

## Generalized Gaussian Quadrature Rules on Arbitrary Polygons

S. E. Mousavi<sup>1</sup>, H. Xiao<sup>2</sup> and N. Sukumar<sup>1,\*</sup>

<sup>1</sup> *Department of Civil and Environmental Engineering, University of California, Davis, CA 95616.*

<sup>2</sup> *Department of Mathematics, University of California, Davis, CA 95616.*

### SUMMARY

In this paper, we present a numerical algorithm based on group theory and numerical optimization to compute efficient quadrature rules for integration of bivariate polynomials over arbitrary polygons. These quadratures have desirable properties such as positivity of weights and interiority of nodes and can readily be used as software libraries where numerical integration within planar polygons is required. We have used this algorithm for the construction of symmetric and non-symmetric quadrature rules over convex and concave polygons. While in the case of symmetric quadratures our results are comparable to available rules, the proposed algorithm has the advantage of being flexible enough so that it can be applied to arbitrary planar regions for the integration of generalized classes of functions. To demonstrate the efficiency of the new quadrature rules, we have tested them for the integration of rational polygonal shape functions over a regular hexagon. For a relative error of  $10^{-8}$  in the computation of stiffness matrix entries, one needs at least 198 evaluation points when the region is partitioned, whereas 85 points suffice with our quadrature rule. Copyright © 2009 John Wiley & Sons, Ltd.

---

\*Correspondence to: N. Sukumar, Department of Civil and Environmental Engineering, University of California, One Shields Avenue, Davis, CA 95616. E-mail: nsukumar@ucdavis.edu

Contract/grant sponsor: NSF; contract/grant number: CMMI-0626481, DMS-0811025, DMS-0513069

KEY WORDS: Gaussian quadrature, numerical integration, Gram-Schmidt orthogonalization, Newton's method, convex and concave polygons, polygonal finite elements

## 1. INTRODUCTION

With the development of increasingly more sophisticated mathematical models and computational algorithms, numerical integration of functions in general domains in two and higher dimensions is commonly required in fields such as engineering and the applied sciences. Gaussian quadratures have been extensively used for numerical integration of functions where exact integration is not possible.

Quadrature theory in one dimension is relatively complete, with Gaussian quadratures being optimal for integration of polynomials, and their construction well understood (for example, see References [1–3] for construction of generalized Gaussian quadratures on the interval). In higher dimensions, however, the situation is considerably more complex. While the interval is the only connected compact subset of  $\mathbb{R}$ , regions of  $\mathbb{R}^2$  come in an infinite variety of shapes, each with its own topological features [4]. As would be expected, one might attempt to generalize one-dimensional quadrature rules to higher dimensions using tensor product. However, this approach is far from optimal in terms of the number of quadrature nodes needed for achieving a pre-selected desirable precision. It seems likely that generalized Gaussian quadratures in higher dimensions have to be studied separately for different geometries of integration regions, and that each region should require a different set of rules, corresponding to its own symmetry features.

The geometry that has been studied the most extensively in two dimensions is the triangle,

since it is a standard tool for representing higher dimensional regions and surfaces. Moderate- to high-order efficient quadrature rules have been developed for polynomials on the triangle [4–12]. However, when elements with five vertices or more are used in polygonal finite element applications [13–16] one has to partition the element into triangular elements for the sake of numerical integration. The partitioning of the element and use of adaptive integration schemes increases programming complexity and computational cost.

Different techniques have been employed for the computation of quadrature rules on polygons. For instance, moment fitting equations that contain integration of the basis functions over the region, have been used to find quadrature rules over the triangle [7, 10]. Lyness and Monegato [17] have used a polar variation of moment fitting equations to construct quadrature rules over regions having the same symmetry as the regular hexagon ( $D_6$  regions). By expressing the basis functions in polar coordinates and taking advantage of the symmetry properties of the region, many of the moment equations are automatically satisfied and it is possible to solve the non-linear set of equations for moderate degree quadrature rules. They find proper quadrature rules with points inside the hexagon and positive weights for polynomials up to order 13. This approach can be extended to other regular polygons with different number of edges and also regions with different types of symmetry. Nooijen et al. [18] have proposed a method in which the points of a quadrature rule are found as the common zeros of two polynomials that satisfy certain symmetry and orthogonality requirements over regular polygons and the corresponding weights are the solution of a linear system of equations. The aforementioned techniques are powerful when the goal is to find symmetric quadratures, but in the case of non-symmetric quadrature rules over non-symmetric regions, they are not easy to implement.

Xiao and Gimbutas [19] presented a numerical algorithm for the construction of efficient quadrature rules that is a combination of a new point-elimination scheme with Newton's method. These quadratures have many desirable properties such as positivity of weights, interiority of nodes, and different symmetry features. This algorithm was successfully applied to the triangle and square [19]. In this paper, we use the same algorithm to construct efficient quadrature rules for bivariate polynomials over convex and concave polygons. In case of regular polygons, quadrature rules with different symmetry features are discussed. Although the comparison of our results with the best available symmetric quadrature rules shows the same or fewer number of nodes for a prescribed accuracy, the main contribution of this paper is to introduce a flexible and general algorithm that can be used for arbitrary polygons with generalized classes of functions that arise in meshfree, isogeometric, and enriched finite element methods, while being efficient with respect to the number of nodes required.

The outline of the remainder of this paper is as follows. Section 2 summarizes the mathematical preliminaries that are used in the construction of generalized Gaussian quadratures. The algorithm for the construction of efficient quadratures is discussed in Section 3. In Section 4.1, we present symmetric and non-symmetric quadrature rules over regular polygons with five to eight vertices that are constructed using our algorithm. Tabulated results for the locations and weights of the Gauss points are only provided for a regular hexagon for orders 1 to 5. Tables showing the number of evaluation points for the obtained quadrature rules for these regular polygons are included and comparison is made whenever a similar quadrature is available. Section 4.2 contains some examples of moderate degree quadrature rules over arbitrary convex and concave polygons. In Section 4.3, the computed quadrature rules are used to integrate mass and stiffness matrix entries that arise in polygonal finite

element method with Laplace shape functions and the results are compared to numerical integration by partitioning the polygon. Finally, a few concluding remarks are made in Section 5.

## 2. MATHEMATICAL PRELIMINARIES

### 2.1. Quadrature rules

A quadrature is a formula of the form

$$\int_{\Omega} \omega(\mathbf{x})f(\mathbf{x}) d\mathbf{x} \approx \sum_{i=1}^n w_i f(\mathbf{x}_i), \quad (1)$$

where  $\Omega$  is the designated integration region,  $f$  is an integrand defined on  $\Omega$  and  $\omega$  is the weight function.  $\omega(\mathbf{x}) = 1$  is used for the quadratures designed in this paper. The points  $\mathbf{x}_i$  are called quadrature nodes, and  $w_i$  are the quadrature weights. Typically, quadratures are designed so that (1) is exact for all functions in a pre-selected set (see (2)). Classical choices of the pre-selected set of functions include polynomials up to a certain degree, trigonometric functions, and basis functions of a particular function space defined on  $\Omega$ . Once (2) is solved, the resulting  $\{\mathbf{x}_i, w_i\}_{i=1}^n$  form a quadrature rule for that specific domain of integration and class of basis functions.

$$\begin{pmatrix} \int_{\Omega} \omega(\mathbf{x})\phi_1(\mathbf{x}) d\mathbf{x} \\ \int_{\Omega} \omega(\mathbf{x})\phi_2(\mathbf{x}) d\mathbf{x} \\ \vdots \\ \int_{\Omega} \omega(\mathbf{x})\phi_m(\mathbf{x}) d\mathbf{x} \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \dots & \phi_1(\mathbf{x}_n) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \dots & \phi_2(\mathbf{x}_n) \\ \vdots & \vdots & \dots & \vdots \\ \phi_m(\mathbf{x}_1) & \phi_m(\mathbf{x}_2) & \dots & \phi_m(\mathbf{x}_n) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}. \quad (2)$$

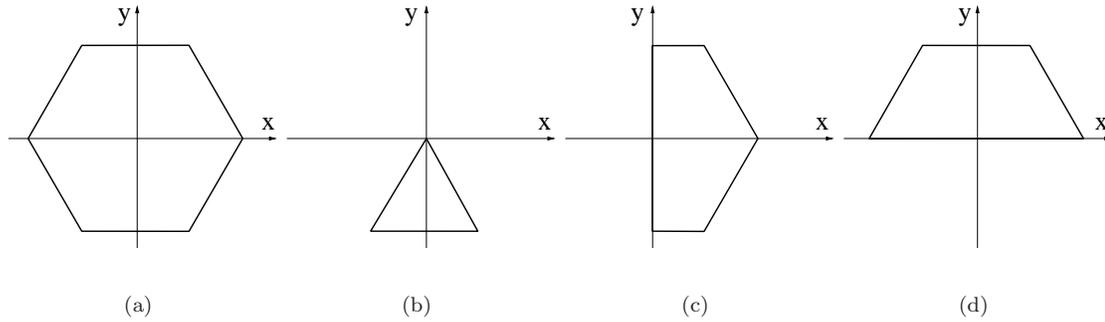


Figure 1. Symmetric generators of the regular hexagon on a unit circle. (a) Identity (no symmetry) (b) Rotational symmetry (c) Mid-edge reflection symmetry and (d) Vertex reflection symmetry.

## 2.2. Symmetry groups, symmetry operations and symmetric basis functions

The symmetry group  $G$  of a region  $\Omega$  is the set of orthogonal linear transformations that map  $\Omega$  onto itself. For example, the regular hexagon is invariant under rotations by  $\frac{\pi}{3}$  radians about its center and is also invariant under a reflection about a diameter connecting two opposite vertices. A two-dimensional region having these characteristics is sometimes known as a  $D_6$ -region [17].

Two points  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are  $S$ -symmetrically related if there is an element  $s \in S$  ( $S \subseteq G$  is a symmetry subgroup of  $\Omega$ ) such that  $\mathbf{x}_2 = s\mathbf{x}_1$ . The set of all points that are  $S$ -symmetrically related with  $\mathbf{x}$  is called the orbit of  $\mathbf{x}$ . If a set of points contains one and only one point from each orbit of  $\Omega$  with respect to each member of symmetry subgroup  $S$ , then the set is called an  $S$ -generator of  $\Omega$ . For example, Figure 1 shows some of the  $S$ -generators of a regular hexagon.

The concept of symmetry can also be applied to function spaces and function sets. The operation of  $s \in S$  on a function  $\phi$  is defined as  $s\phi(\mathbf{x}) = \phi(s^{-1}\mathbf{x})$ . For basis functions  $\{\phi_i\}_{i=1}^m$  over a region  $\Omega$ , the functions  $\Phi_i$  defined by the formula

$$\Phi_i = \sum_{s \in S} s\phi_i \quad (3)$$

are  $S$ -invariant [4], i.e., invariant under all operations  $s \in S$ . Furthermore,  $\Phi_i$  either vanishes or is a non-zero basis function over the  $S$ -generator. The following example illustrates the application of (3) for construction of basis functions over the  $S$ -generators.

*Example of a symmetric basis function:* Consider the monomial  $\phi = x^2$ , which can be used as one of the basis functions over the regular hexagon. Also consider  $S$ , a symmetry subgroup of regular hexagon consisting of rotations about the origin by  $\{k\pi/3\}_{k=0}^5$ . The  $S$ -generator of this symmetry subgroup is shown in Figure 1b. Each element  $s \in S$  can be represented by the transformation

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} \cos(\frac{k\pi}{3}) & -\sin(\frac{k\pi}{3}) \\ \sin(\frac{k\pi}{3}) & \cos(\frac{k\pi}{3}) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

Application of (3) to  $\phi$  produces  $\Phi = 3(x^2 + y^2)$ .  $\Phi$  is  $S$ -invariant and can be used as a basis function for calculation of quadrature rules on the  $S$ -generator.

The set of  $S$ -invariant orthogonal basis functions can be constructed recursively via a Gram-Schmidt procedure starting from the set of  $S$ -invariant basis functions or be drawn from classical results [20]. If the former method is used, symbolic software packages such as **Mathematica**<sup>™</sup> and **Maple**<sup>™</sup> are well-suited to obtain the basis functions analytically. It is noteworthy that while the total number of orthogonal bivariate basis functions up to order  $d$  over a regular polygon is  $(d+1)(d+2)/2$ , one ends up with considerably fewer number of orthogonal basis functions when rotational or reflection symmetry subgroups are used.

With the above concepts in mind, one can define a symmetric quadrature as follows. Given the geometric region  $\Omega$  and a symmetry subgroup  $S$ , the quadrature with nodes  $\{\mathbf{x}_i\}_{i=1}^p$  and corresponding weights  $\{w_i\}_{i=1}^p$  is  $S$ -symmetric if the set of the nodes is  $S$ -symmetric and all nodes that are  $S$ -symmetrically related have the same weight. It is obvious that finding the nodes of an  $S$ -symmetric quadrature and the corresponding weights over the  $S$ -generator is

enough for the construction of the quadrature over the entire region.

Different symmetry operations can be imposed on the basis functions which are used to construct the quadrature rule in order to reduce the number of moment equations that need to be solved. However, symmetry of the quadratures is not a requirement of the algorithm presented in this paper and it can be applied to non-symmetric regions as well.

### 2.3. Newton's method

Having the orthonormal basis functions over the requisite domain of integration, (2) is a non-linear equation that can be solved to get  $n$  nodes  $\mathbf{x}_i$  and the corresponding weights  $w_i$ . Let the vector  $\mathbf{X} = (x_1^1, x_1^2, \dots, x_1^d, w_1, \dots, x_n^1, x_n^2, \dots, x_n^d, w_n)^T$ , which contains  $n' = n(d+1)$  unknowns in  $\mathbb{R}^d$ , and  $\{x_i^j\}_{j=1}^d$  are the coordinates of the  $i$ -th node. Newton's method is a classical tool for solving non-linear equations with the limitation that it converges quadratically only locally for an initial guess close enough to the final solution. Once we have an appropriate initial guess  $\mathbf{X}_k$ , (4) is used to get a better solution  $\mathbf{X}_{k+1}$  until convergence is attained within the acceptable error range:

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{J}^\dagger(\mathbf{X}_k) \mathbf{F}(\mathbf{X}_k), \quad (4)$$

where  $\mathbf{J}^\dagger(\mathbf{X}_k)$  is the pseudo-inverse [21] of the Jacobian  $\mathbf{J} : \mathbb{R}^{n'} \rightarrow \mathbb{R}^m$  defined by (5) for  $m$  basis functions and  $n'$  degrees of freedom:

$$\mathbf{J}(\mathbf{X}) = \begin{pmatrix} \frac{\partial f_1}{\partial X_1}(\mathbf{X}) & \dots & \frac{\partial f_1}{\partial X_{n'}}(\mathbf{X}) \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial X_1}(\mathbf{X}) & \dots & \frac{\partial f_m}{\partial X_{n'}}(\mathbf{X}) \end{pmatrix}. \quad (5)$$

For our specific case of finding quadrature rules,  $\mathbf{F}$  is a non-linear function of the coordinates of quadrature nodes and the corresponding weights:

$$\mathbf{F}(\mathbf{X}) = \begin{pmatrix} f_1(\mathbf{X}) \\ \vdots \\ f_m(\mathbf{X}) \end{pmatrix}, \quad (6)$$

where each of the functions  $f_i$  is defined using (2) by moving one side over and setting the rows to zero.

### 3. ALGORITHM FOR CONSTRUCTION OF EFFICIENT QUADRATURES

The algorithm used in this paper is based on the observation that an  $n$ -point generalized Gaussian quadrature may be approximated by deleting an appropriate point in an  $(n + 1)$ -point quadrature. To be more specific, suppose that the quadrature  $\{\mathbf{x}_i, w_i\}_{i=1}^{n+1}$  has one weight  $w_k = 0$ , then we have also obtained an  $n$ -point quadrature  $\mathbf{x}_1, \dots, \mathbf{x}_{k-1}, \mathbf{x}_{k+1}, \dots, \mathbf{x}_{n+1}$  and  $w_1, \dots, w_{k-1}, w_{k+1}, \dots, w_{n+1}$ . If none of the weights are zero, deleting any one node, which is equivalent to setting the corresponding weight to zero, will make the quadrature only an approximation to an accurate quadrature. However, if the approximation is close enough to a true quadrature configuration, Newton's method can be used to obtain an  $n$ -point quadrature. One advantage of this method is that, with each successful convergence of Newton's method, a new quadrature can be found with one fewer point with respect to the previous one.

To begin this process, one has to find an initial quadrature that will integrate exactly all polynomials up to order  $d$  on  $\Omega$ . For triangles or squares, the most straightforward quadrature can be obtained from the tensor product of one-dimensional Gaussian quadratures for the interval. For polygons of edges greater than four, the initial quadrature may be built similarly,

or can be built on split pieces of the integration domain. In addition, suppose that  $P_d$  denotes all polynomials of degree up to  $d$  on  $\Omega$ , it has been shown that any  $n = \dim(P_d)$  nodes  $\mathbf{x}_i$  in  $\Omega$  can serve as a set of quadrature nodes if there exists a linearly independent set of bounded functions  $g_j, j = 1, \dots, n$  defined on  $\Omega$  and that the matrix  $\mathbf{T} = (t_{ij})$  with  $t_{ij} = g_j(\mathbf{x}_i)$  is non-singular. Furthermore, quadrature weights  $w_i$  can be determined by solving (2) with  $w_i$  not assuming large negative values [22].

In our study, we start with the tensor product of a one-dimensional  $n$ -point Gauss quadrature rule and solve (2) iteratively for the orthonormal basis functions until we get a desirable result, referred to as the corrected tensor product, which can serve as an initial quadrature. Since  $n$  is chosen large enough so that we have a large number of redundant degrees of freedom, Newton's method converges in a few iterations. After computing the initial quadrature, the following matrix is formed:

$$\mathbf{A}_{j,k} = \sqrt{w_k} \phi_j(\mathbf{x}_k), \quad j = 1, \dots, m \text{ and } k = 1, \dots, n^d \quad (7)$$

with  $\{x_k, w_k\}$  being the quadrature nodes and weights and  $\phi_j$  are the orthonormal basis functions. The inner product of rows  $i$  and  $j$  of  $\mathbf{A}$  ( $\forall i, j = 1, \dots, m$ ) gives the inner product of the corresponding orthonormal basis functions over the prescribed configurations. Knowing that the basis functions are already orthogonalized with respect to each other, one finds that  $\mathbf{A}$  has orthonormal rows. Since there are  $m$  linearly independent basis functions, there are at least  $m$  columns that are linearly independent and correspond to the selection of  $m$  nodes in  $\Omega$ . We start our node elimination process with only  $m$  nodes of the quadrature whose related columns are the most orthogonal with each other instead of the whole initial rule, thereby reducing the required computational effort.

What remains is to generate quadratures with reduced number of nodes one by one, until Newton's method does not converge anymore. The efficiency of a quadrature rule can be evaluated considering the lower bound on the number of points required by a quadrature rule of polynomial degree  $d$  for any two dimensional region given by  $(d/2 + 1)(d/2 + 2)/2$  [23] for even  $d$ . It is also proved that for odd degrees of polynomials  $d = 2k + 1$ , the number of evaluation points are more than or equal to the rule for degree  $d = 2k$ .

We tend to eliminate the nodes that are outside the domain or have negative weights first so that the resulting quadrature is proper, namely all nodes are inside the domain and all weights are positive. Non-negative weights lend stability to the quadrature rule by preventing round-off errors [24]. Xiao and Gimbutas [19] introduce a significance index for each quadrature node (see (8)) that can be considered as a measure of contribution of the node in the evaluation of the integral over the domain. If there are no nodes outside the domain, then the one with the least significance is eliminated. The significance index of node  $j$  is defined as

$$s_j = w_j \sum_{i=1}^n \phi_i^2(\mathbf{x}_j). \quad (8)$$

In our algorithm negative weights are not allowed; however if negative weights are permitted, then it's conceivable that a different node elimination criteria can be used.

The numerical algorithm can be summarized as follows:

1. Find an initial quadrature  $\{\mathbf{x}_i, w_i\}$  for the region of integration  $\Omega$ , the class of basis functions  $\phi_j$  and the appropriate weight function  $\omega(\mathbf{x})$  that satisfies (2).
2. Eliminate one of the nodes (e.g., the one with the minimum significance factor).
3. Solve (2) iteratively until convergence is attained. The resulting quadrature now has one fewer node than the initial one. Continue the above elimination procedure until no

additional node can be removed (i.e., Newton iterations do not converge). This is the final quadrature rule.

## 4. NUMERICAL RESULTS

### *4.1. Moderate degree quadratures over regular polygons*

We limit the numerical results to the construction of generalized Gaussian quadrature rules on two-dimensional planar polygons, but the numerical algorithm presented in Section 3 is general and can be applied to polytopes in higher dimensions as well.

Since the geometry of regular polygons is known beforehand, we broke down the process of computing efficient quadrature rules over regular polygons into two stages: first, finding the orthonormal basis functions over different symmetric generators and then implementing node elimination algorithm using the appropriate set of basis functions and a proper choice of initial quadrature. The accuracy of orthogonality of the basis functions has a significant effect on the stability of the node elimination algorithm and the accuracy of the resulting quadratures. The first stage is implemented in **Maple**, which is capable of carrying out symbolic calculations exactly for a configuration like hexagon. The numerical algorithm for computing quadratures and the node elimination process is implemented in **MATLAB**<sup>TM</sup>. The entire node elimination process for quadratures with orders lower than five takes a few seconds and for order five and higher, takes a few minutes on a Linux workstation.

The node elimination algorithm discussed in Section 3 proves to be very heuristic in the sense that the choice of initial quadrature and manipulation of initial and intermediate ones can result in very distinct final quadratures with different number of nodes, nodal coordinates

and weights. Some choices of initial quadratures may end up with quadrature nodes outside the domain or with negative weights. In some cases we managed to perturb an improper quadrature by changing the coordinates and weights of some of its nodes and obtain a proper rule by solving (4) again. In cases where such a remedy could not be found, the best intermediate quadrature with the fewest number of nodes was chosen as the final rule. At the end, a final tuning of the obtained quadratures is performed in `Maple` with higher precision arithmetic to ensure the correctness of all the digits of the quadrature rules that are reported herein.

Table I shows the obtained number of nodes for the quadratures up to order 10 over regular polygons with five, six, seven and eight edges (non-symmetric quadratures). The lower bound of  $numx$  is also provided. In all cases the number of evaluation points are equal or very close to the lower bound. Symmetric quadratures tend to need more evaluation points for a specific geometry and polynomial degree than non-symmetric ones. Unfortunately the authors could not find similar non-symmetric rules to compare with. Table II shows similar results for the hexagon for different symmetric generators for bivariate polynomials up to order 20. A comparison is made with the best available symmetric quadrature rules (Lyness and Monegato [17]) over the hexagon, which reveals that our algorithm is capable of producing the same quadrature rules. However, our symmetric quadrature rules over the hexagon have fewer number of evaluation points compared to those obtained by Nooijen et al. [18] for the same kind of symmetry. Table III shows quadrature rules for the entire hexagon (no symmetry) that can integrate polynomials up to order  $d$  ( $d = 1, 2, \dots, 5$ ). All the nodes are inside the domain and all the weights are positive. As a measure of accuracy of the obtained quadratures, we have tested them on all monomials of type  $x^i y^j$ , where  $i + j \leq d$  and exact integration is also available. The error reported in Table III is the maximum absolute error of integration of

all the binomials with the given quadrature rule. Although the quadratures reported here are accurate only up to 16 decimal places, more accurate quadratures (with 32 correct digits or more) are also computed and available. Figures 2a and 2b show the choice of initial quadrature and the final quadrature on the hexagon for polynomials of order 10. The initial quadrature is a  $11 \times 11$  tensor product and the final quadrature has 23 nodes. Figures 2c to 2h show quadrature rules of order 10 for different symmetric generators of the hexagon and Figures 3a to 3f show non-symmetric quadrature rules of order 10 for polygons with five, seven and eight edges, respectively. All quadrature rules of order up to 10 over regular polygons with five to eight number of edges (in case of hexagon, with the mentioned symmetry types) are computed and available upon request, but not included in this paper for brevity.

#### *4.2. Lower degree quadrature rules over convex and concave polygons*

Sommariva and Vianello [25] have used Green's integral formula to produce Gauss-like quadrature rules that can integrate high-order bivariate polynomials exactly. These quadrature rules can be used for convex and concave polygons, but it is not guaranteed that the integration points always fall inside the polygon, and furthermore the resulting quadrature rules require a relatively large number of nodes (e.g., hundreds of integration points for an accuracy of order ten) to compensate for the arbitrariness of the integration domain. The algorithm that we present in this paper has the flexibility of being applicable to arbitrary polygons, while at the same time achieves near minimal number of integration points.

Since symmetry of the integration domain is not a requirement of the node elimination algorithm, we illustrate its versatility by constructing quadrature rules on convex and concave polygons. In Figure 4, we present quadrature rules over sample convex polygons with five to

eight number of edges and for polynomial degrees 3, 5, and 7. We also constructed quadrature rules of orders 3, 5 and 7 over concave polygons, and the quadrature rules on a concave hexagon are shown in Figure 5.

#### 4.3. Application in polygonal finite element methods

Laplace shape functions are used in polygonal finite element methods and have many desirable properties such as partition of unity, linear completeness and compact support [14]. Figure 6 shows Laplace shape function plots for a node of a regular hexagon. To compute the mass and stiffness matrix entries in Galerkin finite element methods, the product of these shape functions or their derivatives must be integrated over the element. Since these shape functions are non-polynomial, higher-order quadrature rules are needed to obtain sufficient accuracy. Traditionally, one partitions the element and applies quadrature rules over each triangle to carry out the integration [14]. Application of quadrature rules over the polygonal element allows us to use a simple quadrature rule over the entire domain and hence significantly reduces the number of evaluation points.

In Figure 7, convergence curves for computing some of the mass and stiffness matrix entries over the regular hexagon ( $\Omega_0$ ) are presented. Shape functions  $\phi_1$  and  $\phi_2$  correspond to adjacent nodes of the hexagon. The exact (reference) values are calculated by partitioning the hexagon and using a very high-order quadrature rule (1600 evaluation points) over each triangle. The accuracy of our quadrature rules on  $\Omega_0$  are compared to those obtained by partitioning  $\Omega_0$  into triangles and using quadrature rules within each triangle [10]. In Reference [14], 150 evaluation points are used over the hexagon to compute stiffness matrix entries. As seen in Figure 7, for a relative error of  $10^{-8}$  one needs 150 evaluation points for mass matrix entries and 198

points for stiffness matrix entries when integration is done by partitioning, with quadrature rules of orders 10 and 12 being used over each triangle, respectively. On using our symmetric quadrature rules over the hexagon, only 72 and 85 points are needed for the mass and stiffness matrix entries, respectively, to realize the same accuracy.

## 5. CONCLUSIONS

We presented an algorithm based on numerical optimization for construction of efficient quadrature rules in 2D. This algorithm was combined with group theory to produce symmetric quadrature rules. The resulting quadratures have the desirable properties of interiority of nodes and positivity of weights. The algorithm was successfully applied to regular polygons with five to eight number of edges for bivariate polynomials of order up to ten. Efficient symmetric quadrature rules of higher-orders such as 15, 18 and 20 were also constructed. These quadrature rules were used to compute mass and stiffness matrix entries in polygonal finite element methods—for a relative error of  $10^{-8}$  in the computation of stiffness matrix entries, at least 198 evaluation points are needed when the region is partitioned, whereas with our quadrature rules 85 points suffice.

Quadrature rules on regular polygons can be presented as software libraries and readily used in codes where integration over polygons is needed [13–16,26]. The node elimination algorithm is very flexible, and to illustrate its benefits, quadrature rules on convex and concave polygons were also presented. Even though the numerical tests that we have conducted have led to convergence to proper quadratures, it is noteworthy to point out that the node elimination algorithm is heuristic, and we do not have a formal proof for the existence of quadrature rules with nodes that lie within the polygon and have positive weights. Our results motivate further

studies in this direction, and more broadly on generalized quadrature rules for non-polynomial basis functions on triangles [27] and other polygons.

#### ACKNOWLEDGEMENTS

S. E. Mousavi and N. Sukumar acknowledge the research support of the National Science Foundation through contract grants CMMI-0626481 and DMS-0811025 to the University of California at Davis. H. Xiao acknowledges the research support of the National Science Foundation through grant DMS-0513069.

## REFERENCES

1. A. H. Stroud and D. Secrest. *Gaussian Quadrature Formulas*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1966.
2. J. Ma, V. Rokhlin, and S. Wandzura. Generalized Gaussian quadrature of systems of arbitrary functions. *SIAM Journal on Numerical Analysis*, 33(3):971–996, 1996.
3. N. Yarvin and V. Rokhlin. Generalized Gaussian quadratures and singular value decompositions of integral operators. *SIAM Journal on Scientific Computing*, 20(2):669–718, 1998.
4. S. Wandzura and H. Xiao. Symmetric quadrature rules on a triangle. *Computers and Mathematics with Applications*, 45:1829–1840, 2003.
5. P. Silvester. Symmetric quadrature formulae for simplexes. *Mathematics of Computation*, 24(109):95–100, 1970.
6. G. R. Cowper. Gaussian quadrature formulas for triangles. *International Journal for Numerical Methods in Engineering*, 7(3):405–408, 1973.
7. J. N. Lyness and D. Jespersen. Moderate degree symmetric quadrature rules for the triangle. *Journal of the Institute of Mathematics and its Applications*, 15:19–32, 1975.
8. P. Hillion. Numerical integration on a triangle. *International Journal for Numerical Methods in Engineering*, 11:797–815, 1977.
9. M. E. Laursen and M. Gellert. Some criteria for numerically integrated matrices and quadrature formulas for triangles. *International Journal for Numerical Methods in Engineering*, 12:67–76, 1978.
10. D. A. Dunavant. High degree efficient symmetrical Gaussian quadrature rules for the triangle. *International Journal for Numerical Methods in Engineering*, 21:1129–1148, 1985.
11. J. Berntsen and T. O. Espelid. Degree 13 symmetric quadrature rules for the triangle. Technical Report 44, Department of Informatics, University of Bergen, Norway, 1990.
12. H. T. Rathod, K. V. Nagaraja, and B. Venkatesudu. Symmetric Gauss Legendre quadrature formulas for composite numerical integration over a triangular surface. *Applied Mathematics and Computation*, 188:865–876, 2007.
13. G. Dasgupta. Integration within polygonal finite elements. *Journal of Aerospace Engineering*, 16(1):9–18, 2003.
14. N. Sukumar and A. Tabarraei. Conforming polygonal finite elements. *International Journal for Numerical Methods in Engineering*, 61(12):2045–2066, 2004.

15. A. Tabarraei and N. Sukumar. Extended finite element method on polygonal and quadtree meshes. *Computer Methods in Applied Mechanics and Engineering*, 197(5):425–438, 2008.
16. C. Talischi, G. H. Paulino, and C. Le. Honeycomb Wachspress finite elements for structural topology optimization. *Structural and Multidisciplinary Optimization*, 37:569–583, 2009.
17. J. N. Lyness and G. Monegato. Quadrature rules for regions having regular hexagonal symmetry. *SIAM Journal on Numerical Analysis*, 14(2):283–295, 1977.
18. M. Nooijen, G. T. Velde, and E. J. Baerends. Symmetric numerical integration formulas for regular polygons. *SIAM Journal on Numerical Analysis*, 27(1):198–218, 1990.
19. H. Xiao and Z. Gimbutas. A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Computers and Mathematics with Applications*, 2009. in review.
20. T. Koornwinder. Two-variable analogues of the classical orthogonal polynomials. *In Theory and Applications of Special Functions*, Academic Press, Inc., pages 435–495, 1975.
21. R. Penrose. A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, 51:406–413, 1955.
22. P. G. Martinsson, V. Rokhlin, and M. Tygert. On interpolation and integration in finite-dimensional spaces of bounded functions. Technical Report 1317, Yale University, Department of Computer Science, March 2005.
23. A. H. Stroud. *Approximate Calculation of Multiple Integrals*. Prentice-Hall, 1971.
24. R. Cools and A. Haegemans. Why do so many cubature formulae have so many positive weights? *BIT Numerical Mathematics*, 28:792–802, 1988.
25. A. Sommariva and M. Vianello. Product Gauss cubature over polygons based on Green’s integration formula. *BIT Numerical Mathematics*, 47:441–453, 2007.
26. C. Talischi, G. H. Paulino, A. Pereira, and I. F. M. Menezes. Polygonal finite elements for topology optimization: A unifying paradigm. *International Journal for Numerical Methods in Engineering*, 2009. in press.
27. D. J. Holdych, D. R. Noble, and R. B. Secor. Quadrature rules for triangular and tetrahedral elements with generalized functions. *International Journal for Numerical Methods in Engineering*, 73:1310–1327, 2008.

Table I. Total number of nodes for quadratures up to degree  $d$  on polygons with five, six, seven and eight edges (no symmetry).  $S_5$ : pentagon,  $S_6$ : hexagon,  $S_7$ : heptagon and  $S_8$ : octagon.

$d$	Lower bound of $numx$ [23]	Obtained $numx$			
		$S_5$	$S_6$	$S_7$	$S_8$
1	1	1	1	1	1
2	3	3	3	3	3
3	3	4	4	4	4
4	6	6	6	6	6
5	6	7	7	7	7
6	10	11	11	11	11
7	10	13	12	13	12
8	15	16	16	16	17
9	15	19	19	19	19
10	21	23	23	23	24

Table II. Total number of nodes for quadratures up to degree  $d$  on the hexagon with different symmetric generators.  $S_6$ : no symmetry,  $S_6^1$ : rotational symmetry,  $S_6^2$ : reflective symmetry with symmetry line passing through mid-edges and  $S_6^3$ : reflective symmetry with symmetry line passing through vertices.

$d$	Lower bound	Obtained $numx$				Lyness and Monegato [17]
	of $numx$ [23]	$S_6$	$S_6^1$	$S_6^2$	$S_6^3$	$S_6^1$
1	1	1	1	1	1	1
2	3	3	6 <sup>a</sup>	3	3	6
3	3	4	6	4	4	6
4	6	6	7	6	6	7
5	6	7	7	7	7	7
6	10	11	13	11	11	13
7	10	12	13	12	12	13
8	15	16	19	17	17	19
9	15	19	19	18	18	19
10	21	23	25	23	23	25
15	36	NA <sup>b</sup>	48	NA	NA	48
18	55	NA	72	NA	NA	NA
20	66	NA	85	NA	NA	NA

<sup>a</sup> A quadrature rule of an even order  $d = 2k$  over the region  $S_6^1$  can be used for integration of all polynomials up to order  $d = 2k + 1$ , because when the rotational symmetric operator is applied to basis functions of odd degrees over the hexagon, they all vanish.

<sup>b</sup> Not attempted/available.

Table III: Quadrature rules for the hexagon (no symmetry).

	<b>x-coordinate</b>	<b>y-coordinate</b>	<b>weight</b>
order = 1, numx = 1, error = 5.97e-17			
1	0.0000000000000000	0.0000000000000000	2.5980762113533159
order = 2, numx = 3, error = 1.59e-16			
1	0.2686575045336940	-0.3589294778141325	1.3222874367783018
2	0.1779347781296624	0.7136132934950594	0.7224217676333652
3	-0.8742595671808975	-0.0739506280226420	0.5533670069416488
order = 3, numx = 4, error = 7.99e-17			
1	-0.4562707436715773	-0.4567445192965681	0.6493135361102734
2	0.4562707436715773	0.4567445192965681	0.6493135361102734
3	-0.4566000219291868	0.4561263961895862	0.6497245695663845
4	0.4566000219291868	-0.4561263961895862	0.6497245695663845
order = 4, numx = 6, error = 6.10e-17			
1	0.1717991689435392	0.6956751460412101	0.4417308013268524
2	-0.5190713023807637	-0.4865907901660747	0.4510230266576520
3	0.3397944106084456	-0.7780183709464641	0.2483537023160909
4	0.2109699927914922	-0.1105536634395474	0.7489043557284364
5	-0.6031425679675883	0.3543492206502724	0.4732558293048105
6	0.8572062700472260	0.0872308295970980	0.2348084960194733
order = 5, numx = 7, error = 2.40e-16			

Continued on next page

Table III – continued from previous page

	<b>x-coordinate</b>	<b>y-coordinate</b>	<b>weight</b>
1	0.0000000000000000	0.0000000000000000	0.6649837921916225
2	-0.7363417206023289	-0.1334199029395605	0.3221820698602822
3	0.2526258349850509	0.7044005873777404	0.3221820698602822
4	0.4837158856172780	-0.5709806844381799	0.3221820698602822
5	-0.4837158856172780	0.5709806844381799	0.3221820698602822
6	-0.2526258349850509	-0.7044005873777404	0.3221820698602822
7	0.7363417206023289	0.1334199029395605	0.3221820698602822

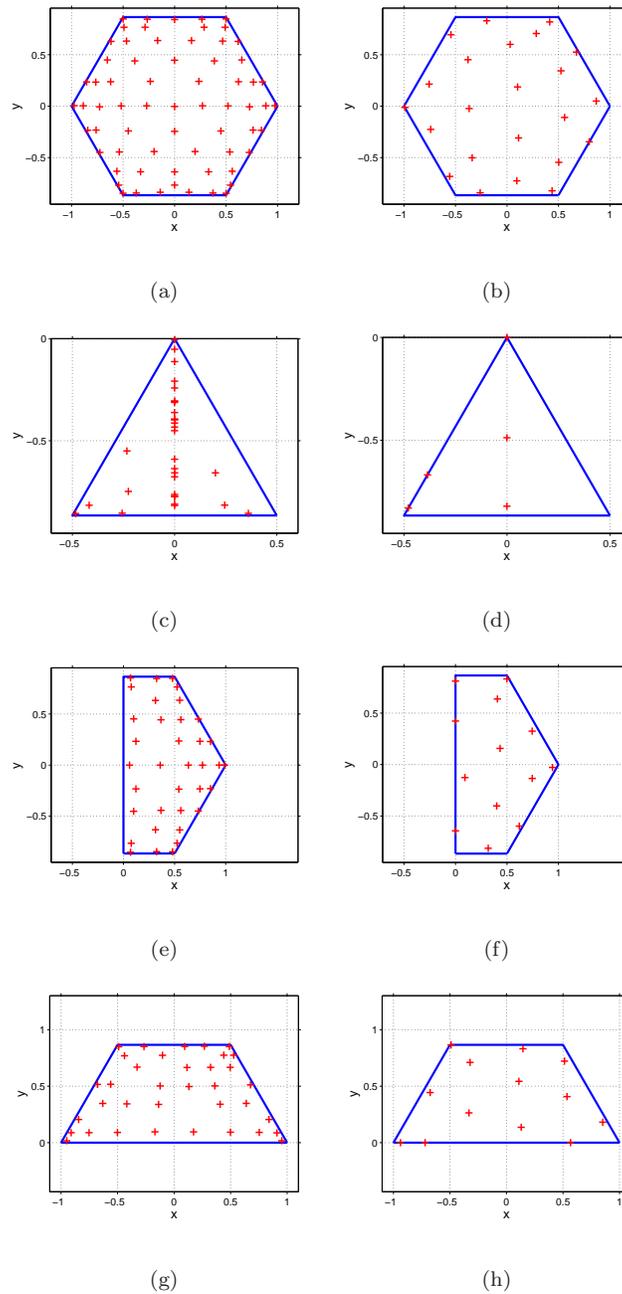


Figure 2. Quadrature rules of degree 10 over the hexagon. Left: Corrected tensor product; and right: final quadrature rule. (a) and (b) No symmetry; (c) and (d) Rotational symmetry; (e) and (f) Mid-edge reflection symmetry; and (g) and (h) Vertex-reflection symmetry.

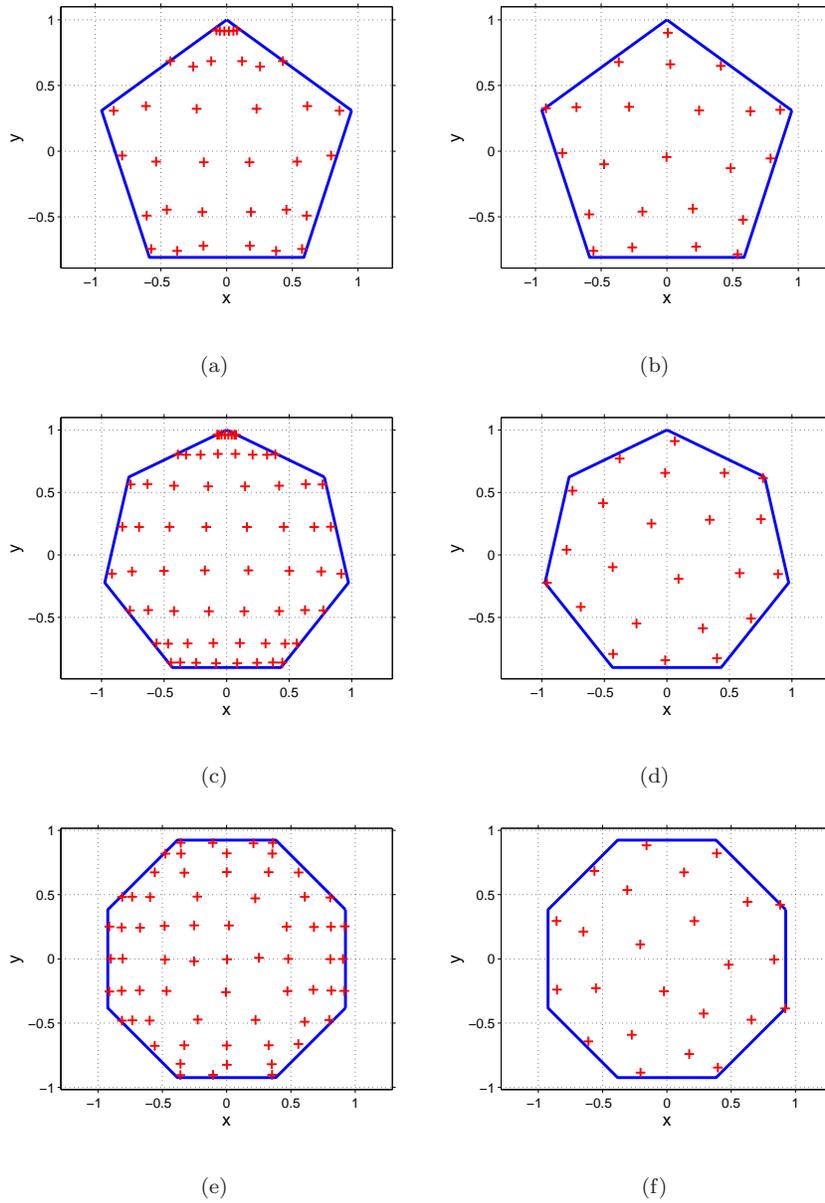


Figure 3. Quadrature rules of degree 10, no symmetry. Left: Corrected tensor product; and right: final quadrature rule. (a) and (b) Regular pentagon; (c) and (d) Regular heptagon; and (e) and (f) Regular octagon.

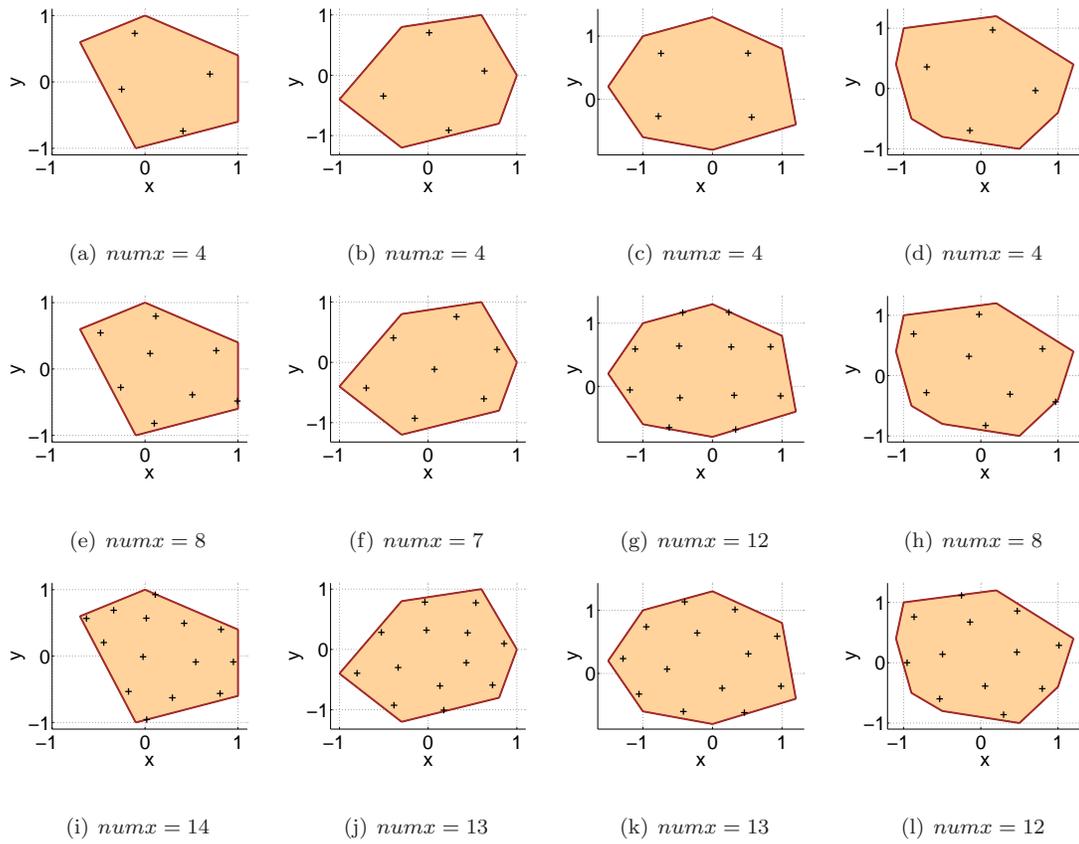


Figure 4. Quadrature rules for convex polygons with five to eight edges. (a-d) Degree three, (e-h) Degree five and (i-l) Degree seven.

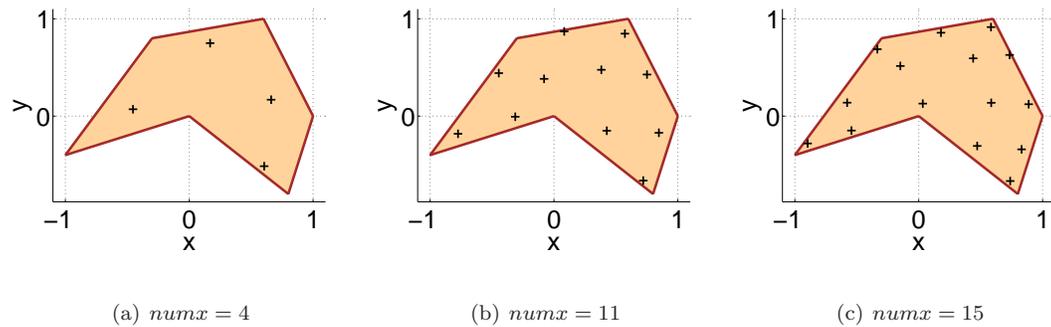


Figure 5. Quadrature rules for a concave hexagon. (a) Degree three, (b) Degree five and (c) Degree seven.

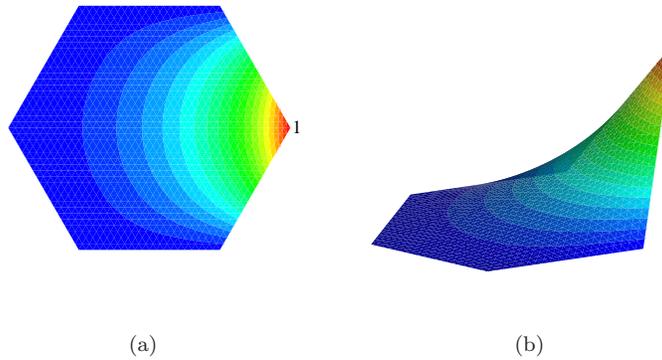


Figure 6. Laplace shape functions on a regular hexagon. (a)  $\phi_1$  (contour plot); and (b)  $\phi_1$  (3D plot).

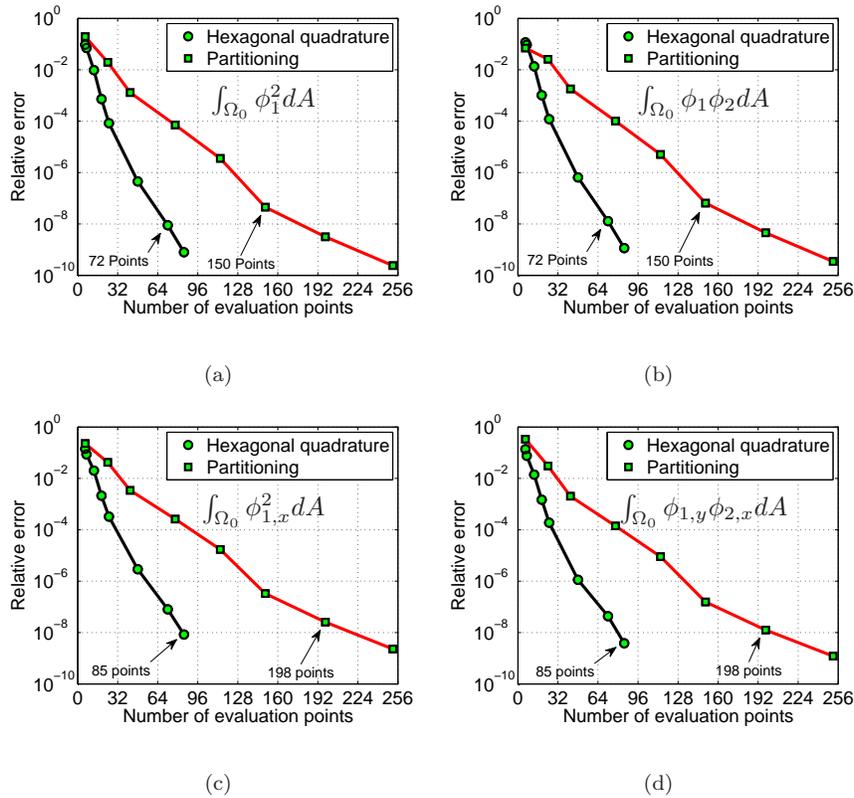


Figure 7. Relative error curves for integration using symmetric quadrature rules on hexagon and partitioning. (a) and (b) Mass matrix entries; and (c) and (d) Stiffness matrix entries.