, with higher degrees requiring advanced solving techniques due to equation complexity. This approach has potential applications in deriving consistency conditions across various domains and symmetries.

The primary advantage of the method employed in this section lies in its ability to explore the root distribution properties of the quartic function governing the barycentric coordinates, through the analysis of its discriminant. By examining the relationship between the roots and the distribution of points, we derive a novel basis that eliminates redundant orbits in the moment equations. This, in turn, facilitates the formulation of accurate consistency conditions and optimal rule structures. However, despite the benefits of this new basis, it introduces challenges in solving the moment equations due to the interdependence between the basis variables p, q and r, which creates redundancy. To render the moment equations solvable, further simplifications are necessary. Moreover, as discussed in the concluding section of this chapter, this redundancy can occasionally prevent the existence of a solution altogether.

In Chapter 4, we explore rotational symmetry in cubature formulation and the role of multisymmetric polynomials in moment equations. By developing a new rotationally symmetric monomial basis that incorporates the Vandermonde polynomial into the fully symmetric polynomial basis from Chapter 3, we created a comprehensive system of moment equations. Augmented by equations from the relationship between Vandermonde polynomials and discriminants, this system becomes zero-dimensional. Using Gröbner bases, we found a novel degree 7 PI rule requiring three fewer points than the optimal fully symmetric PI rule. For higher degrees, numerical solutions computed with the PHG code using the Levenberg-Marquardt algorithm yielded three PI rules for degrees 9 to 11, each with fewer or equal points compared to their fully symmetric counterparts. Degrees beyond 11 face convergence challenges due to the complex system of moment equations and numerous variables, indicating a need for enhanced computational capabilities. This methodology can extend to derive rotationally symmetric cubature rules for other symmetric domains.

The optimal rotationally symmetric cubature rules for a given degree may require fewer integration points than fully symmetric cubature rules, as demonstrated in our work. This is due to the reduced constraints on the distribution of integration points in rotationally symmetric rules. We achieve rotational symmetry by employing the Vandermonde polynomial, leveraging its anti-symmetry property. When combined with a fully symmetric basis, this results in a rotationally symmetric basis. However, this derived basis is not optimal for establishing accurate consistency conditions or constructing optimal rule structures. Furthermore, the moment equations generated by the new rotationally symmetric basis form a positive-dimensional system. The interdependence of the Vandermonde polynomial and other basis variables, governed by the discriminant of a quartic function, complicates the solution process when using symbolic methods such as Gröbner bases. The Levenberg-Marquardt algorithm, by contrast, is a well-established numerical method capable of finding local optima. However, it identifies only a single solution at a time, and there is no guarantee that the solution is globally optimal.

Rotationally symmetric cubature rules could be particularly beneficial in FEM when the numbering of element vertices aligns with the rotational symmetry of the cubature rule. Moreover, in certain finite element simulations, variations in vertex numbering require differing rule performance, highlighting the value of rotationally symmetric cubature rules. However, in general applications, fully symmetric cubature rules are often preferable due to their invariance to vertex numbering.

Additionally, we examined multisymmetric polynomials within cubature rule moment equations, expressing them in terms of elementary multisymmetric polynomials and revealing dimensional redundancies. Developing effective algorithms to manage and simplify these expressions and explain their canonical forms remains an active research area, enhancing our understanding of their algebraic properties and their potential in cubature formulations.

In Chapter 6, we developed a framework to derive cubature rules that ensure finite elements pass the patch test. This framework identifies and integrates necessary monomial sets, simplifying the generation of moment equations. It accommodates various element shapes and interpolations of displacement and geometry. Focusing on quadrilaterals and hexahedrals with serendipity interpolation, the derived rules are validated for the absence of spurious zero-energy modes and evaluated through finite-element test simulations. The research introduces many new rules, notably for cubic serendipity interpolation in hexahedral elements, achieving up to a 34% reduction in integration points compared to traditional product rules. For interpolation orders above three, the method consistently reduces integration points while maintaining accuracy. However, benchmark tests are necessary to assess each new rule, as performance can vary even among those passing the patch test. This highlights the importance of thorough testing in developing and applying new cubature rules in finite element analysis.

Our method demonstrates advantages specifically in the context of serendipity interpolation due to its use of fewer basis functions compared to Lagrange interpolation for Lagrange elements. However, this study is limited to linear problems. To fully explore the potential of our framework, it is necessary to extend the analysis to higher-order cases and nonlinear problems. For example, our method can be extended to consider higher order patch tests with non-constant interpolation of the stress.

Verification and validation are critical processes in ensuring the accuracy and reliability of FE codes, as they address different aspects of the computational model's fidelity. Validation assesses whether the mathematical model implemented in the FE code accurately represents the physical phenomena of the real-world problem. It focuses on ensuring that the simulation produces reliable predictions when compared to experimental or empirical data. In our work, we assumed that the mathematical model has been validated.

Verification, in contrast, ensures that the numerical implementation of the FE method is correctly executed. It verifies that the mathematical equations, derived from theoretical principles, are accurately represented and solved by the software. This process involves checking that the FE code properly assembles stiffness matrices, applies boundary conditions, and solves the governing equations (such as equilibrium equations in structural analysis). Verification is typically performed by comparing the FE results with exact analytical solutions for simple problems (e.g., beam bending or plane stress) or through methods such as the patch test and unit testing. While unit tests are now widely preferred for verifying individual components of FE code, the convergence properties ensured by the patch test remain an essential aspect of comprehensive verification.

In conclusion, this thesis investigates the invariance of Gaussian-type cubature rules, leading to several innovative results. Additionally, it establishes a comprehensive framework for identifying cubature rules tailored for finite element analysis. The outcomes demonstrate significant advancements in numerical integration and its application to FE simulation.

This thesis makes significant contributions to the field and also identifies several opportunities for further research, among which four directions are particularly compelling. The first opportunity highlighted is the construction and simplification of moment equations. We derived fully symmetric and rotationally symmetric bases for tetrahedra. The fully symmetric basis simplifies the moment equations, making the solutions easier to express, and enables explicit formulas for calculating consistency conditions necessary for estimating optimal cubature rules. This suggests that exploring the invariances inherent in the integration domain and cubature formulation can significantly aid in finding superior cubature rules. Consequently, we generalized this idea to include multisymmetric polynomials in the moment equations, expressed using elementary multisymmetric polynomials, similar to symmetric polynomials. However, redundancies emerged in the newly expressed moment equations. Although explicit formulas involving elementary multisymmetric polynomials can be obtained using the Gröbner basis method, the computational load increases with the problem's complexity. Thus, mathematically deriving closed-form relationships is an essential next step for advancing this research area.

The rotationally symmetric basis developed in Chapter 4 has led to the discovery of novel cubature rules that require fewer integration points. However, the consistency conditions for the optimal rule structure have not been identified because those derived from the rotationally symmetric basis result in more points than the optimal rules calculated using the fully symmetric basis in Section 3. This discrepancy is irrational, as fully symmetric polynomials are inherently rotationally symmetric. Therefore, the optimal rule structure for a fully symmetric cubature rule must also be a rotationally symmetric cubature rule. The fact that the optimal

rule structure calculated using the rotationally symmetric basis requires more integration points than the optimal fully symmetric cubature rules indicates a need for a better method to calculate the consistency conditions for optimal rotationally symmetric cubature rules. This presents the second opportunity for further research.

The third opportunity lies in the analysis of the integrand. We have successfully identified the sets of monomials that must be integrated exactly to pass the patch test for FE simulation. This approach can be extended to various elements and interpolations for geometry and assumed fields used in FEM. Therefore, it would be beneficial to establish a library to compare the performance of different cubature rules within our framework under various conditions in FE simulations. This library would particularly assess their ability to integrate the stiffness matrix properly and ensure they facilitate passing the patch test for convergence.

The fourth research opportunity arises from the methodology used to solve moment equations. We have discussed that polynomial systems can be solved either algebraically or numerically. For the algebraic approach, we emphasized the benefits of exploiting invariance in the integration domain and cubature formulation, which simplifies the algebraic solutions and facilitates the resolution of moment equations. For the numerical approach, particularly iterative methods, there is significant potential for addressing higher-dimensional problems. It is valuable to explore strategies that enhance the convergence of these algorithms, either by improving initial guess methods to ensure they reach the optimal solution or by accelerating their convergence rate with high-performance computing (HPC) algorithms.

The progression of these areas relies heavily on advances in mathematical research and computational capabilities, which are expected to evolve rapidly to meet future development needs.

Appendix A

FE formulation for linear elasticity

A.1 Direct formulation to linear elasticity

The finite element method (FEM) simplifies continuum problems by dividing them into discrete elements, each defined by specific parameters. This approach allows solving complex continuum problems using discrete structural engineering methods. While FEM can be extended to nonlinear problems, our research focuses on linear elasticity. Key definitions are based on (Bathe, 1996; MacNeal, 1994; O. C. Zienkiewicz et al., 2013).

Consider a typical triangular finite element, labeled as e in Figure A.1, with nodes at 1, 2, and 3 connected by straight edges. A crucial aspect of FEM is the choice of displacement functions to approximate displacements within an element. Let the displacements $\boldsymbol{u}^e$ at any point within the element be approximated by a column vector $\hat{\boldsymbol{u}}^e$:

$$\boldsymbol{u}^e \approx \hat{\boldsymbol{u}}^e = \sum_i \boldsymbol{N}_i \tilde{\boldsymbol{u}}_i^e = \boldsymbol{N}\tilde{\boldsymbol{u}}^e. \tag{A.1}$$

The functions $\boldsymbol{N}_i = [N_1, N_2, \dots]$ are shape functions (or interpolation functions) for displacements. They must be chosen to yield appropriate nodal displacements when the coordinates of the corresponding nodes are inserted into Equation (A.1). For example, a linear function for a triangular element will yield the shape of N_3 as shown in Figure A.2.

In finite element analysis, once displacements are known across the element, the strains at any point can be calculated using a matrix relationship expressed as:

$$\varepsilon = \boldsymbol{S}\boldsymbol{u}^e \tag{A.2}$$

where ε represents the strain tensor at a point, and $\boldsymbol{S}$ is a linear differential operator. Using the displacement approximation from Equation (A.1), the strain equation becomes:

$$\varepsilon \approx \hat{\varepsilon} = \boldsymbol{B}\tilde{\boldsymbol{u}}^e. \tag{A.3}$$

Here, $\boldsymbol{B}$, known as the strain-displacement matrix, is defined as:

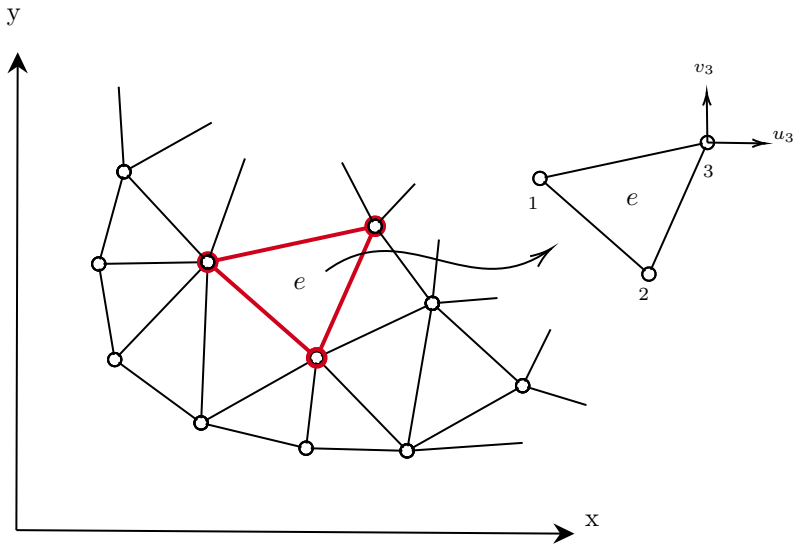


Figure A.1: Region divided into finite elements

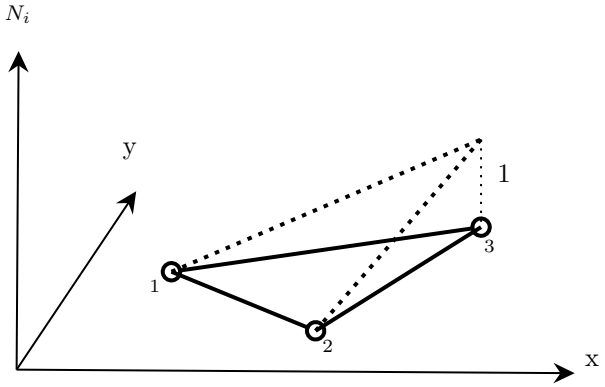


Figure A.2: Shape function N_3 for triangle

$$\boldsymbol{B} = \boldsymbol{SN}. \tag{A.4}$$

The strain-displacement matrix $\boldsymbol{B}$ can be derived from assumed strain distributions, stress distributions, or a combination of both. The applications and implications of these derivations will be discussed in subsequent sections.

In linear elasticity, the relationship between stress and strain is expressed linearly as follows:

$$\sigma = \boldsymbol{D}(\varepsilon - \varepsilon_0) + \sigma_0. \tag{A.5}$$

Here, $\boldsymbol{D}$ represents the elasticity matrix, which contains material properties relevant to the stress-strain response. ε_0 accounts for initial strains, which may arise from factors like temperature changes or structural changes at the microscopic level. σ_0 denotes any initial residual stress present in the material.

Correspondingly, let the vector $\boldsymbol{F}^e$ denote the nodal forces on an element, which are equivalent to the effects of boundary stresses and distributed body forces:

$$\boldsymbol{F}^e = \begin{bmatrix} \boldsymbol{F}_1^e \\ \boldsymbol{F}_2^e \\ \vdots \end{bmatrix} \tag{A.6}$$

Each vector $\boldsymbol{F}_i^e$ in this matrix corresponds to the forces at node i and matches the number of displacement components $\tilde{\boldsymbol{u}}_i^e$ for that node, ensuring correct force application according to the element's orientation and properties.

This thesis considers distributed body forces $\boldsymbol{p}$ acting per unit volume and aligned with displacement directions $\boldsymbol{u}$. However, their impact is not explicitly included in the analysis. Without body forces, equivalent nodal forces are determined solely based on equilibrium, simplifying the computational model and focusing on other structural aspects.

The matrix equation

$$\boldsymbol{K}\tilde{\boldsymbol{u}} = \boldsymbol{P} \tag{A.7}$$

defines the general relationship in finite element analysis, where $\boldsymbol{K}$ is the global stiffness matrix and $\boldsymbol{P}$ is the load vector, including all forces except those from linear elasticity. For linear static analysis, $\boldsymbol{P}$ simplifies to a constant vector.

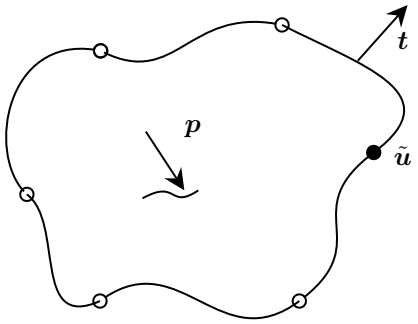


Figure A.3: A representative Finite Element.

The overall system equation for finite element analysis is derived by aggregating the contributions from each element:

$$\boldsymbol{K} = \sum_{e} \boldsymbol{K}^{e} \tag{A.8}$$

In parallel, the total load vector is formulated as:

$$\boldsymbol{P} = \sum_{e} \boldsymbol{P}^{e} + \boldsymbol{P}^{d} \tag{A.9}$$

Here, $\sum_e \boldsymbol{P}^e$ represents the load vector contributions from the individual elements, while $\boldsymbol{P}^d$ pertains to the contributions from displacement-related variables, with the summation extending across all elements.

The element in Figure A.3 models both 2D and 3D finite elements. It includes body force density $\boldsymbol{p}$ and surface or edge traction $\boldsymbol{t}$. The vector $\tilde{\boldsymbol{u}}$ represents boundary variables, mainly displacement-like quantities at discrete nodal points.

When adjacent elements interact solely through shared boundary displacements, it is advantageous to substitute the surface tractions $\boldsymbol{t}$ with generalized forces $\boldsymbol{F^e}$. These forces are exerted on the element due to the boundary displacements $\tilde{\boldsymbol{u}}$.

To determine the element stiffness matrix $\boldsymbol{K}^e$ and the corresponding load vector $\boldsymbol{P}^e$, principles of energy conservation are applied. The principle of virtual work uses virtual displacements $\delta\hat{\boldsymbol{u}}^e$ and virtual strain $\delta\hat{\varepsilon}$. This principle equates the virtual work done by applied forces δW_a with the change in stored potential energy δW_s, expressed as:

$$\delta W_a = \delta W_s. \tag{A.10}$$

The virtual work done by the applied forces, which include the generalized forces from the boundary displacements, is expressed as:

$$\delta W_a = \int_{V_e} \boldsymbol{p}^T \delta \hat{\boldsymbol{u}}^e dV + (\boldsymbol{F^e})^T \delta \tilde{\boldsymbol{u}}^e. \tag{A.11}$$

Here, V_e denotes the volume of the element. The increase in stored energy due to these forces is calculated as:

$$\delta W_s = \int_{V_e} \sigma^T \delta \hat{\varepsilon} dV. \tag{A.12}$$

Consequently, the principle of virtual work (Equation A.10) can be rewritten to establish equilibrium between the internal and external work:

$$\int_{V_e} \sigma^T \delta \hat{\varepsilon} dV = \int_{V_e} \boldsymbol{p}^T \delta \hat{\boldsymbol{u}}^e dV + (\boldsymbol{F^e})^T \delta \tilde{\boldsymbol{u}}^e. \tag{A.13}$$

Using Equations (A.1) and (A.3), substitutions are made as follows:

$$\delta \hat{\varepsilon} = \boldsymbol{B} \delta \tilde{\boldsymbol{u}}^e \tag{A.14a}$$

$$\delta \hat{\boldsymbol{u}}^e = \boldsymbol{N} \delta \tilde{\boldsymbol{u}}^e \tag{A.14b}$$

This leads to the reformulation of the virtual work equation:

$$(\int_{V_e} \sigma^T \boldsymbol{B} dV) \delta \tilde{\boldsymbol{u}}^e = (\int_{V_e} \boldsymbol{p}^T \boldsymbol{N} dV + (\boldsymbol{F^e})^T) \delta \tilde{\boldsymbol{u}}^e. \tag{A.15}$$

Since the elements of $\delta \tilde{\boldsymbol{u}}^e$ are arbitrary and independent, $\delta \tilde{\boldsymbol{u}}^e$ can be cancelled from both sides of Equation (A.15). By transposing both sides, we obtain:

$$\int_{V_e} \boldsymbol{B}^T \sigma dV = \int_{V_e} \boldsymbol{N}^T \boldsymbol{p} dV + \boldsymbol{F}^e. \tag{A.16}$$

We consider only the homogeneous part of the stress-strain relationship from Equation (A.5):

$$\sigma = \boldsymbol{D} \varepsilon. \tag{A.17}$$

Substituting this into the left-hand side of Equation (A.16), we derive:

$$\int_{V_e} \boldsymbol{B}^T \sigma dV = \int_{V_e} \boldsymbol{B}^T \boldsymbol{D} \varepsilon dV = \boldsymbol{K}^e \tilde{\boldsymbol{u}}^e \tag{A.18}$$

where

$$\boldsymbol{K}^e = \int_{V_e} \boldsymbol{B}^T \boldsymbol{D}\boldsymbol{B} dV \tag{A.19}$$

represents the element stiffness matrix. Additionally, the element-associated vector of nodal loads is defined as:

$$\boldsymbol{P}^e = \int_{V_e} \boldsymbol{N}^T \boldsymbol{p} dV \tag{A.20}$$

Consequently, Equation (A.16) simplifies to:

$$\boldsymbol{K}^e \tilde{\boldsymbol{u}}^e = \boldsymbol{P}^e + \boldsymbol{F}^e \tag{A.21}$$

Considering the equilibrium requirement:

$$\sum_e \boldsymbol{F}^e = \boldsymbol{P}^d, \tag{A.22}$$

Equation (A.7) represents the aggregation of Equation (A.21) across all elements linked to $\tilde{\boldsymbol{u}}$. Here, $\boldsymbol{K}$ and $\boldsymbol{P}$ are defined in Equations (A.8) and (A.9). When considering $\boldsymbol{F}^e$ and $\boldsymbol{P}^e$ as directly applied to $\tilde{\boldsymbol{u}}^e$, Equation (A.16) simplifies to:

$$\boldsymbol{f} = \int_{V_e} \boldsymbol{B}^T \sigma dV \tag{A.23}$$

This results in a homogeneous equation that succinctly captures the static equilibrium of forces in the finite element model, aligning with fundamental principles of structural mechanics. This equation is central to our research as it delineates how the generalized nodal forces $\boldsymbol{f}$ can be computed from the stress fields and strain-displacement matrix, bridging theoretical mechanics with practical computational applications.

A.2 Parametric element formulation

When working with parametric elements, performing integration in parametric space is beneficial due to the simpler shapes, such as triangles and cubes. This approach requires establishing a one-to-one mapping between Cartesian and natural coordinates. In three-dimensional problems, this mapping converts natural coordinates (ξ, η, ζ) into Cartesian coordinates (x, y, z):

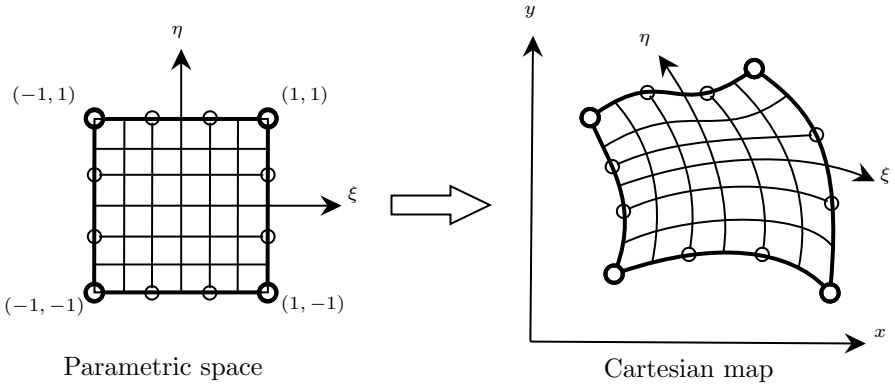


Figure A.4: Two dimensional mapping of a quadrilateral element.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} f_x(\xi,\eta,\zeta) \\ f_y(\xi,\eta,\zeta) \\ f_z(\xi,\eta,\zeta) \end{bmatrix} \tag{A.24}$$

Figure A.4 illustrates a two-dimensional example of mapping, showing the transformation of a quadrilateral element from natural to Cartesian coordinates.

Using this mapping, shape functions are defined in natural coordinates, allowing the transformation of element properties to the Cartesian system. This method optimizes numerical integration by exploiting the geometric simplicity of elements in parametric space. The coordinate transformations (A.24) are specified as:

$$x = N_1' x_1 + N_2' x_2 + \cdots + N_n' x_n = \boldsymbol{N'x} \tag{A.25a}$$

$$y = N_1' y_1 + N_2' y_2 + \cdots + N_n' y_n = \boldsymbol{N'y} \tag{A.25b}$$

$$z = N_1' z_1 + N_2' z_2 + \cdots + N_n' z_n = \boldsymbol{N'z} \tag{A.25c}$$

Here, $\boldsymbol{N'}$ denotes the vector of shape functions mapping natural coordinates to Cartesian coordinates. This ensures each Cartesian coordinate (x, y, z) is a linear combination of the nodal coordinates (x_i, y_i, z_i) weighted by the shape functions. This method seamlessly links the element's geometry with its nodal definitions, simplifying its application in computational models.

To define an element's properties, it is crucial to describe the variation of the unknown, like displacement $\boldsymbol{u}$, within the element. This is done in (A.1) using the expression:

$$\hat{\boldsymbol{u}}^e = \boldsymbol{N}\tilde{\boldsymbol{u}}^e. \tag{A.26}$$

This formulation approximates the displacement field within the element using nodal displacements $\tilde{\boldsymbol{u}}^e$, significantly simplifying the computation of element properties.

In finite element analysis, nodal values may differ from those defining the geometry. When shape functions for geometry and physical quantities, like displacements, are identical, the elements are called isoparametric, represented as:

$$\boldsymbol{N} = \boldsymbol{N}' \tag{A.27}$$

This means the same points define both the geometry and analysis properties of the element. Conversely, elements are subparametric if they use more nodes for field variables than for geometry, and superparametric if they use fewer nodes for field variables than for geometry.

Defining matrices that characterize element properties is critical for finite element analysis. The element stiffness matrix (A.19) depends on the derivatives of shape functions $\boldsymbol{N}$ with respect to Cartesian coordinates. Similarly, the body force vector integral (A.20) crucial for the analysis, also depends on $\boldsymbol{N}$.

To correctly evaluate these matrices, two key transformations are needed. First, express the derivatives with respect to Cartesian coordinates in terms of natural coordinates. Second, convert the element's volume (or area) for integration into natural coordinates. These adjustments ensure accurate limits for numerical integration.

Considering natural coordinates ξ, η, and ζ and their corresponding Cartesian coordinates x, y, and z, the derivative of a shape function N_i with respect to a natural coordinate, such as ξ, can be described using the chain rule for partial differentiation. This derivative is represented by:

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \xi} + \frac{\partial N_i}{\partial z}\frac{\partial z}{\partial \xi} \tag{A.28}$$

Performing similar differentiation with respect to the other natural coordinates, η and ζ, and expressing it in matrix form results in:

$$\begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = \boldsymbol{J} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix}. \tag{A.29}$$

Here, $\boldsymbol{J}$ denotes the jacobian matrix of the transformation. To compute the derivatives of N_i with respect to Cartesian coordinates, $\boldsymbol{J}$ is inverted:

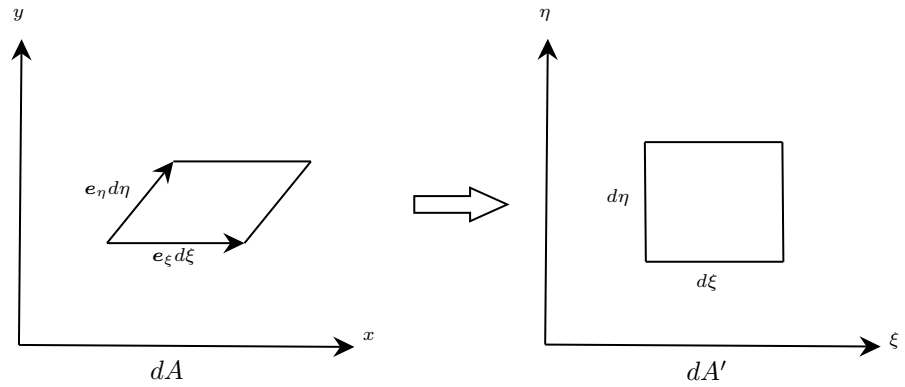


Figure A.5: Parametric mapping of an infinitesimal area.

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = \boldsymbol{J}^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{bmatrix}. \tag{A.30}$$

The jacobian matrix $\boldsymbol{J}$ can be derived using the shape functions specified in Equations (A.25) by substituting the Cartesian coordinates x, y, and z:

$$\boldsymbol{J} = \begin{bmatrix} \sum_i \frac{\partial N_i'}{\partial \xi} x_i & \sum_i \frac{\partial N_i'}{\partial \xi} y_i & \sum_i \frac{\partial N_i'}{\partial \xi} z_i \\ \sum_i \frac{\partial N_i'}{\partial \eta} x_i & \sum_i \frac{\partial N_i'}{\partial \eta} y_i & \sum_i \frac{\partial N_i'}{\partial \eta} z_i \\ \sum_i \frac{\partial N_i'}{\partial \zeta} x_i & \sum_i \frac{\partial N_i'}{\partial \zeta} y_i & \sum_i \frac{\partial N_i'}{\partial \zeta} z_i \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1'}{\partial \xi} & \frac{\partial N_2'}{\partial \xi} & \dots \\ \frac{\partial N_1'}{\partial \eta} & \frac{\partial N_2'}{\partial \eta} & \dots \\ \frac{\partial N_1'}{\partial \zeta} & \frac{\partial N_2'}{\partial \zeta} & \dots \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \end{bmatrix}. \tag{A.31}$$

In two-dimensional problems, components associated with z and ζ in the transformation matrices can be omitted, simplifying (A.31). This reduction aligns with the fewer spatial dimensions, focusing computations on planar domains.

In parametric element analysis, integrands are best expressed in natural coordinates ξ, η, and ζ. Establishing the relationship between infinitesimal measures in metric and parametric spaces is essential. For instance, this relationship is illustrated through a two-dimensional mapping in Figure A.5. This approach simplifies integration by leveraging the geometric regularity of parametric elements.

In parametric space, the infinitesimal area is defined by the equation:

$$dA' = d\xi d\eta \tag{A.32}$$

Correspondingly, the area in metric space is determined by:

$$dA = (\boldsymbol{e}_\xi \times \boldsymbol{e}_\eta) d\xi d\eta \tag{A.33}$$

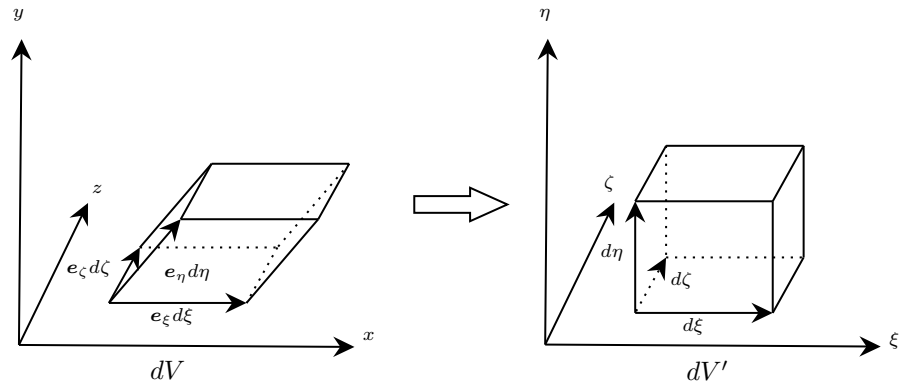


Figure A.6: Parametric mapping of an infinitesimal volume.

where $\boldsymbol{e}_\xi$ and $\boldsymbol{e}_\eta$ are vectors in metric space corresponding to the unit vectors in the ξ and η directions of parametric space:

$$\boldsymbol{e}_\xi = \boldsymbol{i}x_\xi + \boldsymbol{j}y_\xi \tag{A.34a}$$

$$\boldsymbol{e}_\eta = \boldsymbol{i}x_\eta + \boldsymbol{j}y_\eta \tag{A.34b}$$

(A.34c)

Here, $\boldsymbol{i}$ and $\boldsymbol{j}$ are unit vectors in metric space. The two-dimensional Jacobian determinant J_2 is given by:

$$J_2 = \begin{vmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{vmatrix} = x_\xi y_\eta - y_\xi x_\eta. \tag{A.35}$$

Substituting into Equation (A.33), we have:

$$dA = (x_\xi y_\eta - y_\xi x_\eta) d\xi d\eta = J_2 d\xi d\eta. \tag{A.36}$$

In three dimensions, consider a mapping as illustrated in Figure A.6. The infinitesimal volume in metric space can be expressed as:

$$dV = (\boldsymbol{e}_\xi \times \boldsymbol{e}_\eta) \cdot \boldsymbol{e}_\zeta d\xi d\eta d\zeta \tag{A.37}$$

This operation involves both the cross product and dot product, leading to:

$$dV = J_3 d\xi d\eta d\zeta \tag{A.38}$$

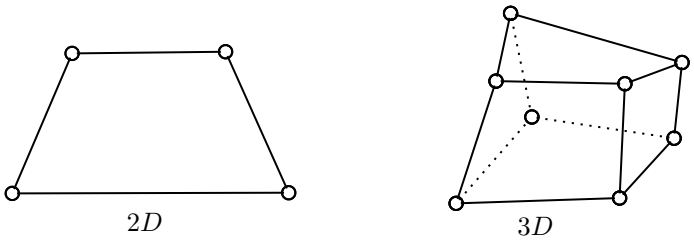


Figure A.7: Distorted elements.

where J_3 is the determinant of the three-dimensional Jacobian matrix. Therefore, integrals such as those in (A.19) and (A.20) can be reformulated as:

$$\int_{\Omega} f d\Omega = \int_{\Omega'} f J d\Omega' \tag{A.39}$$

Here, f represents the integrand, Ω is the metric space, Ω' is the parametric space, and J is the Jacobian determinant. For elements with a linear Cartesian-to-natural coordinate transformation, the Jacobian matrix remains constant. However, for distorted elements (like Figure A.7), the Jacobian matrix may vary or become negative if the distortion is significant. Hence, most finite element analysis software checks for negative Jacobian values and issues warnings if detected.

Bibliography

Ainsworth, M., & Parker, C. (2021a). Mass conserving mixed hp-fem approximations to stokes flow. part ii: Optimal convergence. *SIAM Journal on Numerical Analysis*, *59*(3), 1245–1272.

Ainsworth, M., & Parker, C. (2021b). Mass conserving mixed hp-fem approximations to stokes flow. part i: Uniform stability. *SIAM Journal on Numerical Analysis*, *59*(3), 1218–1244.

Ainsworth, M., & Parker, C. (2022). A mass conserving mixed hp-fem scheme for stokes flow. part iii: Implementation and preconditioning. *SIAM Journal on Numerical Analysis*, *60*(3), 1574–1606.

Anderson, R., Andrej, J., Barker, A., Bramwell, J., Camier, J.-S., Cerveny, J., ... others (2021). Mfem: A modular finite element methods library. *Computers & Mathematics with Applications*, *81*, 42–74.

Bailey, D. H. (2006). Tanh-sinh high-precision quadrature.. Retrieved from `https://api.semanticscholar.org/CorpusID:1700605`

Banerjee, U., & Suri, M. (1992). The effect of numerical quadrature in the p-version of the finite element method. *Mathematics of Computation*, *59*(199), 1. doi: 10.2307/2152977

Bangerth, W., Davydov, D., Heister, T., Heltai, L., Kanschat, G., Kronbichler, M., ... Wells, D. (2016). The deal. ii library, version 8.4. *Journal of Numerical Mathematics*, *24*(3), 135–141.

Barlow, J. (1976, Jan). Optimal stress locations in finite element models. *International Journal for Numerical Methods in Engineering*, *10*(2), 243–251. doi: 10.1002/nme.1620100202

Barlow, J. (1989, Jul). More on optimal stress points—reduced integration, element distortions and error estimation. *International Journal for Numerical Methods in Engineering*, *28*(7), 1487–1504. doi: 10.1002/nme.1620280703

Bathe, K.-J. (1996). *Finite element procedure*. Englewood Cliffs, NJ: Prentice Hall.

Bathe, K.-J. (2001). The inf–sup condition and its evaluation for mixed finite element methods. *Computers & structures*, *79*(2), 243–252.

Bayer, C., & Teichmann, J. (2006, May). The proof of tchakaloff's theorem. *Proceedings of the American Mathematical Society*, *134*(10), 3035–3040. doi: 10.1090/s0002-9939-06-08249-9

Bazeley, G., Cheung, Y., Irons, B., & Zienkiewicz, O. (1966). Triangular elements in bending-conforming and non-conforming solutions. *1st Conf. Matrix Methods in Structural Mechanics*, *AFFDL-TR-66-80*, 547–576.

Beckers, M., & Haegemans, A. (1990, Jan). *The construction of cubature formulae for the tetrahedron* (Report No. TW 128). Department of Computer Science, K. U. Leuven.

Begehr, H., Demidenko, G. V., & Matveeva, I. I. (2021, Feb). An overview of some works of s.l. sobolev. *Complex Variables and Elliptic Equations*, *66*(6–7), 1162–1181. doi: 10.1080/17476933.2021.1882440

Bertrand, F., & Boffi, D. (2024). On the necessity of the inf-sup condition for a mixed finite element formulation. *IMA Journal of Numerical Analysis*, drae002.

Binev, P. (2018). Tree approximation for hp-adaptivity. *SIAM Journal on Numerical Analysis*, *56*(6), 3346–3357.

Bićcanić, N., & Hinton, E. (1979, Jan). Spurious modes in two-dimensional isoparametric elements. *International Journal for Numerical Methods in Engineering*, *14*(10), 1545–1557. doi: 10.1002/nme.1620141009

Briand, E. (2002). *Polynômes multisymétriques* (Unpublished doctoral dissertation). Université Rennes 1.

Briand, E. (2004). When is the algebra of multisymmetric polynomials generated by the elementary multisymmetric polynomials? *Beiträge zur Algebra und Geometrie: Contributions to Algebra and Geometry, 45 (2), 353-368.*.

Cecot, W., Rachowicz, W., & Demkowicz, L. (2003). An hp-adaptive finite element method for electromagnetics. part 3: A three-dimensional infinite element for maxwell's equations. *International journal for numerical methods in engineering*, *57*(7), 899–921.

Chapelle, D., Bathe, K.-J., et al. (2011). *The finite element analysis of shells: fundamentals* (Vol. 1). Springer.

Chen, W., Wang, J., & Zhao, J. (2009). Functions for patch test in finite element analysis of the mindlin plate and the thin cylindrical shell. *Science in China Series G: Physics, Mechanics and Astronomy*, *52*(5), 762–767.

Chen, X., & Hisada, T. (2006). Development of a finite element contact analysis algorithm to pass the patch test. *JSME International Journal Series A Solid Mechanics and Material Engineering*, *49*(4), 483–491.

Chuluunbaatar, G., Chuluunbaatar, O., Gusev, A., & Vinitsky, S. (2022). PI-type fully symmetric quadrature rules on the 3-, ..., 6-simplexes. *Computers & Mathematics with Applications*, *124*, 89-97. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0898122122003364` doi: https://doi.org/10.1016/j.camwa.2022.08.016

Ciarlet, P., & Raviart, P.-A. (1972). The combined effect of curved boundaries and numerical integration in isoparametric finite element methods. *The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations*, 409–474. doi: 10.1016/b978-0-12-068650-6.50020-4

Cools, R. (1997). Constructing cubature formulae: the science behind the art. *Acta Numerica*, *6*, 1–54. doi: 10.1017/s0962492900002701

Cools, R. (1999). Monomial cubature rules since "Stroud": a compilation — part 2. *Journal of Computational and Applied Mathematics*, *112*(1-2), 21–27. doi: 10.1016/S0377-0427(99)00229-0

Cools, R. (2003). An encyclopaedia of cubature formulas. *Journal of Complexity*, *19*(3), 445–453. doi: 10.1016/s0885-064x(03)00011-6

Cools, R., & Haegemans, A. (1994, Jun). An imbedded family of cubature formulae for n-dimensional product regions. *Journal of Computational and Applied Mathematics*, *51*(2), 251–260. doi: 10.1016/0377-0427(92)00007-v

Cools, R., & Rabinowitz, P. (1993). Monomial cubature rules since "Stroud": a compilation. *Journal of Computational and Applied Mathematics*, *48*(3), 309–326. doi: 10.1016/0377-0427(93)90027-9

Cox, D., Little, J., & O'Shea, D. (2013). *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. New York: Springer.

Dalbec, J. P. (1995). *Geometry and combinations of chow forms*. Cornell University.

Davis, P. J. (1967, Oct). A construction of nonnegative approximate quadratures. *Mathematics of Computation*, *21*(100), 578. doi: 10.2307/2005001

de Arantes e Oliveira, E. (1977). The patch test and the general convergence criteria of the finite element method. *International Journal of Solids and Structures*, *13*(3), 159–178. doi: 10.1016/0020-7683(77)90115-9

Demkowicz, L. (2007). *Computing with hp-adaptive finite elements*. Chapman Hall/CRC.

De Veubeke, B. F. (1974, Jan). Variational principles and the patch test. *International Journal for Numerical Methods in Engineering*, *8*(4), 783–801. doi: 10.1002/nme.1620080408

Domokos, M. (2007). Vector invariants of a class of pseudo-reflection groups and multisymmetric syzygies. *arXiv preprint arXiv:0706.2154*.

Domokos, M., & Puskás, A. (2012). Multisymmetric polynomials in dimension three. *Journal of Algebra*, *356*(1), 283–303.

Düster, A., Demkowicz, L., & Rank, E. (2006). High-order finite elements applied to the discrete boltzmann equation. *International journal for numerical methods in engineering*, *67*(8), 1094–1121.

Fix, G. J. (1972). Effects of quadrature errors in finite element approximation of steady state, eigenvalue and parabolic problems**this work was supported in part by the national science foundation under grant no. gp 18064. *The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations*, 525–556. doi: 10.1016/b978-0-12-068650-6.50024-1

Fleischmann, P. (1998). A new degree bound for vector invariants of symmetric groups. *Transactions of the American Mathematical Society*, *350*(4), 1703–1712.

Franke, R. (1971). Orthogonal polynomials and approximate multiple integration. *SIAM Journal on Numerical Analysis*, *8*(4), 757–766. doi: 10.1137/0708070

Fried, I. (1973, Dec). Accuracy and condition of curved (isoparametric) finite elements. *Journal of Sound and Vibration*, *31*(3), 345–355. doi: 10.1016/s0022-460x(73)80278-0

Gatermann, K. (1988). The construction of symmetric cubature formulas for the square and the triangle. *Computing*, *40*(3), 229–240.

Genz, A. C., & Malik, A. A. (1983, Jun). An imbedded family of fully symmetric numerical integration rules. *SIAM Journal on Numerical Analysis*, *20*(3), 580–588. doi: 10.1137/0720038

Golub, G. H., & Welsch, J. H. (1969, Apr). Calculation of gauss quadrature rules. *Mathematics of Computation*, *23*(106), 221. doi: 10.2307/2004418

Hecht, F. (2012). New development in freefem++. *J. Numer. Math.*, *20*(3-4), 251–265. Retrieved from `https://freefem.org/`

Heitzinger, C., Pammer, G., & Rigger, S. (2018). Cubature formulas for multisymmetric functions and applications to stochastic partial differential equations. *SIAM/ASA Journal on Uncertainty Quantification*, *6*(1), 213–242.

Hellen, T. K. (1972, Jul). Effective quadrature rules for quadratic solid isopatametric finite elements. *International Journal for Numerical Methods in Engineering*, *4*(4), 597–599. doi: 10.1002/nme.1620040414

Hinton, E., et al. (1992). Nafems introduction to nonlinear finite element analysis.

Hughes, T. J. (2012). *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation.

Irons, B. (1966). Numerical integration applied to finite element methods. *Conf. on the Use of Digital Computers in Structural Engineering, University of Newcastle*.

Irons, B., & Loikkanen, M. (1983). An engineers' defence of the patch test. *International journal for numerical methods in engineering*, *19*(9), 1391–1401.

Irons, B. M. (1969, Apr). Economical computer techniques for numerically integrated finite elements. *International Journal for Numerical Methods in Engineering*, *1*(2), 201–203. doi: 10.1002/nme.1620010208

Irons, B. M. (1971, Apr). Quadrature rules for brick based finite elements. *International Journal for Numerical Methods in Engineering*, *3*(2), 293–294. doi: 10.1002/nme.1620030213

Irons, B. M., & Razzaque, A. (1972). Experience with the patch test for convergence of finite elements. *The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations*, 557–587. doi: 10.1016/b978-0-12-068650-6.50025-3

Jaśkowiec, J., & Sukumar, N. (2020a). High-order cubature rules for tetrahedra. *International Journal for Numerical Methods in Engineering*, *121*(11), 2418–2436. doi: 10.1002/nme.6313

Jaśkowiec, J., & Sukumar, N. (2020b). High-order symmetric cubature rules for tetrahedra and pyramids. *International Journal for Numerical Methods in Engineering*, *122*(1), 148–171. doi: 10.1002/nme.6528

Junker, F. (1893). Uber symmetrische functionen von mehreren reihen von veränderlichen. *Mathematische Annalen*, *43*(2-3), 225–270.

Kaczmarczyk, Ł., Ullah, Z., Lewandowski, K., Meng, X., Zhou, X.-Y., Athanasiadis, I., ... others (2020). Mofem: An open source, parallel finite element library. *The Journal of Open Source Software*, *5*(45).

Knuth, D. E. (1992). Two notes on notation. *The American Mathematical Monthly*, *99*(5), 403–422. Retrieved from `http://www.jstor.org/stable/2325085`

Kästner, M., Müller, S., Goldmann, J., Spieler, C., Brummund, J., & Ulbricht, V. (2012). Higher-order extended fem for weak discontinuities – level set representation, quadrature and application to magneto-mechanical problems. *International Journal for Numerical Methods in Engineering*, *93*(13), 1403–1424. doi: 10.1002/nme.4435

Lazard, D. (2009). Thirty years of polynomial system solving, and now? *Journal of Symbolic Computation*, *44*(3), 222–231. doi: 10.1016/j.jsc.2008.03.004

Logg, A., Mardal, K.-A., & Wells, G. (2012). *Automated solution of differential equations by the finite element method: The fenics book* (Vol. 84). Springer Science & Business Media.

Lopatin, A., & Reimers, F. (2021). Separating invariants for multisymmetric polynomials. *Proceedings of the American Mathematical Society*, *149*(2), 497–508.

Lyness, J., & Jespersen, D. (1975). Moderate degree symmetric quadrature rules for the triangle. *IMA Journal of Applied Mathematics*, *15*(1), 19-32. doi: 10.1093/imamat/15.1.19

MacMahon, P. A. (1916). *Combinatory analysis*. Univ. Press.

MacNeal, R. H. (1994). *Finite elements: Their design and performance*. New York: Dekker.

Macneal, R. H., & Harder, R. L. (1985). A proposed standard set of problems to test finite element accuracy. *Finite Elements in Analysis and Design*, *1*(1), 3–20. doi: 10.1016/0168-874x(85)90003-4

Maeztu, J. I., & Sainz de la Maza, E. (1995). Consistent structures of invariant quadrature rules for the n–simplex. *Mathematics of Computation*, *64*(211), 1171. doi: 10.2307/2153488

Mantel, F., & Rabinowitz, P. (1977). The application of integer programming to the computation of fully symmetric integration formulas in two and three dimensions. *SIAM Journal on Numerical Analysis*, *14*(3), 391–425. doi: 10.1137/0714024

Milovanović, G. V., & Stanić, M. P. (2014). Numerical integration of highly oscillating functions. *Analytic Number Theory, Approximation Theory, and Special Functions*, 613–649. doi: 10.1007/978-1-4939-0258-3_23

More, J., Garbow, B., & Hillstrom, K. (n.d.). Retrieved from `https://www.netlib.org/minpack/`

Noether, E. (1915, Mar). Der endlichkeitssatz der invarianten endlicher gruppen. *Mathematische Annalen*, *77*(1), 89–92. doi: 10.1007/bf01456821

OEIS Foundation Inc. (2022a). *Entry a001399 in The On-line Encyclopedia of Integer Sequences.* `http://oeis.org/A001399`.

OEIS Foundation Inc. (2022b). *Entry a001400 in The On-line Encyclopedia of Integer Sequences.* `http://oeis.org/A001400`.

Ortner, C. (2012). The role of the patch test in 2d atomistic-to-continuum coupling methods. *ESAIM: Mathematical Modelling and Numerical Analysis*, *46*(6), 1275–1319.

Papanicolopulos, S.-A. (2015). Computation of moderate-degree fully-symmetric cubature rules on the triangle using symmetric polynomials and algebraic solving. *Computers & Mathematics with Applications*, *69*(7), 650-666. doi: https://doi.org/10.1016/j.camwa.2015.02.014

Papanicolopulos, S.-A. (2016a). Efficient computation of cubature rules with application to new asymmetric rules on the triangle. *Journal of Computational and Applied Mathematics*, *304*, 73–83. doi: 10.1016/j.cam.2016.03.013

Papanicolopulos, S.-A. (2016b). New fully symmetric and rotationally symmetric cubature rules on the triangle using minimal orthonormal bases. *Journal of Computational and Applied Mathematics*, *294*, 39–48. doi: 10.1016/j.cam.2015.08.001

Pozrikidis, C. (2005). Finite and spectral element methods using matlab. *University of California San Diego, USA*.

Rabinowitz, P., & Richter, N. (1969). Perfectly symmetric two-dimensional integration formulas with minimal numbers of points. *Mathematics of Computation*, *23*(108), 765–765. doi: 10.1090/s0025-5718-1969-0258281-4

Rachowicz, W., & Demkowicz, L. (2000). An hp-adaptive finite element method for electromagnetics: Part 1: Data structure and constrained approximation. *Computer methods in applied mechanics and engineering*, *187*(1-2), 307–335.

Rachowicz, W., & Demkowicz, L. (2002). An hp-adaptive finite element method for electromagnetics—part ii: A 3d implementation. *International journal for numerical methods in engineering*, *53*(1), 147–180.

Ralston, A. (1965). *A first course in numerical analysis*. McGraw Hill.

Rao, K. M., & Shrinivasa, U. (2001). A set of pathological tests to validate new finite elements. *Sadhana*, *26*, 549–590.

Rees, E. L. (1922). Graphical discussion of the roots of a quartic equation. *The American Mathematical Monthly*, *29*(2), 51–55. doi: 10.1080/00029890.1922.11986100

Renard, Y., & Poulios, K. (2020). Getfem: Automated fe modeling of multiphysics problems based on a generic weak form language. *ACM Transactions on Mathematical Software (TOMS)*, *47*(1), 1–31.

Rigger, S. (2017). *High-dimensional integration: cubature formulas for multisymmetric functions* (Unpublished doctoral dissertation). Wien.

Schläfli, L. (1851). Uber die resultante eines systemes mehrerer algebraischer gleichungen denkschr. *Akad. Wiss*, *4*.

Schläfli, L., et al. (1950). *Gesammelte mathematische abhandlungen* (Vol. 1). Springer.

Schlafli, L. (1852). *Uber die resultante eines systemes mehrerer algebraischer gleichungen: Ein beitrag zur theorie der elimination*. Kaiserlich-Koniglichen Hof- und Staatsdruckerei.

Schneider, T., Hu, Y., Dumas, J., Gao, X., Panozzo, D., & Zorin, D. (2018, Dec). Decoupling simulation accuracy from mesh quality. *ACM Transactions on Graphics*, *37*(6), 1–14. doi: 10.1145/3272127.3275067

Shunn, L., & Ham, F. (2012). Symmetric quadrature rules for tetrahedra based on a cubic close-packed lattice arrangement. *Journal of Computational and Applied Mathematics*, *236*(17), 4348–4364. doi: 10.1016/j.cam.2012.03.032

Sobolev, S. (1962). Formulas of mechanical cubature on the surface of a sphere. *Sibirsk.Math.Z.*, *3*, 769–796.

Solin, P., Segeth, K., & Dolezel, I. (2003). *Higher-order finite element methods*. Chapman and Hall/CRC.

Srinivasan, R., Dattaguru, B., & Singh, G. (2020). Coupled field formulation of exactly integrated membrane finite elements insensitive to distortion. *International Journal for Computational Methods in Engineering Science and Mechanics*, *22*(1), 60–80.

Strang, G. (1972). Variational crimes in the finite element method. *The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations*, 689–710. doi: 10.1016/b978-0-12-068650-6.50030-7

Stroud, A. H. (1971). *Approximate calculation of multiple integrals*. Englewood Cliffs: Prentice-Hall.

Stummel, F. (1979). The generalized patch test. *SIAM Journal on Numerical Analysis*, *16*(3), 449–471. doi: 10.1137/0716037

Talischi, C., & Paulino, G. H. (2014). Addressing integration error for polygonal finite elements through polynomial projections: A patch test connection. *Mathematical Models and Methods in Applied Sciences*, *24*(08), 1701–1727. doi: 10.1142/s0218202514400077

Taylor, R. L., Beresford, P. J., & Wilson, E. L. (1976, Jan). A non-conforming element for stress analysis. *International Journal for Numerical Methods in Engineering*, *10*(6), 1211–1219. doi: 10.1002/nme.1620100602

Taylor, R. L., Simo, J. C., Zienkiewicz, O. C., & Chan, A. C. (1986). The patch test—a condition for assessing fem convergence. *International Journal for Numerical Methods in Engineering*, *22*(1), 39–62. doi: 10.1002/nme.1620220105

Vaccarino, F. (2005). The ring of multisymmetric functions. *Annales de l'institut Fourier*, *55*(3), 717–731. doi: 10.5802/aif.2111

Vinberg, E. (2003). *A course in algebra, in: Graduate studies in mathematics* (Vol. 56). Providence, Rhode Island: American Mathematical Society.

Wang, W., & Papanicolopulos, S.-A. (2023). Explicit consistency conditions for fully symmetric cubature on the tetrahedron. *Engineering with Computers*. doi: 10.1007/s00366-023-01845-4

Weyl, H. (1946). *The classical groups their invariants and representations*. Princeton Univ. Press.

Williams, D., Shunn, L., & Jameson, A. (2014, Aug). Symmetric quadrature rules for simplexes based on sphere close packed lattice arrangements. *Journal of Computational and Applied Mathematics*, *266*, 18–38. doi: 10.1016/j.cam.2014.01.007

Witherden, F. D., & Vincent, P. E. (2015). On the identification of symmetric quadrature rules for finite element methods. *Computers & Mathematics with Applications*, *69*(10), 1232–1241. doi: 10.1016/j.camwa.2015.03.017

Xiao, H., & Gimbutas, Z. (2010a). A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Computers & Mathematics with Applications*, *59*(2), 663–676. doi: 10.1016/j.camwa.2009.10.027

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

Yosibash, Z. (2012). p-fems in biomechanics: Bones and arteries. *Computer Methods in Applied Mechanics and Engineering*, *249*, 169–184.

Zander, N., Bog, T., Kollmannsberger, S., Schillinger, D., & Rank, E. (2015). Multi-level hp-adaptivity: high-order mesh adaptivity without the difficulties of constraining hanging nodes. *Computational Mechanics*, *55*(3), 499–517.

Zhang, L., Cui, T., & Liu, H. (2009). A set of symmetric quadrature rules on triangles and tetrahedra. *Journal of Computational Mathematics*, *64*(1), 89-96.

Zhang, W., & Chen, D. P. (1997). The patch test conditions and some multivariable finite element formulations. *International Journal for Numerical Methods in Engineering*, *40*(16), 3015–3032. doi: 10.1002/(sici)1097-0207(19970830)40:16<3015::aid-nme184>3.0.co;2-1

Zienkiewicz, O., & Taylor, R. (1997). The finite element patch test revisited a computer test for convergence, validation and error estimates. *Computer Methods in Applied Mechanics and Engineering*, *149*(1–4), 223–254. doi: 10.1016/s0045-7825(97)00085-6

Zienkiewicz, O. C. (1971). *The finite element method in engineering science*. McGraw-Hill.

Zienkiewicz, O. C., Qu, S., Taylor, R. L., & Nakazawa, S. (1986, Oct). The patch test for mixed formulations. *International Journal for Numerical Methods in Engineering*, *23*(10), 1873–1883. doi: 10.1002/nme.1620231007

Zienkiewicz, O. C., Taylor, R. L., & Zhu, J. Z. (2013). *The finite element method: Its basis and fundamentals*. Amsterdam: Elsevier. doi: https://doi.org/10.1016/C2009-0-24909-9