

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

Bilinear Quadratures and Their Applications

Permalink

<https://escholarship.org/uc/item/0n2856vf>

Author

Wong, Christopher Allen

Publication Date

2016

Peer reviewed|Thesis/dissertation

Bilinear Quadratures and Their Applications

by

Christopher Allen Wong

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Applied Mathematics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor John Strain, Chair

Professor Maciej Zworski

Professor Sanjay Govindjee

Spring 2016

Bilinear Quadratures and Their Applications

Copyright 2016
by
Christopher Allen Wong

Abstract

Bilinear Quadratures and Their Applications

by

Christopher Allen Wong

Doctor of Philosophy in Applied Mathematics

University of California, Berkeley

Professor John Strain, Chair

A bilinear quadrature numerically evaluates a continuous bilinear map, such as the L^2 inner product, on continuous f and g belonging to known finite-dimensional function spaces. The evaluation of such maps is common in Galerkin methods for differential and integral equations. A framework for constructing bilinear quadratures over arbitrary domains in \mathbb{R}^d is developed. We provide rigorous error estimates under general conditions. We prove that in one dimension, these integration rules include Gaussian quadrature for polynomials and the trapezoidal rule for trigonometric polynomials as special cases. Lastly, we demonstrate that in higher dimensions, bilinear quadratures computed using this framework achieve similar or better performance in evaluating L^2 inner products than existing quadrature rules.

To my parents, Peter and Alexandra,
Who have always supported me in all my endeavors.

Contents

Contents	ii
List of Figures	iii
List of Tables	iv
1 Introduction	1
2 Preliminaries	4
2.1 Classical quadratures	4
2.2 Galerkin methods	8
3 Theory	10
3.1 Bilinear quadrature framework	10
3.2 Error estimates	14
3.3 Connection to classical quadratures	17
3.4 The non-invertible case	20
3.5 Change of variables	20
4 Computation	22
4.1 Construction	22
4.2 Approximate orthogonal projections	24
Bibliography	34
5 Appendix	37
5.1 Bilinear quadratures on the triangle	37
5.2 Bilinear quadratures on the square	46
5.3 Bilinear quadratures on the disk	54

List of Figures

4.1	Evaluation points for bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for L^2 on the interior of an equilateral triangle, for $n = 4, 8$	26
4.2	Evaluation points for bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for L^2 on the interior of a square, for $n = 4, 8$	28
4.3	Evaluation points for bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for L^2 on the unit disk, for $n = 4, 6$	31

List of Tables

4.1	Numerical results for k -point bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for L^2 on the interior of the reference right triangle.	25
4.2	Average ℓ^2 -norm relative error in computing approximate orthogonal projection coefficients onto \mathbb{P}_6 for four different sets of functions on the triangle.	26
4.3	Numerical results for k -point bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for L^2 on the interior of a square.	27
4.4	Average ℓ^2 -norm relative error in computing approximate orthogonal projection coefficients onto \mathbb{P}_5 for four different sets of functions on the square.	29
4.5	Average ℓ^2 -norm relative error in computing approximate orthogonal projection coefficients onto \mathbb{P}_8 for four different sets of functions on the square.	29
4.6	Numerical results for k -point bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for L^2 on the unit disk.	30
4.7	Average ℓ^2 -norm relative error in computing approximate orthogonal projection coefficients onto \mathbb{P}_5 for four different sets of functions on the unit disk.	31
4.8	Numerical results for bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for $H^1[-1,1]$ with the weight function $A(x) = 1 + x^2$	32
4.9	Numerical results for bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for $H^1[-1,1]$ with the weight function $A(x) = e^x$	32
5.1	Coordinates of the 3-point bilinear quadrature on the reference triangle	37
5.2	Coordinates of the 6-point bilinear quadrature on the reference triangle	37
5.3	Coordinates of the 10-point bilinear quadrature on the reference triangle	37
5.4	Coordinates of the 15-point bilinear quadrature on the reference triangle	38
5.5	Coordinates of the 21-point bilinear quadrature on the reference triangle	38
5.6	Coordinates of the 28-point bilinear quadrature on the reference triangle	39
5.7	Coordinates of the 36-point bilinear quadrature on the reference triangle	39
5.8	Coordinates of the 45-point bilinear quadrature on the reference triangle	40
5.9	Weight matrix for the 15-point bilinear quadrature on the reference triangle	41
5.10	Weight matrix for the 28-point bilinear quadrature on the reference triangle	42
5.11	Coordinates of the 3-point bilinear quadrature on $[-1, 1]^2$	46
5.12	Coordinates of the 6-point bilinear quadrature on $[-1, 1]^2$	46
5.13	Coordinates of the 10-point bilinear quadrature on $[-1, 1]^2$	46

5.14	Coordinates of the 15-point bilinear quadrature on $[-1, 1]^2$	47
5.15	Coordinates of the 21-point bilinear quadrature on $[-1, 1]^2$	47
5.16	Coordinates of the 28-point bilinear quadrature on $[-1, 1]^2$	48
5.17	Coordinates of the 36-point bilinear quadrature on $[-1, 1]^2$	48
5.18	Coordinates of the 45-point bilinear quadrature on $[-1, 1]^2$	49
5.19	Weight matrix for the 15-point bilinear quadrature on $[-1, 1]^2$	50
5.20	Weight matrix for the 21-point bilinear quadrature on $[-1, 1]^2$	51
5.21	Coordinates of the 3-point bilinear quadrature on the unit disk	54
5.22	Coordinates of the 6-point bilinear quadrature on the unit disk	54
5.23	Coordinates of the 10-point bilinear quadrature on the unit disk	54
5.24	Coordinates of the 15-point bilinear quadrature on the unit disk	54
5.25	Coordinates of the 21-point bilinear quadrature on the unit disk	55
5.26	Coordinates of the 28-point bilinear quadrature on the unit disk	55
5.27	Coordinates of the 36-point bilinear quadrature on the unit disk	56
5.28	Coordinates of the 45-point bilinear quadrature on the unit disk	57
5.29	Weight matrix for the 21-point bilinear quadrature on the unit disk	57

Acknowledgments

I would like to acknowledge all my fellow graduate students with and from whom I have learned, especially Christopher Melgaard. I am also thankful to all the mathematics faculty at UC Berkeley from whom I have learned so much.

I am especially grateful to my adviser, John Strain, for guidance and support. I also thank Ming Gu, Sanjay Govindjee, Per-Olof Persson, and Maciej Zworski for comments and suggestions. I also acknowledge the financial support of the Air Force Office of Scientific Research, the National Science Foundation, and the Simons Foundation.

Chapter 1

Introduction

Methods for the numerical evaluation of definite integrals are ubiquitous in computational science and engineering. In particular, classical quadratures such as Gaussian and trapezoidal rules accurately evaluate continuous linear functionals such as

$$\int_{\Omega} f(x)w(x) dx$$

for f in a finite-dimensional space of continuous functions over a region Ω and w a weight function. The construction of new methods for evaluating such integrals, especially over regions of Euclidean dimension greater than 1, remains an active area of research.

In this work, we introduce a computational framework for constructing *bilinear quadratures*, which evaluate continuous bilinear forms such as the weighted L^2 inner product

$$\langle f, g \rangle_{L^2} = \int_{\Omega} f(x)g(x)w(x) dx$$

or the weighted H^1 inner product

$$\langle f, g \rangle_{H^1} = \int_{\Omega} \sum_{i,j=1}^d \left(\frac{\partial f}{\partial x_i} a_{ij}(x) \frac{\partial g}{\partial x_j} \right) + f(x)g(x) dx$$

on finite-dimensional spaces of continuous functions f, g on a domain $\Omega \subset \mathbb{R}^d$. In constructing such quadratures, we aim to achieve both efficiency and accuracy.

Efficiency is achieved by using the fewest function evaluations possible. For Euclidean dimension $d > 1$, the optimal efficiency of a classical quadrature is unknown except in special cases. On the other hand, for a bilinear quadrature, determining the minimum number of function evaluations is more tractable because inner products have more algebraic structure than linear functionals. In particular, the minimum number of function evaluations is equal to the dimension of function space being integrated. The inner product of two functions f, g belonging to given finite-dimensional function spaces is computed by the formula

$$\langle f, g \rangle = f(\mathbf{x})^T W g(\mathbf{y}), \tag{1.1}$$

where $f(\mathbf{x}) \in \mathbb{R}^m$ and $g(\mathbf{y}) \in \mathbb{R}^n$ are vectors of f and g values at sets of points \mathbf{x} and \mathbf{y} in Ω respectively, and W is a matrix. Because the rank of the bilinear form is equal to the rank of W , the minimal number of required function evaluations is equal to the dimension of that function space. This result does not depend on the functions being integrated or the geometry of the integration region.

Accuracy is achieved by minimizing an appropriately-defined integration error. In a bilinear quadrature, this leads to a nonlinear optimization problem for \mathbf{x}, \mathbf{y} , and W in (1.1), and is solved using a Newton method for an appropriate objective function [2, 6, 26]. In this work an objective function is developed and demonstrated to yield numerically useful bilinear quadrature rules in a general setting.

Numerical evaluation of inner product integrals has been studied in [1, 2, 5, 15, 18] and as “bilinear quadrature” in [16, 17]. This work borrows some of the framework from these past works but develops and utilizes a different optimization procedure to produce quadrature rules.

The bilinear quadratures resulting from this optimization serve somewhat different purposes from classical quadratures. Classical quadratures are inherently linear and minimize error when integrating functions contained in or nearly in a prescribed function space. In contrast, bilinear quadratures only evaluate integrals related to bilinear forms and minimize error on a *pair* of function spaces. Because bilinear quadratures solve a much more specialized problem than classical quadratures, they are easier to construct.

One of the primary motivations for constructing bilinear quadratures as an alternative to well-established classical quadratures is for the evaluation of L^2 and H^1 inner products. L^2 inner products compute orthogonal projections onto subspaces, while H^1 inner products provide local solutions to elliptic problems, a key ingredient of the finite element method. For example, let $\Omega \subset \mathbb{R}^d$ be a smooth bounded domain, let L be a uniformly elliptic operator, $f \in L^2(\Omega)$, $g \in L^2(\partial\Omega)$, and $\gamma \in L^\infty(\partial\Omega)$. Consider the Robin problem

$$\begin{cases} \text{Find } u \in H^1(\Omega) \text{ satisfying} \\ Lu = f \text{ in } \Omega, \gamma u + \frac{\partial u}{\partial n} = g \text{ on } \partial\Omega. \end{cases} \quad (1.2)$$

When $L = -\Delta$, then if a bilinear form $a : H^1(\Omega) \times H^1(\Omega) \rightarrow \mathbb{R}$ is defined by $a(u, v) = \int_{\Omega} Du \cdot Dv + \int_{\partial\Omega} \gamma uv$, the weak formulation to (1.2) seeks $u \in H^1(\Omega)$ satisfying

$$a(u, v) = \langle f, v \rangle_{L^2(\Omega)} + \langle g, v \rangle_{L^2(\partial\Omega)} \text{ for all } v \in H^1(\Omega). \quad (1.3)$$

Given user-selected finite-dimensional function spaces $\mathcal{F}_0, \mathcal{G}_0$, the Galerkin method constructs an approximate solution to (1.3) by seeking $u_0 \in \mathcal{F}_0$ satisfying

$$a(u_0, v_0) = \langle f, v_0 \rangle_{L^2(\Omega)} + \langle g, v_0 \rangle_{L^2(\partial\Omega)} \text{ for all } v_0 \in \mathcal{G}_0. \quad (1.4)$$

The linear system (1.4) is solved in a basis, which requires computing a number of L^2 inner product integrals.

Because of the general nature of the methods outlined in this work, these methods may prove particularly useful for variants and generalizations of the Galerkin methods other than the standard finite element method. For example, bilinear quadratures can be used to compute local solutions to PDE over high-order or non-simplicial meshes, regions on which classical quadratures may not yet be available. Furthermore, they can be used to compute matrix elements in generalized finite element or Petrov-Galerkin methods, where the basis functions may be non-polynomial or differ from the test functions.

Chapter 2

Preliminaries

2.1 Classical quadratures

In this section we review traditional methods developed for numerically evaluating definite integrals. We collectively call all these methods *classical quadratures*, though in some texts the term “quadrature” is reserved only for one-dimensional integrals. All the results in this section can be found in standard texts [8], though some of the proofs contained below are original.

Definition 1. *Let $\Omega \subset \mathbb{R}^d$ be a connected domain. A classical quadrature q of order n on Ω is a linear functional on $C(\Omega)$ defined by a set $\mathbf{x} = (x_1, \dots, x_n)$, $x_i \in \Omega$, called the nodes, and a vector $\mathbf{w} \in \mathbb{R}^n$, whose components are called the weights, such that for any $f \in C(\Omega)$,*

$$q(f) = \mathbf{w}^T f(\mathbf{x}) = \sum_{i=1}^n w_i f(x_i).$$

Furthermore, given a \mathcal{F}_0 is a subspace of $C(\Omega)$ and Borel measure μ , then q is said to be exact on \mathcal{F}_0 with respect to μ if

$$q(f) = \int_{\Omega} f d\mu$$

for all $f \in \mathcal{F}_0$.

Classical quadratures have a simple measure-theoretic interpretation. Since the Riesz-Markov theorem states that bounded linear functionals on $C(\Omega)$ are the same as signed Borel measures on Ω , then we can view a classical quadrature q that is exact on \mathcal{F}_0 as a linear combination of Dirac measures such that the restrictions of q and μ to \mathcal{F}_0 agree: $q|_{\mathcal{F}_0} = \mu|_{\mathcal{F}_0}$. Restricting only to positive measures (so that the weights $\mathbf{w} \geq 0$), the search for exact classical quadratures can be viewed as a linear program over the infinite-dimensional cone of positive measures on Ω [20]. In light of this, it is understandable that the search for classical quadratures with optimal efficiency can be challenging.

Gaussian quadratures

In the construction of classical quadratures, one of the most important cases is $\mathcal{F}_0 = \mathbb{P}_n = \mathbb{P}_n(\Omega)$, the space of polynomials of degree up to n defined on a region Ω . When Ω is an open interval and μ is a finite absolutely continuous Borel measure on Ω , the classical *Gaussian quadratures* are known to provide exact quadratures on \mathbb{P}_n using the fewest evaluation points.

Definition 2. *Suppose \mathbb{P}_{2n-1} is μ -integrable on an interval Ω . Then a Gaussian quadrature of order n on Ω is a classical quadrature of order n on Ω that is exact on \mathbb{P}_{2n-1} with respect to μ .*

Gaussian quadratures are the starting point for much of the existing theory for classical quadratures in one dimension. However, the one-dimensional theory does not generalize to higher dimensions, which is needed in applications such as the numerical solutions of PDE. To understand why these limitations exist, we develop the existence and uniqueness theory for Gaussian quadratures.

Theorem 1. *Suppose \mathbb{P}_{2n-1} is μ -integrable on an interval Ω , and let $\{\phi_k\}$ denote any set of $L^2(\Omega, \mu)$ -orthonormal polynomials such that $\deg(\phi_k) = k$. Then the following sets are equal:*

1. *The zeros of ϕ_n .*
2. *The eigenvalues of the symmetric bilinear form on \mathbb{P}_{n-1} given by*

$$B(f, g) := \int_{\Omega} xf(x)g(x) d\mu = \langle xf(x), g(x) \rangle_{L^2(\Omega, \mu)}.$$

3. *The nodes $\{x_i\}$ of a Gaussian quadrature of order n on Ω .*

Proof. (1 \iff 2) Since $B(\cdot, \cdot)$ is symmetric it is diagonalizable with n real eigenvalues $\{\lambda_i\}$. If a polynomial $\psi_i(x)$ is an eigenvector for λ_i , then for $0 \leq j \leq n-1$,

$$\langle x\psi_i(x), \phi_j(x) \rangle = \lambda_i \langle \psi_i(x), \phi_j(x) \rangle \implies \langle (x - \lambda_i)\psi_i(x), \phi_j(x) \rangle = 0.$$

Since $(x - \lambda_i)\psi_i(x) \in \mathbb{P}_n$ for each i , and the only polynomials in \mathbb{P}_n that are orthogonal to each of $\phi_0, \dots, \phi_{n-1}$ are multiples of ϕ_n , then each $(x - \lambda_i)$ is a factor of $\phi_n(x)$. Thus $\phi_n(x)$ is a multiple of $(x - \lambda_1) \dots (x - \lambda_n)$ and its zeros are the eigenvalues of $B(\cdot, \cdot)$.

(2 \iff 3) Suppose a Gaussian quadrature with positive weights $\{w_i\}$ and nodes $\{x_i\}$ exists. With respect to the basis of orthonormal polynomials $\{\phi_k\}$, the bilinear form $B(\cdot, \cdot)$ has a symmetric matrix representation B with entries given by

$$B_{ij} = \int_{\Omega} x\phi_i(x)\phi_j(x) d\mu = \sum_{k=1}^n w_k x_k \phi_i(x_k)\phi_j(x_k).$$

If

$$u_k = \begin{bmatrix} \sqrt{w_k} \phi_0(x_k) \\ \vdots \\ \sqrt{w_k} \phi_{n-1}(x_k) \end{bmatrix}, X = \begin{bmatrix} x_1 & & \\ & \ddots & \\ & & x_n \end{bmatrix},$$

then

$$B = \sum_{k=1}^n x_k u_k u_k^T = U X U^T. \quad (2.1)$$

Since $\delta_{ij} = \sum_{k=1}^n w_k \phi_i(x_k) \phi_j(x_k)$, then $I = U U^T$ and U is a unitary matrix. Then (2.1) is the unitary diagonalization of the symmetric matrix B with eigenvalues given by the x_k 's. \square

Theorem 2. *For every $n > 0$ and every finite absolutely continuous Borel measure μ on an interval Ω , a Gaussian quadrature of order n exists and is unique.*

Proof. By Theorem 1, a Gaussian quadrature q of order n only exists if its nodes $\{x_i\}$ are the roots of ϕ_n , the n -th degree orthonormal polynomial with respect to $L^2(\Omega, \mu)$. Suppose such a q exists with nodes $\{x_i\}$ and weights $\{w_i\}$. Let $f \in \mathbb{P}_{2n-1}$. Then factorize $f(x) = p(x)\phi_n(x) + r(x)$, where $\deg p, \deg r < n$. Then

$$q(f) = \sum_i w_i (p(x_i)\phi_n(x_i) + r(x_i)) = \sum_i w_i r(x_i) = \int_{\Omega} r d\mu,$$

which implies that the existence of q follows from the exactness of q as a classical quadrature on \mathbb{P}_{n-1} . Setting $m_i = \int_{\Omega} x^{i-1} d\mu$ and $\mathbf{m} = (m_i)$, then a vector of weights $\mathbf{w} = (w_i)$ satisfies $V\mathbf{w} = \mathbf{m}$, where V is the Vandermonde matrix evaluated at (x_1, \dots, x_n) . Therefore the weights exist and are unique by the invertibility of V , and hence the Gaussian quadrature q does indeed exist.

Furthermore, to show that the weights are positive, so that the square root taken in (2.1) in the proof of Theorem 1 is legitimate, consider a monic polynomial f with a double root at every x_i except $i = k$. Then f has degree $2n - 2$ and is nonnegative, and hence $q(f) = w_k f(x_k) > 0$, which implies $w_k > 0$, and this holds for each $k = 1, \dots, n$. \square

Remarks. In fact, the weights w_i are given by the equation

$$1/w_i = \sum_{j=0}^n (\phi_j(x_i))^2.$$

See [8] for details. Furthermore, Theorem 1 also provides an efficient method to construct these quadratures. The matrix B in (2.1) is tridiagonal, so its eigenvalues can be calculated quickly, even for very large n [14].

Gaussian quadrature is optimal for integrating polynomials on an interval, but it is clear that Theorem 1 does not extend readily to higher-dimensional domains. The zeros of a

multivariate polynomial are generally not isolated (consider $f(x, y) = xy$) so they cannot all be used as nodes of a classical quadrature. Additionally, the connection between nodes and eigenvalues no longer holds since the eigenvalues are only scalars (the connection extends to two-dimensional domains with complex eigenvalues [25]).

Trapezoidal rule

The trapezoidal rule is one of the simplest numerical integration methods. It can be viewed as a classical n -point quadrature that is exact on the $(2n-1)$ -dimensional space of trigonometric polynomials

$$T_{n-1} = \text{span}\{1, \sin x, \dots, \sin(n-1)x, \cos x, \dots, \cos(n-1)x\}.$$

Because of its simplicity and its accuracy on periodic functions, it has enjoyed widespread use.

Definition 3. On $\Omega = (0, 2\pi)$, the $(n+1)$ -point trapezoidal rule \mathcal{T}_n is a classical quadrature of order $n+1$ given by

$$\mathcal{T}_n(p) := \frac{\pi}{n}p(0) + \frac{\pi}{n}p(2\pi) + \frac{2\pi}{n} \sum_{j=1}^{n-1} p\left(\frac{2\pi j}{n}\right)$$

that is exact on T_{n-1} .

Alternatively, by identifying endpoints and setting $\Omega = S^1$, \mathcal{T}_n is a classical quadrature of order n exact on T_{n-1} .

Concretely, since T_{n-1} is a rotationally-invariant function space on S^1 , the trapezoidal rule yields a family of n -point classical quadratures for T_{n-1} given by

$$\mathcal{T}_n(p) = \int_0^{2\pi} p(x) dx = \frac{2\pi}{n} \sum_{j=0}^{n-1} p(x_j) \quad \text{for all } p \in T_{n-1}, \quad \text{whenever } x_{j+1} - x_j = \frac{2\pi}{n}. \quad (2.2)$$

Lobatto quadratures

In some cases it may be advantageous to use a classical quadrature that does not use the optimal number of function evaluations. Lobatto quadratures are one such example.

Definition 4. Let \mathbb{P}_{2n-1} be μ -integrable on an interval $\Omega = [a, b]$. Then the corresponding Lobatto quadrature is a classical quadrature of order $n+1$ exact on \mathbb{P}_{2n-1} with respect to μ such that if x_0, \dots, x_n are the nodes, then $x_0 = a$ and $x_n = b$.

Notice that Lobatto quadratures use one more node than Gaussian quadratures. However, in exchange, two of the nodes are the endpoints of the integration interval.

Theorem 3. Suppose \mathbb{P}_{2n-1} is μ -integrable on Ω , and let $\{\phi_k\}$ denote the unique set of polynomials such that $\deg(\phi_k) = k$ and which are orthonormal on $L^2(\Omega)$ with respect to the measure $(x-a)(x-b)d\mu$. Then there exists a unique Lobatto quadrature of order $n+1$, and its nodes are the zeros of $(x-a)(x-b)\phi_{n-1}$.

Proof. Suppose a Lobatto quadrature q of order $n+1$ exists. Then, for $f \in \mathbb{P}_{2n-1}$, write $f(x) = p(x)\phi_{n-1}(x)(x-a)(x-b) + r(x)$, where $\deg p, \deg r < n-1$. Then, by the orthogonality of ϕ_{n-1} with respect to the measure $(x-a)(x-b)d\mu$, $\int_{\Omega} f d\mu = \int_{\Omega} r d\mu$. Therefore, by the exactness condition for q implies that

$$0 = \int_{\Omega} p(x)\phi_{n-1}(x)(x-a)(x-b) d\mu = q(p\phi_{n-1}(x-a)(x-b)).$$

Since this must hold for every polynomial p of degree less than $n-1$, then the function $\phi_{n-1}(x)(x-a)(x-b)$ must vanish on every node of q .

With the nodes determined, the weights can be determined that make the classical quadrature exact by forming a linear system, similar to in the proof of Theorem 2. To show existence and uniqueness it suffices to write down an explicit formula for the solution to this matrix linear system. We omit that calculation here. \square

2.2 Galerkin methods

In this section we briefly describe the general principles of Galerkin methods for solving partial differential equations, highlighting the importance of using quadratures in implementing these methods numerically.

The underlying abstract problem is, given a continuous bilinear form a on a pair of Banach spaces U, V , and f a linear functional on V ,

$$\begin{cases} \text{Find } u \in U \text{ such that} \\ a(u, v) = f(v) \forall v \in V. \end{cases} \quad (2.3)$$

Note that traditionally, the symmetric case $U = V$ is often referred to simply as the *Galerkin method*, while the general non-symmetric case is often referred to as the *Petrov-Galerkin method*.

The process of taking a PDE and converting into the form (2.3) appropriate for solving is an area of study in itself. However, for certain problems, such as elliptic PDE and hyperbolic conservation laws, this process is well-studied. If we consider the Poisson equation $-\Delta u = f$ with homogeneous Dirichlet boundary condition on Ω , then it can be written in a *weak formulation* of the form (2.3) by setting $U = V = H_0^1(\Omega)$ and $a(u, v) = \int_{\Omega} Du \cdot Dv$.

Similarly, for a scalar conservation law of the form

$$\frac{\partial u}{\partial t} + \operatorname{div}(f(u)) = g, \quad x \in \Omega,$$

to be solved for $u(x, t)$ with appropriate initial and boundary conditions, a weak form is given by

$$\int_{\Omega} \frac{\partial u}{\partial t} v - f(u) \cdot Dv \, dx + \int_{\partial\Omega} \mathbf{n} \cdot f(u) v \, ds = \int_{\Omega} gv \, dx$$

for all v in some appropriate function space.

In each case, to numerically compute the solution u , u is approximated by a function contained in a finite-dimensional function space whose integrals can be computed, and similarly for the test functions v . The result is a discretized matrix system, which can be solved using standard numerical methods. It is imperative then to find methods for numerically computing the integrals involved in forming these matrices. Bilinear quadratures are a natural choice.

Finite element methods

Some of the most widely used types of Galerkin methods are also *finite element methods*. In such methods, a mesh is constructed that partitions the domain Ω into subdomains Ω_i . Then, (2.3) is appropriately modified and restricted to each subdomain Ω_i , on which the solution u is approximated using an approximation space U_i of low dimension. For example, in the traditional *continuous Galerkin* finite element method, $U = V$ is taken to be a space of continuous piecewise polynomials up to a fixed degree. Then (2.3) is solved locally on the subdomain Ω_i :

$$\begin{cases} \text{Find } u_i \in \mathbb{P}_n(\Omega_i) \text{ such that} \\ a(u_i, v_i) = f(v_i) \forall v_i \in \mathbb{P}_n(\Omega_i), \end{cases} \quad (2.4)$$

under the additional continuity constraint that the boundary values of each local solution match the values of solutions on neighboring subdomains. By making the mesh $\{\Omega_i\}$ sufficiently fine, u_i can be well-approximated by polynomials of relatively low degree n , sometimes $n = 1$. Thus the matrix discretization of $a(\cdot, \cdot)$ is very sparse and the resulting linear system is more easily solved.

Another variant of such methods is the *discontinuous Galerkin* finite element method. Discontinuous Galerkin is often more suitable for problems where boundary information propagates in particular directions, such as in equations modeling fluid flow. In this variant, each local solution on U_i is computed without any continuity condition at the boundary. A *numerical flux function*, determined using a priori knowledge about the underlying equation, is used to minimize the difference at the interface between two subdomains.

In all cases, the most common choice of basis functions is polynomials on simplices. Thus, the integration of polynomials over simplices presents an important test case for any candidate quadratures.

Chapter 3

Theory

3.1 Bilinear quadrature framework

Abstract formulation

In this section the problem of evaluating a general continuous bilinear form on a pair of Banach spaces is considered. Results are given in sufficient generality so that they apply to any continuous bilinear forms. Later, these results are applied to useful special cases such as the L^2 and H^1 inner products.

Definition 5. Let \mathcal{F} and \mathcal{G} be real Banach spaces. Then a bilinear quadrature of order (m, n) on $\mathcal{F} \times \mathcal{G}$ is a bilinear form Q defined by linear maps $L_1 : \mathcal{F} \rightarrow \mathbb{R}^m$ and $L_2 : \mathcal{G} \rightarrow \mathbb{R}^n$ and a bilinear map $B : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}$, such that, for each $f \in \mathcal{F}$ and $g \in \mathcal{G}$,

$$Q(f, g) = B(L_1 f, L_2 g).$$

Definition 6. Let \mathcal{F}, \mathcal{G} be real Banach spaces with a continuous bilinear form $\langle \cdot, \cdot \rangle : \mathcal{F} \times \mathcal{G} \rightarrow \mathbb{R}$. Finite-dimensional subspaces $\mathcal{F}_0 \subset \mathcal{F}$ and $\mathcal{G}_0 \subset \mathcal{G}$ are a dual pair if they separate points:

$$\begin{aligned} \forall f \in \mathcal{F}_0 \setminus \{0\}, \exists g \in \mathcal{G}_0 \text{ such that } \langle f, g \rangle \neq 0, \\ \forall g \in \mathcal{G}_0 \setminus \{0\}, \exists f \in \mathcal{F}_0 \text{ such that } \langle f, g \rangle \neq 0. \end{aligned}$$

If $\mathcal{F}_0, \mathcal{G}_0$ are a dual pair then $\dim(\mathcal{F}_0) = \dim(\mathcal{G}_0)$.

Definition 7. Let \mathcal{F}, \mathcal{G} be real Banach spaces with a continuous bilinear form $\langle \cdot, \cdot \rangle : \mathcal{F} \times \mathcal{G} \rightarrow \mathbb{R}$, and let $\mathcal{F}_0 \subset \mathcal{F}$ and $\mathcal{G}_0 \subset \mathcal{G}$ be a dual pair. A bilinear quadrature Q on $\mathcal{F} \times \mathcal{G}$ is exact with respect to $\mathcal{F}_0 \times \mathcal{G}_0$ if

$$\langle f, g \rangle = Q(f, g) \text{ for every } f \in \mathcal{F}_0, g \in \mathcal{G}_0.$$

Such a bilinear quadrature evaluates the bilinear form on $\mathcal{F}_0 \times \mathcal{G}_0$ exactly. If the parent spaces \mathcal{F}, \mathcal{G} are implied, we will abuse notation by referring to an exact bilinear quadrature on $\mathcal{F}_0 \times \mathcal{G}_0$.

Remark. If \mathcal{F}, \mathcal{G} are infinite-dimensional and Q is exact on $\mathcal{F}_0 \times \mathcal{G}_0$, then

$$\sup_{f \in \mathcal{F}, g \in \mathcal{G}} |Q(f, g) - \langle f, g \rangle| = \infty,$$

so a bilinear quadrature can only be accurate on finite-dimensional subspaces.

Lemma 1. *Let $\mathcal{F}_0 \subset \mathcal{F}$ and $\mathcal{G}_0 \subset \mathcal{G}$ be a dual pair, and let $L_1 : \mathcal{F} \rightarrow \mathbb{R}^m$, $L_2 : \mathcal{G} \rightarrow \mathbb{R}^n$ be linear. Then there exists bilinear $B : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}$ such that the map $Q(f, g) = B(L_1 f, L_2 g)$ is an exact quadrature on $\mathcal{F}_0 \times \mathcal{G}_0$ if and only if $L_1|_{\mathcal{F}_0}$ and $L_2|_{\mathcal{G}_0}$ are both injective.*

Proof. Suppose B exists. If $f, \tilde{f} \in \mathcal{F}_0$ are distinct then there exists $g \in \mathcal{G}_0$ such that

$$0 \neq \langle f - \tilde{f}, g \rangle = Q(f - \tilde{f}, g) = B(L_1(f - \tilde{f}), L_2 g),$$

so $L_1 f \neq L_1 \tilde{f}$ and $L_1|_{\mathcal{F}_0}$ is injective. Similarly for $L_2|_{\mathcal{G}_0}$.

Suppose $L_1|_{\mathcal{F}_0}$ and $L_2|_{\mathcal{G}_0}$ are injective. Their Moore-Penrose pseudoinverses $(L_1|_{\mathcal{F}_0})^+$ and $(L_2|_{\mathcal{G}_0})^+$ left-invert L_1 and L_2 , respectively. Define a bilinear map on $\mathbb{R}^m \times \mathbb{R}^n$ by

$$B(x, y) = \langle (L_1|_{\mathcal{F}_0})^+ x, (L_2|_{\mathcal{G}_0})^+ y \rangle.$$

Then, for all $f \in \mathcal{F}_0, g \in \mathcal{G}_0$,

$$\begin{aligned} B(L_1 f, L_2 g) &= \langle (L_1|_{\mathcal{F}_0})^+ L_1|_{\mathcal{F}_0} f, (L_2|_{\mathcal{G}_0})^+ L_2|_{\mathcal{G}_0} g \rangle \\ &= \langle f, g \rangle. \end{aligned}$$

□

From Lemma 1 a necessary condition for an exact bilinear quadrature is that $m, n \geq \dim(\mathcal{F}_0) = \dim(\mathcal{G}_0)$. Minimal order is achieved when $m = n = \dim(\mathcal{F}_0)$ and $B(x, y)$ is uniquely given by

$$B(x, y) = \langle (L_1|_{\mathcal{F}_0})^{-1} x, (L_2|_{\mathcal{G}_0})^{-1} y \rangle.$$

Exact bilinear quadratures are not unique, as there are many possible linear maps L_1, L_2 . Furthermore, B may not be unique, since if $n > \dim(\mathcal{F}_0)$, then $L_1|_{\mathcal{F}_0}$ has infinitely many left inverses. Therefore, a method is needed to choose among the infinitely many bilinear quadratures. One metric of quality is that, in addition to its exactness on $\mathcal{F}_0 \times \mathcal{G}_0$, the bilinear quadrature also approximates $\langle f, g \rangle$ for some set of g 's outside of \mathcal{G}_0 .

Definition 8. *Let $\mathcal{F}_0 \subset \mathcal{F}$ and $\mathcal{G}_0 \subset \mathcal{G}$ be a dual pair, and let $\mathcal{G}_1 \subset \mathcal{G}$ be another finite-dimensional subspace such that*

$$\mathcal{G}_1 \subset \mathcal{F}_0^\perp := \{g \in \mathcal{G} : \langle f, g \rangle = 0 \text{ for all } f \in \mathcal{F}_0\}.$$

Let \mathcal{Q} be a set of bilinear quadratures exact on $\mathcal{F}_0 \times \mathcal{G}_0$. Then $Q \in \mathcal{Q}$ is called minimal on \mathcal{G}_1 if

$$Q = \arg \min_{\tilde{Q} \in \mathcal{Q}} \sigma(\tilde{Q}; \mathcal{F}_0, \mathcal{G}_1), \tag{3.1}$$

where

$$\sigma(\tilde{Q}; \mathcal{F}_0, \mathcal{G}_1) := \max_{\substack{0 \neq g \in \mathcal{G}_1 \\ 0 \neq f \in \mathcal{F}_0}} \frac{|\tilde{Q}(f, g)|}{\|f\|_{\mathcal{F}} \|g\|_{\mathcal{G}}}.$$

If Q is minimal on \mathcal{G}_1 , then it approximates the pairing of \mathcal{F}_0 and $\mathcal{G}_0 \oplus \mathcal{G}_1$. Precisely, if $f \in \mathcal{F}_0, g \in \mathcal{G}_0 \oplus \mathcal{G}_1$, and we write $g = g_0 + g_1$ with $g_i \in \mathcal{G}_i$, then

$$|Q(f, g) - \langle f, g \rangle| = |Q(f, g_1)| \leq \sigma(Q; \mathcal{F}_0, \mathcal{G}_1) \|f\|_{\mathcal{F}} \|g_1\|_{\mathcal{G}}. \quad (3.2)$$

Thus, minimizing $\sigma(Q; \mathcal{F}_0, \mathcal{G}_1)$ will improve the approximation.

One important special case for bilinear quadratures is the symmetric case, which is when $F = G$ is an inner product space. In this case, a bilinear quadrature computes an orthogonal projection.

Definition 9. Let \mathcal{F}_0 and \mathcal{G}_0 be a dual pair in an inner product space. Let $\{f_i\}$ be an orthonormal basis for \mathcal{F}_0 . Given a bilinear quadrature Q exact on $\mathcal{F}_0 \times \mathcal{G}_0$, the approximate orthogonal projection onto \mathcal{F}_0 arising from Q is the linear map P_Q given by

$$P_Q(g) = \sum_i Q(f_i, g) f_i.$$

A more general version of (3.2) involving orthogonal projections is stated and proven later in Theorem 4.

Integral formulation

In this section, the bilinear quadrature framework is applied to the evaluation of Sobolev inner products on function spaces. Let $\Omega \subset \mathbb{R}^d$ be a bounded domain, let $\mathcal{F} = \mathcal{G} = C^r(\bar{\Omega})$, r a non-negative integer, and consider the H^s Sobolev inner product

$$\langle f, g \rangle_{H^s} = \sum_{|\alpha| \leq s} \langle D^\alpha f, D^\alpha g \rangle_{L^2(\Omega)}$$

for a fixed $s \leq r$. Notice that the H^s inner product is continuous with respect to the topology of $C^r(\bar{\Omega})$.

Choose a dual pair $\mathcal{F}_0, \mathcal{G}_0$ in \mathcal{F} . Exactness on $\mathcal{F}_0 \times \mathcal{G}_0$ requires linear maps $L_1 : C^r(\bar{\Omega}) \rightarrow \mathbb{R}^m, L_2 : C^r(\bar{\Omega}) \rightarrow \mathbb{R}^n$, and bilinear form $B : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}$ so that for every $f \in \mathcal{F}_0, g \in \mathcal{G}_0$, $B(L_1 f, L_2 g) = \langle f, g \rangle_{H^s}$.

Appropriate linear maps L_1, L_2 are evaluations at particular points in Ω . Thus, for the points $\mathbf{x} = (x_1, \dots, x_m)^T \in \Omega^m, \mathbf{y} = (y_1, \dots, y_n)^T \in \Omega^n$, define

$$L_1 f := f(\mathbf{x}) = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix}, \quad L_2 g := g(\mathbf{y}) = \begin{bmatrix} g(y_1) \\ \vdots \\ g(y_n) \end{bmatrix}.$$

Given orthonormal basis $\beta = \{f_1, \dots, f_k\}$ for \mathcal{F}_0 and basis $\{g_1, \dots, g_k\}$ for \mathcal{G}_0 , let $M \in \mathbb{R}^{k \times k}$ be the Gram matrix with entries

$$M_{ij} = \langle f_i, g_j \rangle_{H^s}.$$

Since $\mathcal{F}_0, \mathcal{G}_0$ are a dual pair, M is invertible. Define matrix functions

$$F(\mathbf{x}) := \begin{bmatrix} f_1(x_1) & \dots & f_k(x_1) \\ \vdots & & \vdots \\ f_1(x_m) & \dots & f_k(x_m) \end{bmatrix}, \quad G(\mathbf{y}) := \begin{bmatrix} g_1(y_1) & \dots & g_k(y_1) \\ \vdots & & \vdots \\ g_1(y_n) & \dots & g_k(y_n) \end{bmatrix}.$$

To make L_1 and L_2 injective, choose \mathbf{x} and \mathbf{y} , such that $F(\mathbf{x})$ and $G(\mathbf{y})$ have full column rank. If $B(v, w) = v^T W w$ for all v, w for an $m \times n$ matrix W , then the bilinear quadrature is exact if and only if

$$F(\mathbf{x})^T W G(\mathbf{y}) = M. \quad (3.3)$$

Therefore a bilinear quadrature rule

$$Q(f, g) = f(\mathbf{x})^T W g(\mathbf{y}) \quad (3.4)$$

evaluates $\langle f, g \rangle_{H^s}$ exactly for any $f \in \mathcal{F}_0, g \in \mathcal{G}_0$. The corresponding approximate orthogonal projection onto \mathcal{F}_0 is

$$P_Q(g) = \sum_{i=1}^k [f_i(\mathbf{x})^T W g(\mathbf{y})] f_i.$$

In the basis β , the approximate projection is computed by

$$[P_Q(g)]_\beta = F(\mathbf{x})^T W g(\mathbf{y}) \in \mathbb{R}^k. \quad (3.5)$$

Good values for the matrix W and evaluation points \mathbf{x}, \mathbf{y} must be determined. Without loss of generality, suppose that the bases $\{f_i\}$ and $\{g_j\}$ are H^s -orthonormal in $C^r(\bar{\Omega})$. Select finite-dimensional $\mathcal{G}_1 \subset C^r(\bar{\Omega})$ for the minimization (3.1) and define the feasible set \mathcal{Q} to be all quadratures of the form (3.4) satisfying (3.3). If $\{\gamma_1, \dots, \gamma_p\}$ is an orthonormal basis for \mathcal{G}_1 , define

$$\Gamma(\mathbf{x}) := \begin{bmatrix} \gamma_1(x_1) & \dots & \gamma_p(x_1) \\ \vdots & & \vdots \\ \gamma_1(x_n) & \dots & \gamma_p(x_n) \end{bmatrix} \in \mathbb{R}^{n \times p}.$$

By identifying a bilinear quadrature Q with its associated points \mathbf{x}, \mathbf{y} and bilinear form matrix W , (3.1) can be reformulated as

$$\begin{aligned} \arg \min_{Q \in \mathcal{Q}} \sigma(Q; \mathcal{F}_0, \mathcal{G}_1) &= \arg \min_{Q \in \mathcal{Q}} \max_{\substack{g \in \mathcal{G}_1, \|g\|_G=1 \\ f \in \mathcal{F}_0, \|f\|_F=1}} |Q(f, g)| \\ &= \arg \min_{\mathbf{x}, \mathbf{y}, W} \max_{\substack{a, b \in \mathbb{R}^k \\ \|a\|_2 = \|b\|_2 = 1}} |b^T F(\mathbf{x})^T W \Gamma(\mathbf{y}) a|, \end{aligned} \quad (3.6)$$

subject to $F(\mathbf{x})^T W G(\mathbf{y}) = M$. For fixed a , \mathbf{x} , \mathbf{y} , W , the expression in (3.6) is maximized when $b = F(\mathbf{x})^T W \Gamma(\mathbf{y}) a$. Therefore (3.6) is equivalent to a matrix 2-norm minimization, or equivalently, a minimization of the leading singular value, of the matrix $F(\mathbf{x})^T W \Gamma(\mathbf{y})$. Thus (3.1) in the H^s integral formulation reduces to computing

$$\arg \min_{\mathbf{x}, \mathbf{y}, W} \sigma_1 (F(\mathbf{x})^T W \Gamma(\mathbf{y})) \quad \text{subject to } F(\mathbf{x})^T W G(\mathbf{y}) = M, \quad (3.7)$$

where $\sigma_1(\cdot)$ indicates the leading singular value of the matrix. Minimization (3.7) is independent of \mathbf{x} , since by (3.3) $F(\mathbf{x})^T W = M L$, where L is a left inverse of $G(\mathbf{y})$. Therefore \mathbf{x} is chosen by performing a similar minimization on the left, setting an orthonormal basis $\{\lambda_i\}$ for a space $\mathcal{F}_1 \subset \mathcal{G}_0^\perp$, defining the corresponding matrix function $\Lambda(\mathbf{x})$, and computing

$$\arg \min_{\mathbf{x}, \mathbf{y}, W} \sigma_1 (\Lambda(\mathbf{x})^T W G(\mathbf{y})) \quad \text{subject to } F(\mathbf{x})^T W G(\mathbf{y}) = M, \quad (3.8)$$

where similarly the dependence of (3.8) on \mathbf{y} may be dropped since $W G(\mathbf{y})$ is equal to $L^T M$, where L is a left inverse of $F(\mathbf{x})$.

In the symmetric case $\mathcal{F}_0 = \mathcal{G}_0$, $M = I$, $\mathcal{F}_1 = \mathcal{G}_1$, and $m = n = k$ with $\mathbf{x} = \mathbf{y}$, the problem can be greatly simplified. Since $F(\mathbf{x}) = G(\mathbf{y})$, then the minimizations (3.7) and (3.8) are equivalent. Furthermore, as $F(\mathbf{x})$ is square and has full rank, then it is invertible and $F(\mathbf{x})^T W = F(\mathbf{x})^{-1}$, and hence the minimization simplifies to

$$\arg \min_{\mathbf{x}} \sigma_1 (F(\mathbf{x})^{-1} \Gamma(\mathbf{x})). \quad (3.9)$$

In this special case, the “weight” matrix W is dropped from the optimization because it is uniquely determined by the choice of evaluation points \mathbf{x} , which is made possible because the exactness of a bilinear quadrature can be formulated as a matrix equation. In contrast, for classical quadratures there is not always theory proving the existence of weights guaranteeing exactness. In these cases, classical quadratures often eschew exactness and are constructed by moment-fitting methods, which typically minimize the errors in a least-squares sense.

Remark. It is interesting to note that in the symmetric case, the associated approximate orthogonal projection can compute an interpolation, since it takes in function values and returns coefficients in a basis. Thus, interpolation can be viewed as a special case for bilinear quadratures. However, in the general case a bilinear quadrature computes something very different than an interpolation.

In subsequent sections special attention is given to the symmetric case because it is used for evaluating orthogonal projections.

3.2 Error estimates

In this section, upper bounds on several error norms in computing an approximate orthogonal projection of the form (3.5) are estimated.

Theorem 4 (Euclidean norm error estimate). *Let $\mathcal{F}_0, \mathcal{G}_0$ be a dual pair in an inner product space \mathcal{F} and Q a bilinear quadrature of the form (3.4) that is exact on $\mathcal{F}_0 \times \mathcal{G}_0$. Let P_Q be the approximate orthogonal projection onto \mathcal{F}_0 arising from Q with coordinate representation (3.5). If P is the exact orthogonal projection operator onto \mathcal{F}_0 , $\mathcal{G}_1 \subset \mathcal{F}_0^\perp$, and $g = g_0 + g_1 \in \mathcal{G}_0 \oplus \mathcal{G}_1$ such that $g_i \in \mathcal{G}_i$,*

$$\|[P_Q(g) - P(g)]_\beta\|_2 \leq \sigma(Q; \mathcal{F}_0, \mathcal{G}_1) \|g_1\|, \quad (3.10)$$

where $\|\cdot\|_2$ is the Euclidean norm and $[\cdot]_\beta$ indicates representation in the coordinates of β .

Proof. Since $\langle f_i, g \rangle = Q(f_i, g_0)$, then

$$\begin{aligned} \|[P_Q(g) - P(g)]_\beta\|_2 &= \left(\sum_{i=1}^k |Q(f_i, g) - \langle f_i, g \rangle|^2 \right)^{1/2} \\ &= \left(\sum_{i=1}^k |Q(f_i, g) - Q(f_i, g_0)|^2 \right)^{1/2} \\ &= \left(\sum_{i=1}^k |Q(f_i, g_1)|^2 \right)^{1/2}, \end{aligned}$$

the Euclidean norm of the vector $(Q(f_i, g_1)) \in \mathbb{R}^k$. Since for any column vector v , $\|v\|_2$ can be written as $\max_{\alpha \neq 0} \alpha^T v / \|\alpha\|_2$, then the above can be rewritten as

$$\max_{\alpha \neq 0} \frac{1}{\|\alpha\|_2} \sum_{i=1}^k \alpha_i Q(f_i, g_1),$$

where $\alpha = (\alpha_i) \in \mathbb{R}^k$. Since each $f \in \mathcal{F}_0$ can be written as $f = \sum_i \alpha_i f_i$ for a unique α , then

$$\begin{aligned} \max_{\alpha \neq 0} \frac{1}{\|\alpha\|_2} \sum_{i=1}^k \alpha_i Q(f_i, g_1) &= \max_{0 \neq f \in \mathcal{F}_0} \frac{Q(f, g_1)}{\|f\|} \\ &\leq \sigma(Q; \mathcal{F}_0, \mathcal{G}_1) \|g_1\|. \end{aligned}$$

□

Theorem 4 provides an error bound for an approximate orthogonal projection when the projected function g is in $\mathcal{G}_0 \oplus \mathcal{G}_1$. If \mathcal{F}_0 is a space of polynomials, then it is also useful to obtain an error estimate that depends on the regularity of g .

Theorem 5 (Uniform norm error estimates for polynomials). *Let $\mathcal{F} = C(\bar{\Omega})$ with $\Omega \subset \mathbb{R}^d$ a bounded open set with Lipschitz boundary, equipped with the L^2 inner product. Let \mathcal{F}_0 be the set of multivariate polynomials of degree at most n with an orthonormal basis $\beta = \{f_i\}$,*

let $P : \mathcal{F} \rightarrow \mathcal{F}$ be the orthogonal projection onto \mathcal{F}_0 , and suppose P_Q is an approximate orthogonal projection onto \mathcal{F}_0 with coordinate representation (3.5). Then there exists a constant $C > 0$ such that for every $g \in C^{n+1}(\bar{\Omega})$,

$$\|[Pg - P_Qg]_\beta\|_\infty \leq C\|D^{n+1}g\|_{L^\infty}, \quad (3.11)$$

where

$$\|D^{n+1}g\|_{L^\infty} := \sum_{|\alpha|=n+1} \max_{x \in \Omega} |D^\alpha g(x)|$$

and $\|\cdot\|_\infty$ is the vector max norm.

Proof. Using (3.5) and the exactness of P_Q on \mathcal{F}_0 , then writing $g = Pg + (I - P)g = g_0 + g_1$, we have

$$\begin{aligned} \|[Pg - P_Qg]_\beta\|_\infty &= \|F^T W g_1(\mathbf{x})\|_\infty \\ &\leq \|F^T W\|_{\infty \rightarrow \infty} \|(I - P)g\|_{C^0} \\ &\leq \|F^T W\|_{\infty \rightarrow \infty} \|(I - P)(g - q)\|_{C^0}, \end{aligned}$$

where q is any element of \mathcal{F}_0 and $\|\cdot\|_{\infty \rightarrow \infty}$ is the matrix norm induced by the $\|\cdot\|_\infty$ vector norm. Then

$$\|[Pg - P_Qg]_\beta\|_\infty \leq \|F^T W\|_{\infty \rightarrow \infty} (1 + \|P\|_{C^0 \rightarrow C^0}) \|g - q\|_{C^0},$$

where the C^0 operator norm of P is given by

$$\|P\|_{C^0 \rightarrow C^0} = \max_{x \in \Omega} \int_{\Omega} \left| \sum_i f_i(t) f_i(x) \right| dt.$$

Because Ω has Lipschitz boundary, the Deny-Lions/Bramble-Hilbert lemma [11] applies such that for all $g \in C^{n+1}(\Omega)$ there exists a constant $C_{BH} > 0$ (dependent on n and Ω) such that

$$\inf_{q \in \mathcal{F}_0} \|g - q\|_{C^0} \leq C_{BH} \|D^{n+1}g\|_{L^\infty},$$

which combined with the previous inequality yields the desired result with

$$C = \|F^T W\|_{\infty \rightarrow \infty} (1 + \|P\|_{C^0 \rightarrow C^0}) C_{BH}.$$

□

Remark. The requirement that the domain have Lipschitz boundary is not a limitation of bilinear quadratures, but is rather a sufficient condition for the Sobolev extension property, which is needed for the Deny-Lions/Bramble-Hilbert lemma. See [11] for details.

In the presence of error in function evaluation (such as from finite-precision arithmetic), the conditioning of an approximate orthogonal projection is also important to quantify.

Theorem 6. Let P_Q be an approximate orthogonal projection of the form (3.5). If $\delta g(\mathbf{y})$ is the difference between the computed $g(\mathbf{y})$ and its true value, and similarly for $\delta P_Q(g)$, then for a given vector norm $\|\cdot\|$,

$$\frac{\|[\delta P_Q(g)]_\beta\|}{\|[P_Q(g)]_\beta\|} \leq \kappa \frac{\|\delta g(\mathbf{y})\|}{\|g(\mathbf{y})\|},$$

where $\kappa = \kappa(F(\mathbf{x})^T W)$ is the matrix condition number with respect to $\|\cdot\|$.

Proof. From (3.5), $[P_Q(g)]_\beta$ is the matrix-vector multiplication of $F(x)^T W$ on $g(\mathbf{y})$. Assuming this matrix is formed exactly, then the standard perturbation theory for linear systems shows the ratio between the relative forward error and relative backward error of matrix-vector multiplication is bounded above by the condition number of the matrix; see [9] § 2.2. \square

3.3 Connection to classical quadratures

We now investigate how classical and bilinear quadratures are related. While the two are mathematically distinct, we have already seen that for the case of integrating polynomials and trigonometric polynomials on intervals, Gaussian quadrature and trapezoidal rule are optimally efficient. It is therefore of interest to see how bilinear quadratures compare to these cases when computing inner product integrals that can also be evaluated using classical quadratures.

Consider a classical quadrature with nodes $\mathbf{x} = \{x_i\}$ and weights $\mathbf{w} = \{w_i\}$. For a function $h(x) = f(x)g(x)$ with f, g belonging to function spaces \mathcal{F}, \mathcal{G} respectively, the classical quadrature q evaluated on h is a bilinear quadrature Q on f, g given by

$$Q(f, g) = f(\mathbf{x})^T \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_n \end{bmatrix} g(\mathbf{x}) = \mathbf{w}^T h(\mathbf{x}) = q(h).$$

The matrix W is diagonal with entries given by the weights of the classical quadrature. Thus for a general bilinear quadrature of the form (3.4) the entries of W can be viewed as analogues of the weights.

Univariate polynomials

As we have seen, Gaussian quadratures are the most efficient classical quadratures for univariate polynomials on intervals. We now show that they also arise as a special case of a bilinear quadrature in one dimension. We also propose a way to generalize to quadratures evaluating inner products of polynomials on multidimensional domains.

Theorem 7. *The nodes of a Gaussian quadrature of order n are the same as the points \mathbf{x} in the unique bilinear quadrature of order (n, n) for L^2 inner products on $\mathbb{P}_{n-1} \times \mathbb{P}_{n-1}$ that is minimal on $\text{span}\{\phi_n\}$, where ϕ_n is the orthonormal polynomial of degree n . Furthermore, the matrix W in (3.4) is a diagonal matrix whose diagonal entries are the weights of the Gaussian quadrature.*

Proof. Let $\phi_0, \dots, \phi_{n-1}$ be the orthonormal polynomials up to degree $n-1$ such that $\deg \phi_k = k$, $\mathbf{x} = (x_1, \dots, x_n)^T \in \Omega^n$, and define

$$\Phi(\mathbf{x}) = \begin{bmatrix} \phi_0(x_1) & \dots & \phi_{n-1}(x_1) \\ \vdots & & \vdots \\ \phi_0(x_n) & \dots & \phi_{n-1}(x_n) \end{bmatrix}.$$

Then the minimization problem (3.9) becomes

$$\min_{\mathbf{x} \in \Omega^n} \sigma_1 \left(\Phi(\mathbf{x})^{-1} \begin{bmatrix} \phi_n(x_1) \\ \vdots \\ \phi_n(x_n) \end{bmatrix} \right).$$

This is uniquely minimized (up to reordering of the x_i 's) when \mathbf{x} is the set of zeros of ϕ_n in which case $\sigma_1 = 0$. The corresponding bilinear quadrature Q exactly evaluates products where one polynomial has degree $n-1$ and the other has degree n . Then Q has the same evaluation points as a bilinear quadrature formed from the Gaussian quadrature. Since W is unique, then it must be equal to the diagonal matrix with entries given by the weights of the Gaussian quadrature. \square

The above result suggests that an accurate way to compute inner products of polynomials on a multidimensional domain is to utilize a symmetric bilinear quadrature that is exact on $\mathbb{P}_n \times \mathbb{P}_n$ and minimal on $\mathbb{P}_{n+1} \cap \mathbb{P}_n^\perp$. Just as Gaussian quadratures integrate nearly-polynomial functions accurately, bilinear quadratures constructed in the above manner are expected to evaluate inner products of nearly-polynomial functions accurately. We explore this idea further in Chapter 4, in which we give numerical results for such quadratures.

Trigonometric polynomials

Similarly, a natural question is whether computing a bilinear quadrature for L^2 on trigonometric polynomials yields something optimal, namely the trapezoidal rule. Indeed, when the order n is odd, the trapezoidal rule is again a special case of a bilinear quadrature:

Theorem 8. *Let $n > 0$ be an odd integer. Then classical quadratures on T_{n-1} of the form (2.2) are equivalent to symmetric bilinear quadratures of order (n, n) on $T_{(n-1)/2} \times T_{(n-1)/2}$ that are minimal on $\text{span}\{\sin \frac{(n+1)x}{2}, \cos \frac{(n+1)x}{2}\}$.*

Proof. Set $n = 2k + 1$. Since T_{n-1} is rotationally invariant on the circle, then if Q is a symmetric bilinear quadrature on $T_k \times T_k$ of the form (3.4), $\sigma(Q)$ is invariant under rotations of the evaluation points $\mathbf{x} = (x_1, \dots, x_n)$. Therefore without loss of generality $x_1 = 0, x_j \in [0, 2\pi)$. Using a complex exponential basis for T_{n-1} , define

$$F(\mathbf{x}) = \frac{1}{\sqrt{2\pi}} \begin{bmatrix} 1 & \dots & 1 \\ e^{-ikx_2} & \dots & e^{ikx_2} \\ \vdots & & \vdots \\ e^{-ikx_n} & \dots & e^{ikx_n} \end{bmatrix}, \Gamma(\mathbf{x}) = \frac{1}{\sqrt{2\pi}} \begin{bmatrix} 1 & 1 \\ e^{-i(k+1)x_2} & e^{i(k+1)x_2} \\ \vdots & \vdots \\ e^{-i(k+1)x_n} & e^{i(k+1)x_n} \end{bmatrix},$$

and it suffices to prove that choosing $x_j = 2\pi j/n$ solves the minimization problem (3.1).

If $x_j = 2\pi j/n$, then the first column of $F(\mathbf{x})$ is the second column of $\Gamma(\mathbf{x})$, and the last column of $F(\mathbf{x})$ is the first column of $\Gamma(\mathbf{x})$. In this case, $F(\mathbf{x})^{-1}\Gamma(\mathbf{x}) = [e_n \ e_1]$, where e_j is the j -th standard coordinate vector, and hence $\sigma_1(F(\mathbf{x})^{-1}\Gamma(\mathbf{x})) = 1$.

We claim that for any choice of nodes $\mathbf{x} \in [0, 2\pi)^n$ with $x_1 = 0$,

$$\sigma_1(F(\mathbf{x})^{-1}\Gamma(\mathbf{x})) \geq 1. \quad (3.12)$$

If (3.12) is established, then setting $x_j = 2\pi(j-1)/n$ yields a minimal quadrature in \mathcal{Q} . To show this, let \mathbf{u} be the first column of $F(\mathbf{x})^{-1}\Gamma(\mathbf{x})$. We will show that $\|\mathbf{u}\|_2 \geq 1$, from which (3.12) follows. The column \mathbf{u} satisfies the equation

$$F(\mathbf{x})\mathbf{u} = \frac{1}{\sqrt{2\pi}} \begin{bmatrix} 1 \\ e^{-i(k+1)x_2} \\ \vdots \\ e^{-i(k+1)x_n} \end{bmatrix},$$

which is equivalent to the Vandermonde system

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & e^{ix_2} & \dots & e^{i(n-1)x_2} \\ \vdots & \vdots & & \vdots \\ 1 & e^{ix_n} & \dots & e^{i(n-1)x_n} \end{bmatrix} \mathbf{u} = \begin{bmatrix} 1 \\ e^{-ix_2} \\ \vdots \\ e^{-ix_n} \end{bmatrix}.$$

Setting $z_j = e^{ix_j}$, then the entries of \mathbf{u} are the coefficients of a degree $n-1$ complex polynomial $p(z)$ such that $p(z_j) = 1/z_j$. Setting $q(z) = zp(z)$, it suffices to find a degree n polynomial $q(z)$ such that $q(0) = 0$ and $q(z_j) = 1$. Such a q is unique and

$$q(z) = 1 - \prod_{j=1}^n (1 - z/z_j).$$

Then the leading coefficient of q , which is also the leading coefficient of p , has absolute value 1, and hence $\|\mathbf{u}\|_2 \geq 1$. \square

Remark. The trapezoidal rule *uniquely* generates a minimal bilinear quadrature, since in that case $q(z) = \alpha z^n$ for some $|\alpha| = 1$. If x_j 's are not equispaced, $q(z)$ has some nonzero lower-order coefficients.

3.4 The non-invertible case

Up to this point, we have focused on bilinear quadratures that use the minimum number of evaluation points, which allows the matrix function $F(\mathbf{x})$ to be invertible and which greatly simplifies the minimization problem (3.9). Minimizing the number of evaluation points will reduce the computational cost of evaluating a quadrature, but it may be convenient to use more points than optimal. This may allow the use of particular points for which function values are already known, or may increase the accuracy and stability of the quadrature. The Lobatto quadratures for univariate polynomials on the interval are the classic example. In this section, we discuss how to formulate bilinear Lobatto quadratures, which use additional evaluation points. In this case, the matrix function $F(\mathbf{x})$ is non-invertible.

Similar to Gaussian quadratures and trapezoidal rules, a Lobatto quadrature of order $n + 1$ corresponds to a symmetric bilinear quadrature that is exact on $\mathbb{P}_{n-1} \times \mathbb{P}_{n-1}$ and minimal on \mathbb{P}_n . The W matrix is diagonal and the matrix $F(\mathbf{x})$ is given by

$$F(\mathbf{x}) = \begin{bmatrix} \phi_1(a) & \dots & \phi_k(a) \\ \phi_1(x_1) & & \phi_k(x_1) \\ \vdots & & \vdots \\ \phi_1(x_{n-1}) & & \phi_k(x_{n-1}) \\ \phi_1(b) & \dots & \phi_k(b) \end{bmatrix}.$$

Unlike in the Gaussian quadrature case, the matrix $F = F(\mathbf{x})$ is not square, so there exists infinitely many matrices W satisfying (3.3). Therefore, the simplified minimization condition (3.9) cannot be employed, and one must optimize over both the quadrature nodes \mathbf{x} and matrices W . In general, suppose F is $m \times k$ and G is $n \times k$, both with full column rank. Then all matrices W satisfying (3.3) are of the form

$$W = (F^T)^+ M G^+ + Y - F F^+ Y G G^+, \quad (3.13)$$

where Y is an arbitrary $m \times n$ matrix. In the symmetric case $F = G$ and $M = I$, a minimal bilinear quadrature is found through the unconstrained minimization

$$\text{Find } Y \in \mathbb{R}^{m \times m} \text{ and } \mathbf{x} \text{ minimizing } \sigma_1(F^+ \Gamma + F^T Y (I - F F^+) \Gamma) \quad (3.14)$$

While computationally more expensive, this optimization procedure can be used to compute symmetric bilinear Lobatto quadratures. First fix points \mathbf{x}_0 that the bilinear quadrature is required to use, then construct the (typically non-square) matrix function $F(\mathbf{x}_0, \mathbf{x})$, where only the points \mathbf{x} are varying. Then minimize according to (3.14). This procedure is applicable for arbitrary domains Ω , any space of continuous functions, and any inner product on that space.

3.5 Change of variables

One of the major limitations of quadrature rules, both classical and bilinear, is that they may lose accuracy under an arbitrary change of coordinates. However, when Φ is an affine

invertible change of variables, given a bilinear quadrature computing $L^2(\Omega)$ inner products, a new quadrature can be cheaply constructed for $L^2(\Phi(\Omega))$ inner products. For continuous functions f_i on Ω , set

$$\tilde{f}_i(\Phi(x)) = f_i(x) |\det(D\Phi)|^{-1/2}.$$

Then

$$\langle \tilde{f}_i, \tilde{f}_j \rangle_{L^2(\Phi(\Omega))} = \int_{\Phi(\Omega)} \tilde{f}_i(y) \tilde{f}_j(y) dy = \int_{\Omega} f_i(x) f_j(x) dx = \langle f_i, f_j \rangle_{L^2(\Omega)}.$$

The Jacobian $D\Phi$ is constant when Φ is affine, so if the bilinear quadrature on $L^2(\Omega)$ exact on $\mathcal{F}_0 \times \mathcal{G}_0$ is

$$Q(f, g) = f(\mathbf{x})^T W g(\mathbf{y}),$$

a bilinear quadrature on $L^2(\Phi(\Omega))$ for $(\mathcal{F}_0 \circ \Phi^{-1}) \times (\mathcal{G}_0 \circ \Phi^{-1})$ is given by

$$\tilde{Q}(\tilde{f}, \tilde{g}) = \tilde{f}(\Phi(\mathbf{x}))^T \tilde{W} \tilde{g}(\Phi(\mathbf{y})), \quad \tilde{W} = W |\det(D\Phi)|^{-1}. \quad (3.15)$$

For an H^s inner product with $s > 0$, in general a new bilinear quadrature cannot be cheaply constructed under a change of variables. However, when $\Phi(x) = \lambda Ux + b$ is affine with $\lambda \in \mathbb{R}$ and U a unitary matrix, a change of variables can still be performed at low cost. For example, let $s = 1$ and W be the matrix in a bilinear quadrature of form (3.4) computing $H^1(\Omega)$ inner products. Then write $W = W_0 + W_1$, where W_0 is the matrix for a bilinear quadrature that computes $L^2(\Omega)$ inner products on the same function space. Then a new bilinear quadrature for $H^1(\Phi(\Omega))$ is formed with the matrix

$$\tilde{W} = |\lambda|^{-d} W_0 + |\lambda|^{-d-2} W_1,$$

where d is the Euclidean dimension of the domain Ω , and the evaluation points are mapped by Φ .

For non-affine changes of variables, in general it is not possible to construct a new bilinear quadrature from an old one. Because the Jacobian is non-constant, the W matrix will have to recomputed entirely by solving (3.7) again.

Chapter 4

Computation

4.1 Construction

In this section, a numerical procedure to produce symmetric bilinear quadrature rules is developed and tested.

Orthogonalization

For a function space \mathcal{F}_0 , one may initially have a numerical routine to evaluate (up to machine precision) basis functions ψ_1, \dots, ψ_k for \mathcal{F}_0 that are not orthonormal. To construct a bilinear quadrature according to (3.7), however, orthonormal basis functions f_1, \dots, f_k must be evaluated. To do this, the Gram matrix M consisting of inner products $\langle \psi_i, \psi_j \rangle = M_{ij}$ is first computed. These inner products may be computed exactly, if closed-form expressions for the relevant integrals are known, or approximated with high accuracy using an existing, high-order classical quadrature rule for the integration region Ω . During the construction of a bilinear quadrature, the orthogonalization process is the only instance during which the geometry of the integration region Ω is explicitly used. Once computed, the orthonormal basis functions capture the information pertaining to the behavior of \mathcal{F}_0 on Ω .

After M is known, the matrix of evaluations $F(\mathbf{x})$ for the orthonormal functions is computed by performing a “weighted” QR decomposition on the matrix of evaluations $\Psi(\mathbf{x})$ for the known basis functions. We assume that the inner products $\langle \psi_i, \psi_j \rangle = M_{ij}$ can be computed exactly, and we seek an upper-triangular matrix R such that

$$\Psi(\mathbf{x}) = F(\mathbf{x})R, \quad (4.1)$$

so that if, for a given \mathbf{x} , a weight matrix W satisfies

$$\Psi(\mathbf{x})^T W \Psi(\mathbf{x}) = M, \quad (4.2)$$

then $F(\mathbf{x})$ satisfies

$$F(\mathbf{x})^T W F(\mathbf{x}) = I. \quad (4.3)$$

Substituting (4.1) into (4.2) and using (4.3), we find that $M = R^T R$. Therefore the following simple orthogonalization procedure can be used:

1. Compute the Cholesky factorization $M = R^T R$.
2. Given \mathbf{x} , solve the triangular matrix system $\Psi(\mathbf{x}) = F(\mathbf{x})R$ for the unknown matrix $F(\mathbf{x})$.

Next, the same procedure is used to evaluate orthonormal basis functions for \mathcal{F}_1 in order to construct the matrix $\Gamma(\mathbf{x})$. Note that while the choice of \mathcal{F}_0 is usually determined by the user's application, there is no canonical choice of $\mathcal{F}_1 \subseteq \mathcal{F}_0^\perp$, and different choices for \mathcal{F}_1 will yield bilinear quadratures that are accurate under different conditions.

Nonlinear optimization

For the invertible symmetric case we have reduced our problem to the minimization problem (3.7):

$$\text{Find } \mathbf{x} \text{ minimizing } \sigma_1(F(\mathbf{x})^{-1}\Gamma(\mathbf{x})).$$

This is a nonlinear optimization problem in $d \cdot k$ variables, where d is the dimension of the integration region Ω and $k = \dim(\mathcal{F}_0)$.

The problem of minimizing the largest singular value of a matrix function $A(\mathbf{x})$ is equivalent to minimizing the largest eigenvalue of the symmetric positive semidefinite matrix $A^T(\mathbf{x})A(\mathbf{x})$. This type of eigenvalue optimization problem has been extensively studied in its own right; see [19] [22].

Often $F(\mathbf{x})$ and $\Gamma(\mathbf{x})$, but not their derivatives, can be accurately computed. Also, the multiplicity of the largest singular values are generally unknown. Consequently, a quasi-Newton method is ideal for the optimization procedure. The objective function is non-convex and typically has multiple local minima, so the optimization procedure is run with many initial guesses. Furthermore, in the presence of many nearby local minima, after each convergent result, the computed points can be perturbed by a small value δ and the procedure run again with perturbed points as another initial guess. This is repeated until suitable convergence. While this procedure may be expensive, computing a quadrature is typically a one-time cost, after which the quadrature can be used repeatedly for its applications.

In our numerical experiments, we employ a quasi-Newton method with BFGS updates as implemented as part of Matlab's `fminunc` routine [3, 12, 13, 21]. Since $F(\mathbf{x})^{-1}\Gamma(\mathbf{x})$ is a small, dense matrix, its norm is computed by calculating its full SVD. Up to 10^5 random initial sets of points uniformly distributed across the domain are used, and the procedure is iterated until convergence in double-precision arithmetic.

For our numerical implementation we do not enforce the constraint that the evaluation points x_i remain in the integration domain Ω . While in general the full constrained minimization problem may be necessary, we have empirically observed that it is not necessary for

our construction of quadratures on polynomials. This can be explained by observing that the orthogonal polynomials grow rapidly outside of a convex domain Ω ; thus points outside the domain are not expected to be good candidates for the solution to the minimization problem.

Remark. In some cases it may be possible to explicitly compute the gradient of the objective, in which case a different numerical optimization method can be used. However, we have chosen to use BFGS since it is robust and is applicable to many cases.

4.2 Approximate orthogonal projections

One of the natural applications of bilinear quadratures is the computation of approximate orthogonal projections. Recall that given the space \mathcal{F}_0 on Ω with L^2 -orthonormal basis $\beta = \{f_i\}$, orthogonal projection operator P onto \mathcal{F}_0 , and given $g \in C^\infty(\Omega)$, we wish to compute

$$[Pg]_\beta \approx \begin{bmatrix} \langle f_1, g \rangle_{L^2} \\ \vdots \\ \langle f_k, g \rangle_{L^2} \end{bmatrix}.$$

The column vector $[Pg]_\beta$ can be computed using a bilinear quadrature, or using a classical quadrature for each vector component $\int_\Omega f_i g$. Thus the calculation of approximate orthogonal projections provides a good test case for comparing the performance of the methods generated using our proposed framework to that of existing classical quadratures.

Bilinear quadratures on the triangle

In practical applications one of the most important cases to consider is the L^2 product of polynomials on a simplex. For example, in the finite element method one typically solves a two-dimensional elliptic PDE locally on polynomials supported on triangular domains. The discretization requires computing a number of inner products.

Because the space of polynomials is affine-invariant it suffices to find evaluation points for polynomials on a reference triangle. Given a bilinear quadrature on a reference triangle, a bilinear quadrature on any other triangle can be cheaply obtained using the change of variables formula (3.15). A basis of orthogonal polynomials on the right triangle with vertices $(-1, -1), (-1, 1), (1, -1)$ is given by

$$K_{m,n}(x, y) = \left(\frac{1-v}{2}\right)^m P_m\left(\frac{2x+y+1}{1-y}\right) P_n^{2m+1,0}(y), \quad (4.4)$$

where P_m is the m th Legendre polynomial and $P_n^{\alpha,\beta}$ is the n th Jacobi polynomial with parameters α, β . These functions can be computed efficiently and stably as in [26].

Using this basis, symmetric bilinear quadratures exact for the L^2 inner product over this right triangle on $\mathcal{F}_0 = \mathbb{P}_n$ and minimal on $\mathcal{F}_1 = \mathbb{P}_{n+1} \cap \mathbb{P}_n^\perp$ were computed. The minimal

Table 4.1: Numerical results for k -point bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for L^2 on the interior of the reference right triangle.

n	k	σ	$\kappa_\infty(F^T W)$
0	1	0.00000	1.00000e+0
1	3	0.14507	2.82218e+0
2	6	0.30373	6.29185e+0
3	10	0.47762	1.15455e+1
4	15	0.65817	2.03810e+1
5	21	0.78394	3.39955e+1
6	28	0.87930	4.71065e+1
7	36	0.95305	8.48889e+1
8	45	1.05595	1.09107e+2

number of evaluation points were used, in which case the number of points required is

$$k = \dim(\mathbb{P}_n) = \binom{n+2}{2}.$$

In Table 4.1, for each computed bilinear quadrature rule, the minimized largest singular value $\sigma = \sigma_1(F(\mathbf{x})^{-1}\Gamma(\mathbf{x}))$ is given, along with the ∞ -norm condition number of the matrix for the approximate orthogonal projection.

In Figure 4.1, the evaluation points of two bilinear quadrature rules on the equilateral triangle are shown. Notice that the points possess some symmetries. The expectation that quadrature points for polynomials should have some symmetries has been exploited in the past to reduce the complexity of searching for classical quadratures [26]. In the quasi-Newton method used to solve (3.9), however, no symmetry conditions were explicitly enforced.

We now compare the computed bilinear quadratures on triangles against existing high-order classical quadrature schemes on triangles in the setting of orthogonal projections. Since the approximate orthogonal projection matrix $F^T W$, weights of the classical quadrature, and locations of evaluation points are all precomputed, the flop cost for each method is solely determined by the number of evaluation points needed. The 28-point bilinear quadrature as computed in Table 4.1 was utilized. For comparison we chose two different 28-point classical quadratures, each exact on polynomials of degree up to 11, due to Dunavant [10] and Xiao and Gimbutas [26], respectively. These quadratures were computed using the libraries available from [4]. Both classical quadratures were similarly transformed to an equilateral triangle of side length 1.

To test the accuracy of the associated approximate orthogonal projection, we wish to project various types of smooth functions onto the polynomial space. We draw the projected function g from four different probability distributions of functions, which we denote by $\mathbb{P}'_5, \mathbb{P}'_6, C$, and TP .

Figure 4.1: Evaluation points for bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for L^2 on the interior of an equilateral triangle, for $n = 4, 8$.

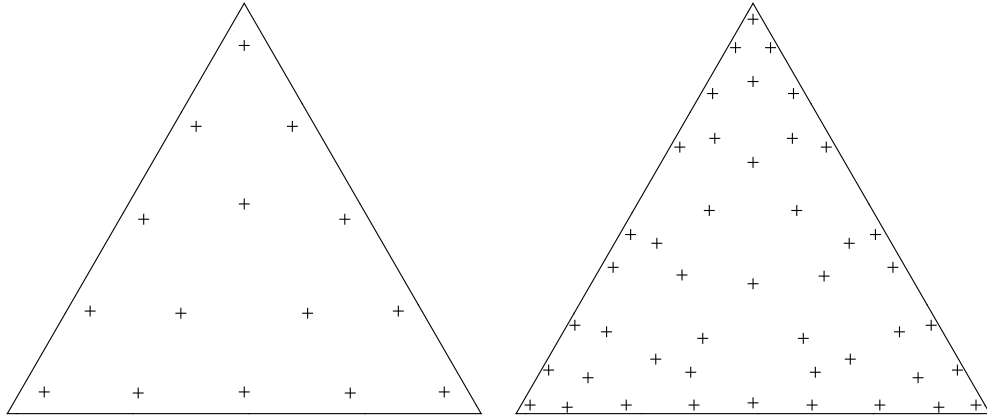


Table 4.2: Average ℓ^2 -norm relative error in computing approximate orthogonal projection coefficients onto \mathbb{P}_6 for four different sets of functions on the triangle.

	\mathbb{P}'_5	\mathbb{P}'_6	C	TP
Dunavant	9.38e-14	3.97e-01	4.97e-05	9.06e-03
Xiao/Gimbutas	3.29e-15	2.73e-01	1.91e-05	4.74e-03
Bilinear	3.92e-15	3.99e-15	6.74e-06	1.71e-03

We define

$$\mathbb{P}'_n := \{g \in \mathbb{P}_n : \|g\|_{L^2} = 1\},$$

with probability measure given by drawing a random vector of coefficients uniformly in $[-1, 1]^k$, and then normalizing the coefficients to have ℓ^2 -norm 1, and using those as the basis coefficients for the orthonormal polynomials on the triangle.

The set C contains smooth functions with slow decay, and is defined by functions of the form

$$g(x, y) = \frac{1}{1 + (a_1x + a_2y)^2},$$

where $a = (a_1, a_2)$ is drawn uniformly from the unit circle.

The set TP contains smooth non-polynomial functions with oscillations, and has elements of the form

$$g(x, y) = e^{a_1x+a_2y} \cos(4b_1x + 4b_2y)p(x, y),$$

where parameters (a_1, a_2) and (b_1, b_2) are both drawn uniformly from the unit circle, and $p(x, y)$ is a random element of \mathbb{P}'_2 with L^2 norm 1 as chosen in the same manner as for the first two cases.

Table 4.3: Numerical results for k -point bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for L^2 on the interior of a square.

n	k	σ	$\kappa_\infty(F^T W)$
0	1	0.00000	1.00000e+0
1	3	0.67739	2.91852e+0
2	6	0.79523	7.50137e+0
3	10	0.92888	1.17526e+1
4	15	0.97590	2.68367e+1
5	21	0.99701	3.14417e+1
6	28	1.00066	6.42937e+1
7	36	1.00711	7.34237e+1
8	45	1.00759	1.04579e+2

For each randomly chosen function g , we computed the column vector $[P_Q g]_\beta$ using the three quadrature methods. The exact value $[Pg]_\beta$ was computed with a 295-point classical quadrature that exactly integrates polynomials up to degree 40, as computed in [26]. The ℓ^2 norm relative error was averaged over 10^4 randomly generated g for each of the four classes of functions. The resulting average relative errors are shown in Table 4.2.

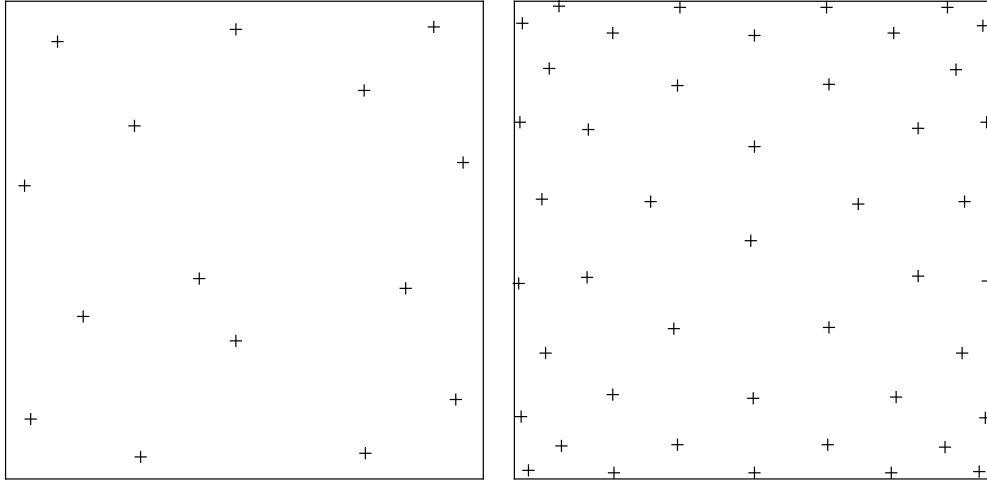
On \mathbb{P}'_5 , all three quadrature rules achieve very high accuracy, with the Dunavant quadrature losing one digit of accuracy and both Xiao/Gimbutas and bilinear quadratures correctly computing the orthogonal projection up to double precision. This is expected since all quadratures are designed to integrate such polynomial functions exactly.

On \mathbb{P}'_6 , neither classical quadratures are accurate to full precision because both classical quadratures are only capable of exactly integrating polynomials of degree up to 11. Since the bilinear quadrature can exactly integrate $\mathbb{P}_6 \times \mathbb{P}_6$, it has mean error on the order of machine precision.

On the sets C and TP , none of the quadratures are accurate to machine precision since none of the functions are polynomials. However, the bilinear quadrature achieves better accuracy than the classical quadratures despite having the same number of evaluation points.

The existing classical quadratures are already very good, integrating non-polynomial functions from C and TP with several digits of accuracy. Additionally, the classical quadrature of Xiao/Gimbutas performs better than the Dunavant quadrature in all four cases. However, the bilinear quadrature was as good or better than the classical quadratures in each case, despite using the same number of evaluations. This result is partly explained by the fact that bilinear quadratures are specifically designed for the orthogonal projection problem, while classical quadratures are designed for evaluating a linear functional.

Figure 4.2: Evaluation points for bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for L^2 on the interior of a square, for $n = 4, 8$.



Bilinear quadratures on the square

We now compute and test bilinear quadratures for L^2 inner products of polynomials on the interior of a square. We observe that, just as in the case of triangles, minimizing according to (3.9) produces well-behaved evaluation points.

Producing classical quadratures for polynomials on the square is an interesting case, as it is easy to construct one by taking the tensor product of two Gaussian quadratures on an interval. The resulting quadrature, however, exactly integrates basis functions of the form $x^\alpha y^\beta$ with $0 \leq \alpha \leq n$, $0 \leq \beta \leq n$ for some n , rather than integrating polynomials whose *total degree* does not exceed some value.

To produce a bilinear quadrature for polynomials on $[-1, 1]^2$, the orthogonal polynomials $P_n(x)P_m(y)$ are chosen, where P_n is the n th Legendre polynomial. Table 4.3 shows the minimized leading singular value σ and matrix condition number κ_∞ for several k -point bilinear quadratures on the square. Interestingly, the evaluation points on the square do not appear to obey any symmetries, though this might be partly explained by the fact that order of the symmetry group does not necessarily divide the number of points. Attempting to enforce symmetry conditions before running the optimization procedure leads to quadrature rules whose σ value is larger than if symmetry is not enforced.

The computed bilinear quadratures were tested against the best-known existing classical quadratures on the square. Just as with quadratures on triangles, the error metric used is the ℓ^2 norm relative error on the components of an approximate orthogonal projection. Exact values were taken to be given by a higher-order 167-point classical quadrature. The 21-point and 45-point bilinear quadratures, as computed in Table 4.3, were compared against 22-point and 44-point classical quadratures, respectively, as computed in [26]. Unfortunately, there are no classical quadrature rules available that use exactly 21 or 45 evaluation points. However,

Table 4.4: Average ℓ^2 -norm relative error in computing approximate orthogonal projection coefficients onto \mathbb{P}_5 for four different sets of functions on the square.

	\mathbb{P}'_5	\mathbb{P}'_6	C	TP
Xiao/Gimbutas (22 pts)	9.69e-16	4.77e-01	8.58e-03	3.26e-01
Bilinear (21 pts)	1.10e-15	5.28e-01	1.03e-02	3.74e-01

Table 4.5: Average ℓ^2 -norm relative error in computing approximate orthogonal projection coefficients onto \mathbb{P}_8 for four different sets of functions on the square.

	\mathbb{P}'_8	\mathbb{P}'_9	C	TP
Xiao/Gimbutas (44 pts)	4.15e-01	5.98e-01	2.44e-03	7.96e-02
Bilinear (45 pts)	1.97e-15	4.34e-01	6.17e-04	3.22e-02

since the Xiao/Gimbutas quadratures are known to be very high quality, a comparison can still be used to validate the effectiveness of constructing bilinear quadratures using the chosen optimization problem. The results are shown in Tables 4.4 and 4.5.

Comparing the 21-point bilinear quadrature and the 22-point classical quadrature, we observe that for all four classes of functions, both methods perform similarly, though in each case, the Xiao/Gimbutas quadrature performs slightly better. This is possibly due to the fact that it uses one additional evaluation point. As the bilinear quadrature is designed to integrate products of degree 5 polynomials exactly, and the classical quadrature is designed to integrate degree 10 polynomials exactly, both methods compute the orthogonal projection of \mathbb{P}'_5 onto \mathbb{P}_5 with error on the order of machine precision.

Comparing the 45-point bilinear quadrature and the 44-point classical quadrature, we observe that for all three of the four classes of functions, both methods perform similarly, though the bilinear quadrature performs slightly better. In this case, the bilinear quadrature uses one additional evaluation point. The bilinear quadrature integrates products of degree 8 polynomials exactly, while the classical quadrature exactly integrates up to degree 15, which explains the discrepancy for projections of \mathbb{P}'_8 onto \mathbb{P}_8 .

We conclude that overall both methods perform similarly, with some differences resulting from using slightly different numbers of evaluation points. This is a relatively favorable result for bilinear quadratures, since it is able to compete with the best quadratures computed to date, despite using a general framework that does not depend on the function spaces or the integration domain.

Table 4.6: Numerical results for k -point bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for L^2 on the unit disk.

n	k	σ	$\kappa_\infty(F^T W)$
0	1	0.00000	1.00000e+0
1	3	0.67617	3.04857e+0
2	6	0.79868	5.50559e+0
3	10	0.89712	1.01509e+1
4	15	0.94133	1.59179e+1
5	21	0.97804	2.24193e+1
6	28	1.00337	3.94055e+1
7	36	1.02908	5.60579e+1
8	45	1.07413	6.75064e+1

Bilinear quadratures on the disk

We now compute and test bilinear quadratures for L^2 inner products of polynomials on the interior of a disk. To construct the bilinear quadrature on the unit disk, we use the Zernike polynomials $Z_{m,n}(r, \theta)$, an orthogonal basis of polynomials defined in polar coordinates by

$$Z_{m,n}(r, \theta) := Q_{m,n}(r) \cos(m\theta), \quad Z_{-m,n}(r, \theta) := Q_{m,n}(r) \sin(m\theta),$$

$$Q_{m,n}(r) := \sum_{k=0}^{(n-m)/2} (-1)^k \binom{n-k}{k} \binom{n-2k}{\frac{n-m}{2}-k} r^{n-2k},$$

where $n \geq m \geq 0$ are integers and $n - m$ is even.

In optics and optical engineering, the orthogonal projection onto the space of Zernike polynomials on the disk is known as the *discrete Zernike transform* and is useful for determining optical aberrations [24]. The literature on efficient numerical methods for integration on the disk is much less extensive than those for simplices, with notable surveys in [7, 23]. Therefore, accurate and efficient bilinear quadratures on the disk may prove valuable in these applications.

Table 4.6 shows the minimized leading singular value σ and matrix condition number κ_∞ for several k -point bilinear quadratures on the unit disk.

The 21-point bilinear quadrature on the disk, exact on products of polynomials up to degree 5, was compared with a 21-point classical quadrature as presented by Stroud [23]. This classical quadrature is exact for polynomials up to degree 9. The two methods were similarly tested for their relative accuracy in computing orthogonal projections onto \mathbb{P}_5 , similar to the tests on the triangle and square. Exact values were computed using a high-order 200-point radially symmetric classical quadrature whose Fortran 90 implementation is available from [4]. The errors are shown in Table 4.7.

Figure 4.3: Evaluation points for bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for L^2 on the unit disk, for $n = 4, 6$.

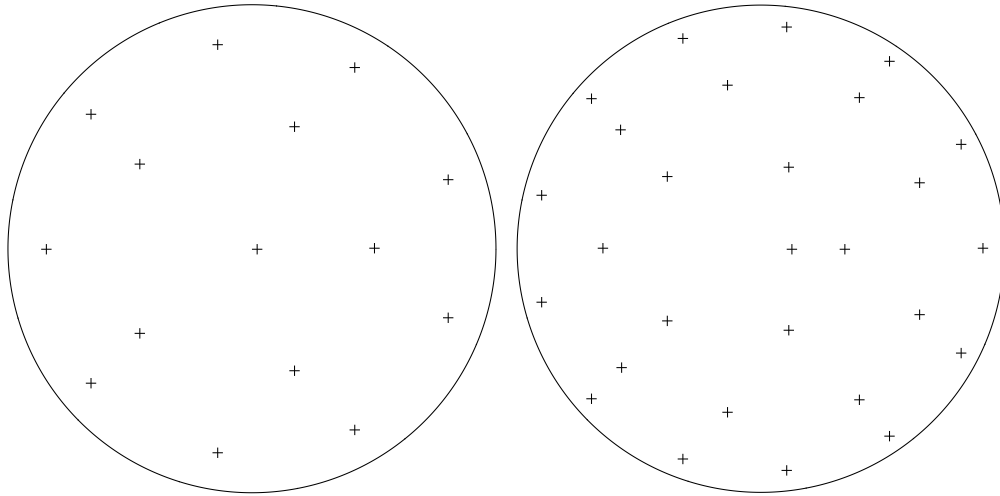


Table 4.7: Average ℓ^2 -norm relative error in computing approximate orthogonal projection coefficients onto \mathbb{P}_5 for four different sets of functions on the unit disk.

	\mathbb{P}'_4	\mathbb{P}'_5	C	TP
Stroud	3.01e-15	4.39e-01	3.85e-03	2.97e-01
Bilinear	2.24e-15	2.44e-15	5.10e-03	2.39e-01

The orthogonal projection errors for \mathbb{P}'_4 onto \mathbb{P}_5 were on the order of machine precision, as expected. For \mathbb{P}'_5 , the bilinear quadrature is exact up to machine precision, since it is designed for products of polynomials up to degree 5, while the Stroud quadrature is inaccurate since it can only exactly integrate up to polynomial degree 9. For smooth, non-polynomial function classes C and TP , the two quadrature methods perform similarly. Once again, this is a favorable result for bilinear quadratures since the framework used to construct them is very general. In contrast, the Stroud quadrature for polynomials was constructed using an algebraic, problem-specific approach for finding a closed form solution to the system of equations that guarantee the classical quadrature is exact.

Bilinear quadratures for the Sobolev inner product

While we have exhaustively tested the performance of bilinear quadratures on L^2 inner products because they can be compared against existing classical quadratures, one of the advantages of our framework is that they can be used to evaluate any continuous bilinear

Table 4.8: Numerical results for bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for $H^1[-1, 1]$ with the weight function $A(x) = 1 + x^2$.

n	k	σ	$\kappa_\infty(F^T W)$
1	2	0.00000	5.00000e+0
2	3	0.00000	1.38132e+1
3	4	0.00000	2.72011e+1
4	5	0.00000	6.59254e+1
5	6	0.00000	1.21461e+2
6	7	0.00000	1.86818e+2
7	8	0.00000	2.86549e+2
8	9	0.00000	4.22824e+2

Table 4.9: Numerical results for bilinear quadratures on $\mathbb{P}_n \times \mathbb{P}_n$ for $H^1[-1, 1]$ with the weight function $A(x) = e^x$.

n	k	σ	$\kappa_\infty(F^T W)$
1	2	0.00000	4.52560e+0
2	3	0.00000	1.30446e+1
3	4	0.00000	2.49183e+1
4	5	0.00000	5.13338e+1
5	6	0.00000	9.28987e+1
6	7	0.00000	1.50063e+2
7	8	0.00000	2.28284e+2
8	9	0.00000	3.30651e+2

form. One such bilinear form is the Sobolev inner product

$$\langle f, g \rangle_{H^1} = \int_{\Omega} Df(x) \cdot A(x) Dg(x) + f(x)g(x) dx,$$

where $A(x)$ is symmetric positive definite $d \times d$ matrix function on Ω . Such inner products might arise, for example, in the solution to the heat equation using the finite element method. One advantage of a bilinear quadrature for H^1 is that the above integral can be numerically evaluated using only point evaluations of f, g and does not require evaluating any derivatives.

For $\Omega = [-1, 1]$, bilinear quadratures for H^1 on $\mathbb{P}_n \times \mathbb{P}_n$ and minimal on $\mathbb{P}_{n+1} \cap \mathbb{P}_n^\perp$ were computed for two positive weight functions $A(x) = 1 + x^2$ and $A(x) = e^x$. Orthogonalization was performed by starting with the Legendre polynomials and computing the Gram matrix M using a 40-point classical Gaussian quadrature.

In Tables 4.8 and 4.9 the singular value σ and condition number κ_∞ are shown for the two bilinear quadratures for H^1 . In all cases, σ is zero up to machine precision, since the

exact solution to the minimization (3.7) is the roots of the $(n + 1)$ th-degree H^1 -orthogonal polynomial, just as for Gaussian quadratures. We observe that the condition number of the approximation projection matrix $F^T W$ is slightly larger than in the L^2 case. This can be explained by the fact that small perturbations in the function values can lead to large perturbations in the derivatives.

Bibliography

- [1] W. Robert Boland and C.S. Duris. “Product type quadrature formulas”. In: *BIT Numerical Mathematics* 11.2 (1971), pp. 139–158. ISSN: 0006-3835. DOI: 10.1007/BF01934362. URL: <http://dx.doi.org/10.1007/BF01934362>.
- [2] James Bremer, Zydrunas Gimbutas, and Vladimir Rokhlin. “A nonlinear optimization procedure for generalized Gaussian quadratures”. In: *SIAM J. Sci. Comput.* 32.4 (2010), pp. 1761–1788. ISSN: 1064-8275. DOI: 10.1137/080737046. URL: <http://dx.doi.org/10.1137/080737046>.
- [3] C. G. Broyden. “The Convergence of a Class of Double-rank Minimization Algorithms”. In: *IMA Journal of Applied Mathematics* 6.1 (1970), pp. 76–90. DOI: 10.1093/imamat/6.1.76. eprint: <http://imamat.oxfordjournals.org/content/6/1/76.full.pdf+html>. URL: <http://imamat.oxfordjournals.org/content/6/1/76.abstract>.
- [4] John Burkardt. *Source codes in Fortran90; Source codes in Matlab*. <http://people.sc.fsu.edu/~jburkardt/>. 2015.
- [5] Y. Chen. “Inner product quadratures”. In: *ArXiv e-prints* (May 2012). arXiv: 1205.0601 [math.NA].
- [6] H. Cheng, V. Rokhlin, and N. Yarvin. “Nonlinear optimization, quadrature, and interpolation”. In: *SIAM J. Optim.* 9.4 (1999), 901–923 (electronic). ISSN: 1052-6234. DOI: 10.1137/S1052623498349796. URL: <http://dx.doi.org/10.1137/S1052623498349796>.
- [7] R. Cools and K. J. Kim. “A survey of known and new cubature formulas for the unit disk”. In: *Korean Journal of Computational & Applied Mathematics* 7.3 (2000), pp. 477–485. ISSN: 1865-2085. DOI: 10.1007/BF03012263. URL: <http://dx.doi.org/10.1007/BF03012263>.
- [8] Philip J. Davis and Philip Rabinowitz. *Methods of numerical integration*. Second. Computer Science and Applied Mathematics. Academic Press, Inc., Orlando, FL, 1984, pp. xiv+612. ISBN: 0-12-206360-0.
- [9] James W. Demmel. *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997, pp. xii+419. ISBN: 0-89871-389-7. DOI: 10.1137/1.9781611971446. URL: <http://dx.doi.org/10.1137/1.9781611971446>.

- [10] D. A. Dunavant. “High degree efficient symmetrical Gaussian quadrature rules for the triangle”. In: *International Journal for Numerical Methods in Engineering* 21.6 (1985), pp. 1129–1148. ISSN: 1097-0207. DOI: 10.1002/nme.1620210612. URL: <http://dx.doi.org/10.1002/nme.1620210612>.
- [11] Alexandre Ern and Jean-Luc Guermond. *Theory and practice of finite elements*. Vol. 159. Applied Mathematical Sciences. Springer-Verlag, New York, 2004, pp. xiv+524. ISBN: 0-387-20574-8. DOI: 10.1007/978-1-4757-4355-5. URL: <http://dx.doi.org/10.1007/978-1-4757-4355-5>.
- [12] R. Fletcher. “A new approach to variable metric algorithms”. In: *The Computer Journal* 13.3 (1970), pp. 317–322. DOI: 10.1093/comjnl/13.3.317. eprint: <http://comjnl.oxfordjournals.org/content/13/3/317.full.pdf+html>. URL: <http://comjnl.oxfordjournals.org/content/13/3/317.abstract>.
- [13] Donald Goldfarb. “A family of variable-metric methods derived by variational means”. In: *Math. Comp.* 24 (1970), pp. 23–26. ISSN: 0025-5718.
- [14] Gene H. Golub and John H. Welsch. “Calculation of Gauss quadrature rules”. In: *Math. Comp.* 23.106, loose microfiche suppl (1969), A1–A10. ISSN: 0025-5718.
- [15] J.D. Gribble. “Interpolatory inner product quadrature formulas”. In: *BIT Numerical Mathematics* 20.4 (1980), pp. 466–474. ISSN: 0006-3835. DOI: 10.1007/BF01933640. URL: <http://dx.doi.org/10.1007/BF01933640>.
- [16] L. Knockaert. “A bilinear quadrature rule for the finite Hankel transform”. In: *AFRICON 2007*. 2007, pp. 1–4. DOI: 10.1109/AFRICON.2007.4401442.
- [17] Eberhard Luik and Karl Zeller. “Numerische Approximation von Bilinearformen”. In: *Results in Mathematics* 11.3-4 (1987), pp. 374–383. ISSN: 0378-6218. DOI: 10.1007/BF03323280. URL: <http://dx.doi.org/10.1007/BF03323280>.
- [18] Joseph F. McGrath. “Gaussian product-type quadratures”. In: *Applied Mathematics and Computation* 5.3 (1979), pp. 265–280. ISSN: 0096-3003. DOI: [http://dx.doi.org/10.1016/0096-3003\(79\)90018-3](http://dx.doi.org/10.1016/0096-3003(79)90018-3). URL: <http://www.sciencedirect.com/science/article/pii/0096300379900183>.
- [19] Michael L. Overton and Robert S. Womersley. “Second derivatives for optimizing eigenvalues of symmetric matrices”. In: *SIAM J. Matrix Anal. Appl.* 16.3 (1995), pp. 697–718. ISSN: 0895-4798. DOI: 10.1137/S089547989324598X. URL: <http://dx.doi.org/10.1137/S089547989324598X>.
- [20] Ernest K. Ryu and Stephen P. Boyd. “Extensions of Gauss Quadrature Via Linear Programming”. In: *Foundations of Computational Mathematics* (2014), pp. 1–19. ISSN: 1615-3375. DOI: 10.1007/s10208-014-9197-9. URL: <http://dx.doi.org/10.1007/s10208-014-9197-9>.
- [21] D. F. Shanno. “Conditioning of quasi-Newton methods for function minimization”. In: *Math. Comp.* 24 (1970), pp. 647–656. ISSN: 0025-5718.

- [22] Alexander Shapiro and Michael K. H. Fan. “On eigenvalue optimization”. In: *SIAM J. Optim.* 5.3 (1995), pp. 552–569. ISSN: 1052-6234. DOI: 10.1137/0805028. URL: <http://dx.doi.org/10.1137/0805028>.
- [23] A. H. Stroud. *Approximate calculation of multiple integrals*. Prentice-Hall Series in Automatic Computation. Englewood Cliffs, N.J.: Prentice-Hall Inc., 1971, pp. xiii+431.
- [24] William J. Tango. “The circle polynomials of Zernike and their application in optics”. In: *Applied physics* 13.4 (1977), pp. 327–332. ISSN: 1432-0630. DOI: 10.1007/BF00882606. URL: <http://dx.doi.org/10.1007/BF00882606>.
- [25] B. Vioreanu and V. Rokhlin. “Spectra of Multiplication Operators as a Numerical Tool”. In: *SIAM J. Sci. Comput.* 36.1 (2014), A267–A288. DOI: 10.1137/110860082. eprint: <http://dx.doi.org/10.1137/110860082>. URL: <http://dx.doi.org/10.1137/110860082>.
- [26] Hong Xiao and Zydrunas Gimbutas. “A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions”. In: *Comput. Math. Appl.* 59.2 (2010), pp. 663–676. ISSN: 0898-1221. DOI: 10.1016/j.camwa.2009.10.027. URL: <http://dx.doi.org/10.1016/j.camwa.2009.10.027>.

Chapter 5

Appendix

In this section we provide tables of points and entries of the weight matrix for select bilinear quadrature rules, computed to double precision. Source codes for computing these values (written in Fortran 2003) can be found at the [cawong89](#) GitHub account.

5.1 Bilinear quadratures on the triangle

In this section, the *reference triangle* refers to the triangle with vertices $(-1, 1)$, $(-1, -1)$, $(1, -1)$.

Table 5.1: Coordinates of the 3-point bilinear quadrature on the reference triangle

x	y
+2.9475992120325478e-01	-6.4726245320687859e-01
-6.4749743873001875e-01	+2.9475992292897008e-01
-6.4726248246399398e-01	-6.4749746973133360e-01

Table 5.2: Coordinates of the 6-point bilinear quadrature on the reference triangle

x	y
-7.6385006163765512e-01	-1.2040186182237422e-01
-7.9064472323785773e-01	-7.9112218664354761e-01
-7.9112239648291971e-01	+5.8176742102601953e-01
+5.8176730675887967e-01	-7.9064491625694788e-01
-1.2040227075039041e-01	-1.1574799359145108e-01
-1.1574766444091067e-01	-7.6385010552052024e-01

Table 5.3: Coordinates of the 10-point bilinear quadrature on the reference triangle

x	y
-3.8235440373639296e-01	-8.4891768366580356e-01
+2.2999240003121374e-01	-3.8042073897376882e-01
-8.5597016842768514e-01	-8.5757939818067108e-01

+2.3127223014561840e-01	-8.4891793725327869e-01
-8.4920871679319387e-01	+2.3101067471104852e-01
+7.1354978240775757e-01	-8.5757963973920537e-01
-8.4957165987014416e-01	-3.8042083006432192e-01
-8.5640128949708239e-01	+7.1280224166227979e-01
-3.3279216315763921e-01	-3.3441547955618250e-01
-3.8180145054273024e-01	+2.3100984065084240e-01

Table 5.4: Coordinates of the 15-point bilinear quadrature on the reference triangle

x	y
-8.9685323147221696e-01	-5.3068672184260833e-02
+3.9829935262579569e-01	-4.9845801019006913e-01
+3.9783826502924763e-01	-9.0031475458763799e-01
-4.9809401116545987e-01	+3.9820025708336504e-01
-8.9984121191959499e-01	-4.9845839577018969e-01
+7.9203093691075377e-01	-8.9412991027103328e-01
-9.0010586245029323e-01	+3.9820094872069339e-01
-5.0077331923900514e-02	-5.3068723866631140e-02
+2.2897509356805679e-02	-5.1099587701369831e-01
-8.9790205264859624e-01	-8.9412986960279162e-01
-5.1619419311807149e-02	-8.9676235358117817e-01
-4.9752353267552379e-01	-9.0031481111135714e-01
-5.1108770490345223e-01	+2.2175634490085173e-02
-8.9617296305369876e-01	+7.9234503447688287e-01
-5.1190172367381492e-01	-5.1099654964736330e-01

Table 5.5: Coordinates of the 21-point bilinear quadrature on the reference triangle

x	y
-5.7748191202487031e-01	+5.0048837230108767e-01
-6.9797843018782713e-01	-8.9408398054284011e-01
-9.2610057775740140e-01	-1.5110518899882375e-01
+1.5688797186581363e-01	-6.5210930162064784e-01
-9.0457611912865710e-01	+1.7596453431448431e-01
-9.0270974491005374e-01	-9.3338111378822786e-01
+5.9206240872012905e-01	-8.9408397909121362e-01
-5.0477866815169592e-01	-6.5210930317548166e-01
-6.2381910520645167e-01	+2.4763820714977297e-01
-5.8111594397608789e-02	-3.0505788662007655e-01
+8.3609085727776722e-01	-9.3338111085108721e-01
-2.7138839772059808e-01	+1.7596452252056732e-01
-6.3683052111611460e-01	-3.0505788753085294e-01
-9.2300646998653335e-01	+5.0048838428067544e-01
+7.7205765901862011e-02	-1.5110519088286647e-01
+5.4672349804261899e-01	-6.3164530792828644e-01
-2.9564926969895122e-01	-4.0870145173203126e-01
-9.1507818644613148e-01	-6.3164531260533474e-01
+2.3590406816402681e-01	-9.2605653162753621e-01
-9.2386459031343193e-01	+8.4772918005677744e-01
-3.0984754129813807e-01	-9.2605653099968965e-01

Table 5.6: Coordinates of the 28-point bilinear quadrature on the reference triangle

x	y
-9.4786800506156588e-01	-9.2797168970730581e-01
-4.0714258494019867e-01	+3.2934464210555431e-01
+1.5246393344057665e-01	-7.1124486525734998e-01
-9.2674307863019800e-01	-7.6750892905403179e-01
-9.4367853970915383e-01	+8.8734945107288821e-01
+2.3677284544142310e-01	-9.3604071415994217e-01
-1.7200527532836050e-02	-4.3624769617765458e-01
-7.0362931681597907e-01	-6.1660417993456162e-01
-4.4122914975762406e-01	-7.1123842424818695e-01
-2.4028475913393982e-01	-3.7877977992330600e-02
-5.4653912702304575e-01	-4.3624750057440653e-01
+6.5021530013931061e-01	-9.3723522750423682e-01
-9.3387916441266916e-01	-4.6646580196396574e-03
-4.4539669661265274e-01	-1.0919694705526113e-01
+6.9426151653210166e-01	-7.6752522351218588e-01
-7.1298565945519499e-01	-9.3723268560279027e-01
+3.3827583816320916e-01	-4.0077751450612503e-01
-9.4467196789371244e-01	+6.1874527202339669e-01
-9.2220463873534142e-01	+3.2935216737510797e-01
-9.3749199701042762e-01	-4.0077995366467739e-01
-7.2183239765563501e-01	-3.7881625574916325e-02
-6.1449850452367581e-02	-4.6726533658036718e-03
+3.2023402371470905e-01	-6.1660375798156253e-01
-4.5666179608208880e-02	-9.0866131793989391e-01
-3.0073152892688532e-01	-9.3604169426405792e-01
-7.1013843060771409e-01	+4.2027433445601825e-01
-6.7408012667761175e-01	+6.1874693980567397e-01
+8.7583652906494824e-01	-9.2798118234868798e-01

Table 5.7: Coordinates of the 36-point bilinear quadrature on the reference triangle

x	y
-7.7237059769943139e-01	-3.8278899655718024e-01
+4.5380145368677677e-01	-5.1573575441167574e-01
-6.1661695181291987e-01	-1.9086929855188761e-01
-5.3714737150720737e-01	-9.3946026843342845e-01
-9.5267794239006331e-01	+9.0542029385926048e-01
-3.8650025978775471e-01	-7.6809970335599642e-01
-3.8562535095131906e-01	+1.5892457760974768e-01
+1.5456205481682861e-01	-7.6810269591835534e-01
-7.3600861660641470e-01	+6.9326251165379360e-01
-5.2549896945967611e-01	+4.6424913104803583e-01
-9.3804670410766799e-01	-5.1576161023917955e-01
-7.6543318884896605e-01	+5.3087049488120153e-01
-2.2107168256377613e-01	+1.6661302260405911e-01
-9.5735662424241486e-01	+6.9336374851122873e-01
-2.2549130285301264e-01	-9.4703799834402158e-01
-9.4568051895182859e-01	-2.2374622893552509e-01
+1.6943058087463878e-01	-2.2372834546444079e-01
+9.1720135482826065e-02	-5.4615453774336098e-01

-1.8624791570782390e-01	-6.2755875841716724e-01
+4.7663027918886391e-01	-9.3944316878365797e-01
-7.2647325383647754e-01	-9.5747118179388180e-01
-9.5913616998126405e-01	-7.2470614115921428e-01
-1.9248751623005655e-01	-1.9083835022522816e-01
+9.0514183148629113e-01	-9.6808215054407620e-01
-5.4561346384264797e-01	-5.4613562522044434e-01
-9.3699384753616699e-01	-9.6814700979090829e-01
-7.7025890375382311e-01	-7.6839648258249493e-01
+6.8397828012341411e-01	-9.5745193471913737e-01
+1.5514751755837558e-01	-3.8280670321893184e-01
-9.4553945886408952e-01	+1.6665016936073290e-01
+6.8387004245041827e-01	-7.2470688197953181e-01
-7.7329059800203481e-01	+1.5889909859737891e-01
+1.7252537498677939e-01	-9.4700999276038311e-01
+5.3865042868059543e-01	-7.6841176569749825e-01
-9.3877378745791540e-01	+4.6425217304782185e-01
-5.4842891571497865e-01	+9.6958382530335369e-02

Table 5.8: Coordinates of the 45-point bilinear quadrature on the reference triangle

x	y
-9.6043525595086865e-01	+9.1979639899349563e-01
-8.1667491155978866e-01	+7.8503533906373169e-01
-6.0711247031531757e-01	+5.5842351671092305e-01
-3.4208134883117686e-01	+2.9851128059553167e-01
+8.1107244702225814e-02	-1.2617015044401581e-01
+2.3337093971936032e-01	-2.8433183930241429e-01
+5.3577969086897814e-01	-5.6991907470096148e-01
+7.5668982040175148e-01	-7.8589736331796411e-01
+9.1897643490268710e-01	-9.5746493426731449e-01
-9.6778328201821573e-01	+7.8452047825112636e-01
-8.0892892557269225e-01	+6.1664975813915102e-01
-5.0579178306698125e-01	+3.3977698143225649e-01
-3.1280488759434499e-01	-9.2266888892046945e-03
-8.8340640433914770e-03	-1.7123673955457119e-01
+2.7790393280015818e-01	-7.3478500763657595e-01
+6.0891167076987995e-01	-8.2546253076852150e-01
+7.6908554190507150e-01	-9.7055989797217845e-01
-9.5131575998554374e-01	+5.5927317332365256e-01
-8.3326864790937427e-01	+3.3962057612537161e-01
-6.1124735691818211e-01	+2.2282081312286703e-01
+2.7800574498561222e-02	-6.3123996804935634e-01
-3.6715091965185305e-02	-3.2710611433342196e-01
+4.1930074355921171e-01	-6.0032659310880976e-01
+5.1593214067522630e-01	-9.5963466043644186e-01
-9.5643019511403871e-01	+2.9821812092087685e-01
-6.7851893262684071e-01	-9.7414360424940630e-03
-6.3587332728570456e-01	-3.2629054160862214e-01
-3.1483407752352466e-01	-3.6928758458050165e-01
+1.5971610049847892e-01	-7.9643511061381411e-01
+2.3117926205675440e-01	-9.6044965006813310e-01
-9.5509123406033059e-01	-1.2647391630751670e-01

-8.1959175096370640e-01	-1.6997891685868585e-01
-3.9713206863104528e-01	-6.3208553668182965e-01
-3.6341850137129461e-01	-7.9720592207953145e-01
-2.5388200233483461e-02	-9.4763683911006569e-01
-9.4923094926223128e-01	-2.8439604637716098e-01
-8.2002480572206204e-01	-5.9960959268875924e-01
-5.4301696443973513e-01	-7.3540751254658654e-01
-2.7047322424994907e-01	-9.6014077655970453e-01
-9.6512738913123075e-01	-5.6921327588739024e-01
-7.8186707877175154e-01	-8.2524520693381209e-01
-5.5708511783860803e-01	-9.5991800715590903e-01
-9.7077554582147541e-01	-7.8546619223681113e-01
-7.9840506067592409e-01	-9.7084532996779027e-01
-9.6232095380184213e-01	-9.5699626196426468e-01

Table 5.9: Weight matrix for the 15-point bilinear quadrature on the reference triangle

Column 1	Column 2	Column 3	Column 4
+1.7314840719851010e-01	-1.5473514756742693e-03	-1.5582974218723931e-03	-3.1298752284857775e-03
-1.5473514756742693e-03	+1.4829550999500887e-01	+4.5078487288155965e-03	+2.8234443616926650e-02
-1.5582974218723931e-03	+4.5078487288155965e-03	+1.4749535537480057e-01	-5.4746383610184996e-04
-3.1298752284857775e-03	+2.8234443616926650e-02	-5.4746383610184996e-04	+1.4754094380022262e-01
-4.3048397478703426e-02	-2.1320516175396230e-03	-4.9507534578380327e-04	-5.4324597951871628e-04
+9.4042053558521059e-04	-6.8496161870620267e-03	-4.3949401841515386e-03	-6.9674032393434848e-03
-4.2229141479369930e-02	-5.4325508851372605e-04	-2.0172062394950121e-03	+4.4921146664280579e-03
+4.9752089103417552e-03	-4.3048808029663564e-02	-3.1191718714423723e-03	-4.2229518829799396e-02
-4.9799416397087763e-03	-8.8733082820452291e-03	-8.2552248657111055e-03	+3.5838067323792546e-03
+1.0732541246081070e-02	+4.6196448394392116e-04	-6.2484587989508570e-03	+4.6779893783661220e-04
+4.9750757929196343e-03	-3.1690603123949728e-03	-4.2572080514432148e-02	-1.5245532854353382e-03
-3.1190684405302030e-03	-4.9504470110993664e-04	+2.8061277210926178e-02	-2.0171639360201826e-03
+6.2119696939990060e-04	+3.6794073662468332e-03	+7.8124043474909451e-03	-8.3345352904171423e-03
+9.7546604149445996e-03	-6.6130932876606241e-03	+4.6921766011634415e-04	-5.4805689475342853e-03
+7.1347236706140074e-04	+8.0274594885260710e-03	+3.4502072764345075e-03	+7.9862017235375980e-03
Column 5	Column 6	Column 7	Column 8
-4.3048397478703426e-02	+9.4042053558521059e-04	-4.2229141479369930e-02	+4.9752089103417552e-03
-2.1320516175396230e-03	-6.8496161870620267e-03	-5.4325508851372605e-04	-4.3048808029663564e-02
-4.9507534578380327e-04	-4.3949401841515386e-03	-2.0172062394950121e-03	-3.1191718714423723e-03
-5.4324597951871628e-04	-6.9674032393434848e-03	+4.4921146664280579e-03	-4.2229518829799396e-02
+1.4829535679440883e-01	+4.6197309843886650e-04	+2.8234214766017493e-02	-1.5471698563612754e-03
+4.6197309843886650e-04	+7.0940272506991975e-02	+4.6780329947735907e-04	+1.0732621773709438e-02
+2.8234214766017493e-02	+4.6780329947735907e-04	+1.4754100536621506e-01	-3.1298463062591222e-03
-1.5471698563612754e-03	+1.0732621773709438e-02	-3.1298463062591222e-03	+1.7314789355100146e-01
+8.0274404354063085e-03	+3.6984427492137823e-03	+7.9863656170862576e-03	+7.1376021489397211e-04
-6.8495981961270583e-03	+1.4682969191083771e-03	-6.9673436466527426e-03	+9.4034651230184764e-04
-3.1689833843072009e-03	+9.0602362086193086e-03	+1.5245320172636359e-03	+4.9750123929727975e-03
+4.5077625056611451e-03	-6.2486665674383567e-03	-5.4752814076639673e-04	-1.5581520803615531e-03
+3.6793935523794200e-03	-2.6463359087690748e-03	-8.3347977032414249e-03	+6.2155465013525571e-04
-6.6132396257754626e-03	+1.6927830310856081e-03	-5.4811958053197600e-03	+9.7542781091326941e-03
-8.8734831046793795e-03	-2.5746345579008459e-03	+3.5837601544965953e-03	-4.9803366911511642e-03
Column 9	Column 10	Column 11	Column 12
-4.9799416397087763e-03	+1.0732541246081070e-02	+4.9750757929196343e-03	-3.1190684405302030e-03
-8.8733082820452291e-03	+4.6196448394392116e-04	-3.1690603123949728e-03	-4.9504470110993664e-04
-8.2552248657111055e-03	-6.2484587989508570e-03	-4.2572080514432148e-02	+2.8061277210926178e-02

+3.5838067323792546e-03	+4.6779893783661220e-04	-1.5245532854353382e-03	-2.0171639360201826e-03
+8.0274404354063085e-03	-6.8495981961270583e-03	-3.1689833843072009e-03	+4.5077625056611451e-03
+3.6984427492137823e-03	+1.4682969191083771e-03	+9.0602362086193086e-03	-6.2486665674383567e-03
+7.9863656170862576e-03	-6.9673436466527426e-03	-1.5245320172636359e-03	-5.4752814076639673e-04
+7.1376021489397211e-04	+9.4034651230184764e-04	+4.9750123929727975e-03	-1.5581520803615531e-03
+2.7274280775242876e-01	-2.5746022562111674e-03	+8.0079024112659212e-04	+3.4501790189666276e-03
-2.5746022562111674e-03	+7.0939565271663799e-02	+9.0598136277898998e-03	-4.3943190848463355e-03
+8.0079024112659212e-04	+9.0598136277898998e-03	+1.7300394394938470e-01	-4.2572325504869793e-02
+3.4501790189666276e-03	-4.3943190848463355e-03	-4.2572325504869793e-02	+1.4749538574862822e-01
-1.2851880955015216e-02	-2.6463266009664138e-03	-4.8302385889239692e-03	+7.8123433774444292e-03
-2.6251264448137208e-03	+1.6928103370707724e-03	+9.1820273973518209e-04	+4.6923359569973964e-04
-1.2006293243989934e-02	+3.6984616802569811e-03	+8.0074644546664834e-04	-8.2553288781737847e-03
Column 13	Column 14	Column 15	
+6.2119696939990060e-04	+9.7546604149445996e-03	+7.1347236706140074e-04	
+3.6794073662468332e-03	-6.6130932876606241e-03	+8.0274594885260710e-03	
+7.8124043474909451e-03	+4.6921766011634415e-04	+3.4502072764345075e-03	
-8.3345352904171423e-03	-5.4805689475342853e-03	+7.9862017235375980e-03	
+3.6793935523794200e-03	-6.6132396257754626e-03	-8.8734831046793795e-03	
-2.6463359087690748e-03	+1.6927830310856081e-03	-2.5746345579008459e-03	
-8.3347977032414249e-03	-5.4811958053197600e-03	+3.5837601544965953e-03	
+6.2155465013525571e-04	+9.7542781091326941e-03	-4.9803366911511642e-03	
-1.2851880955015216e-02	-2.6251264448137208e-03	-1.2006293243989934e-02	
-2.6463266009664138e-03	+1.6928103370707724e-03	+3.6984616802569811e-03	
-4.8302385889239692e-03	+9.1820273973518209e-04	+8.0074644546664834e-04	
+7.8123433774444292e-03	+4.6923359569973964e-04	-8.2553288781737847e-03	
-2.7291126828122414e-01	+3.6358576696919457e-03	-1.2851408731457268e-02	
+3.6358576696919457e-03	+7.0700694881114481e-02	-2.6251242185308600e-03	
-1.2851408731457268e-02	-2.6251242185308600e-03	+2.7274264299903106e-01	

Table 5.10: Weight matrix for the 28-point bilinear quadrature on the reference triangle

Column 1	Column 2	Column 3	Column 4
+2.7578108111253389e-02	+1.2189090003462560e-03	+4.2608099432383211e-04	-1.2088726324330982e-02
+1.2189090003462560e-03	+9.9068533813387441e-02	+1.1332401565486720e-05	-3.1232328011715597e-03
+4.2608099432383211e-04	+1.1332401565486720e-05	+1.8591500503218103e-01	-1.0444989633838163e-04
-1.2088726324330982e-02	-3.1232328011715597e-03	-1.0444989633838163e-04	+5.6930971537903496e-02
+3.3970467042719561e-04	+2.1550176070055586e-03	-3.2612436421614743e-04	-5.1735541555930327e-04
+7.5135938919125057e-04	+5.1234704040420864e-04	-1.7792150441973523e-02	+4.6030392506798560e-03
-2.1198319501122835e-03	+3.7386709328863497e-03	-5.6214741831206103e-02	+8.8202445019561638e-04
+1.3399134121217251e-03	+3.6293453863553539e-03	-9.4954922611718683e-03	-7.8145134277359772e-03
-5.0219438652087832e-03	+1.2156818884104309e-03	+1.5446615097434896e-02	+1.6017418814621331e-02
+1.0830281300436703e-05	-1.3155717083007695e-02	+1.0054003802129004e-02	+6.7331298111587585e-04
+1.9360118016455426e-04	-6.6874102491959091e-03	+2.3265167002322833e-02	-7.9206247537582481e-04
-2.6362878394590360e-04	+2.2881164950339819e-03	+1.3885055940760812e-03	-1.4028302081511412e-03
-5.0547057377795940e-03	-3.9841454237018363e-03	+2.5490019851030759e-03	+2.8733112940335288e-03
-1.2721969513759754e-03	+2.1664281703764870e-02	+1.5140173293347400e-02	+4.3478354933652328e-03
+4.5697559177956218e-04	-3.9360776151315504e-03	+1.6016441277526564e-02	-5.8312136438493350e-04
+1.6237535181526136e-03	-2.3637808700689445e-04	-2.1093305685189914e-03	-3.8913956458129484e-03
+6.5511873943400480e-04	+1.3652664125799817e-02	+4.4699533711241796e-03	-1.0996567755392084e-03
-2.2498175365009234e-03	-2.8357451606117151e-03	+5.2116923076925411e-04	+3.8816288818211055e-03
+4.3426833351178015e-03	+8.1132891074484504e-03	+1.2143124795273399e-03	-3.9396773476290748e-03
+5.1963115856037620e-03	+2.5937876322222270e-03	+2.0560513909241279e-03	+1.4429471099068902e-03
+2.7462446959654603e-03	-3.5177319654263782e-03	-1.0156231894151418e-02	-8.7254610426178558e-03

-1.1150008943409057e-04	-2.4240229731561168e-02	-6.3056513563078233e-03	+1.3469220785489441e-04
-1.0000885940160162e-03	-1.6342342782367299e-02	-4.7639026595685648e-02	+2.6460109617397610e-03
-3.5908459426406446e-04	+3.8061333411611429e-03	-1.1029297294524851e-02	-8.7037801242963331e-03
-7.9617151046552177e-04	-7.8551271746690233e-04	+7.9059785951979406e-03	+3.7414858824861104e-03
-1.1915496881594817e-03	-6.7290195567423711e-03	-1.1677018738359246e-03	+1.8372132901049263e-03
-8.9202917253479989e-05	-2.9556785861392754e-02	+1.0756759604372702e-03	+5.8526337954289335e-04
-4.0875495755872041e-05	+4.3410924269056501e-03	-5.0204798748365916e-03	+4.5693011578868862e-04
Column 5	Column 6	Column 7	Column 8
+3.3970467042719561e-04	+7.5135938919125057e-04	-2.1198319501122835e-03	+1.33991341212127251e-03
+2.1550176070055586e-03	+5.1234704040420864e-04	+3.7386709328863497e-03	+3.6293453863553539e-03
-3.2612436421614743e-04	-1.7792150441973523e-02	-5.6214741831206103e-02	-9.4954922611718683e-03
-5.1735541555930327e-04	+4.6030392506798560e-03	+8.8202445019561638e-04	-7.8145134277359772e-03
+2.1064376042377785e-02	-1.5065538645192428e-04	+5.1614361615686993e-04	-1.3388892041948772e-03
+1.5065538645192428e-04	+1.1089323481696914e-01	+5.5717288750043161e-03	-5.4994148414757383e-03
+5.1614361615686993e-04	+5.5717288750043161e-03	+2.1516607496957324e-01	-1.5514550241601364e-02
-1.3388892041948772e-03	-5.4994148414757383e-03	-1.5514550241601364e-02	+1.6687323352890476e-01
-3.2648536678226255e-04	+7.9054411306782955e-03	+2.3266970445396391e-02	-4.7647785011431819e-02
-5.5349260170097121e-04	-5.5999106349904925e-04	-2.9198789182297073e-02	-1.0866062394542691e-02
+5.1671972365268369e-04	-1.9769490696797624e-02	+2.0920122269522147e-02	-4.7259257317819561e-02
+3.1337602024699617e-04	-4.1033955273879494e-03	+2.2157376123502891e-03	+1.6839182491994620e-03
-1.9923309428261211e-03	-1.5757525927789485e-03	-1.2176258784445358e-02	+1.4771343608492539e-02
+1.7751769121762682e-03	-5.4539824824209621e-03	-2.5333967898782599e-02	+7.2166229409807107e-03
-5.1681377393799483e-04	+3.7425295626600701e-03	-7.8733260939605094e-04	+2.6465053138225608e-03
+3.1351565756690224e-04	-9.3778288850261213e-03	+6.2420017316622773e-03	+6.0626365889634611e-03
+1.1456752173759615e-03	+3.1777990688408170e-03	+1.2273710229922002e-02	-4.0273708117700589e-04
+7.6786792282342140e-04	+9.0811558285750318e-04	+5.3949511890072387e-04	-7.769794557552701e-03
+2.1580998365854477e-03	-7.8545805953266845e-04	-6.6872802391815204e-03	-1.6343590849122843e-02
+1.1464617111831782e-03	+3.0280116900032201e-03	+7.6747854788147504e-04	-1.8933436347747513e-02
-5.5411503290900537e-04	+6.3429083662466003e-03	+2.5359701166147830e-02	+2.3531878578830656e-02
-1.9907917184942624e-03	-4.8939965266633066e-04	+6.8723282230390348e-03	+4.6519824833307181e-03
-1.3388450385362193e-03	+2.3589084887936724e-04	-4.7270562108486312e-02	+5.8727779877887918e-04
+1.3426370693467838e-03	-7.7588230215530268e-02	+3.1680294681871807e-02	+8.1280961505820638e-03
-1.5063282884769061e-04	+3.9473024602858967e-02	-1.9768872247157850e-02	+2.3661820379152637e-04
+9.6936790581991002e-04	-1.3335674837075999e-03	+6.2489228729137021e-04	-4.4737979393178057e-03
+7.7077140113945050e-04	+4.7079179626933543e-04	-1.7769173294606563e-03	+7.2303050043401405e-04
+3.3947804455467033e-04	-7.9141909075068943e-04	+1.9251769686691553e-04	-1.0011572200731103e-03
Column 9	Column 10	Column 11	Column 12
-5.0219438652087832e-03	+1.0830281300436703e-05	+1.9360118016455426e-04	-2.6362878394590360e-04
+1.2156818884104309e-03	-1.3155717083007695e-02	-6.6874102491959091e-03	+2.2881164950339819e-03
+1.5446615097434896e-02	+1.0054003802129004e-02	+2.3265167002322833e-02	+1.3885055940760812e-03
+1.6017418814621331e-02	+6.7331298111587585e-04	-7.9206247537582481e-04	-1.4028302081511412e-03
-3.2648536678226255e-04	-5.5349260170097121e-04	+5.1671972365268369e-04	+3.1337602024699617e-04
+7.9054411306782955e-03	-5.5999106349904925e-04	-1.9769490696797624e-02	-4.1033955273879494e-03
+2.3266970445396391e-02	-2.9198789182297073e-02	+2.0920122269522147e-02	+2.2157376123502891e-03
-4.7647785011431819e-02	-1.0866062394542691e-02	-4.7259257317819561e-02	+1.6839182491994620e-03
+1.8592597145400633e-01	-1.0158352728281755e-02	-5.6223387443134119e-02	-2.1100769563984034e-03
-1.0158352728281755e-02	+1.8764377823961126e-01	+2.5361471433597130e-02	-3.8554028422247231e-03
-5.6223387443134119e-02	+2.5361471433597130e-02	+2.1516490914793199e-01	+6.2427614806852046e-03
-2.1100769563984034e-03	-3.8554028422247231e-03	+6.2427614806852046e-03	+5.2879346853375531e-02
-6.3057088243058555e-03	-4.4377064327032869e-03	+6.8723332794520097e-03	-7.1013868686791289e-05
+1.5142330291537895e-02	-7.2825295795245187e-02	-2.5336152395517033e-02	+1.6026824394762306e-03
-1.0374683146656160e-04	-8.7258151966791101e-03	+8.8033205620127492e-04	-3.8908928602809840e-03
+1.3881149714164039e-03	-1.0206645605165480e-03	+2.2167089052018848e-03	+2.6975600108272929e-03
+2.0579545911177158e-03	+1.122636276208288e-02	+7.6860296959120980e-04	-3.5092356448932686e-04

+1.0759192268402608e-03	+2.1781088976186971e-04	-1.7769611584289465e-03	-4.5373914191862723e-04
+1.1993699851011436e-05	-3.5182151032975245e-03	+3.7368983084950700e-03	-2.3650363397358754e-04
+4.4730760349496659e-03	+4.4545976137271692e-03	+1.2273036566816957e-02	-9.7242927402815642e-04
+1.0055844273724917e-02	+2.6723499160260997e-02	-2.9197282134496030e-02	-1.0212833892929579e-03
+2.5489679513823363e-03	-3.0271794047340664e-02	-1.2177916385341606e-02	-9.0731571334624959e-04
-9.4973890686204706e-03	+2.3533992826739725e-02	-1.5513686044932869e-02	+6.0643308948611840e-03
-1.1028382910976355e-02	-1.1715686941448603e-02	+3.1680128592864315e-02	+1.5404149965457045e-02
-1.7790971272262864e-02	+6.3436203663348604e-03	+5.5724824191627879e-03	-9.3783071753730917e-03
-1.1686031360187091e-03	+7.8386189861750933e-03	+6.2523852599089696e-04	+1.0689248517750721e-03
+5.2175006371765254e-04	+3.3146242597728161e-03	+5.3725228672955665e-04	-1.5103039456872169e-03
+4.2775803380074164e-04	+2.7478416328565471e-03	-2.1201601292624699e-03	+1.6192294575884465e-03
Column 13	Column 14	Column 15	Column 16
-5.0547057377795940e-03	-1.2721969513759754e-03	+4.5697559177956218e-04	+1.6237535181526136e-03
-3.9841454237018363e-03	+2.1664281703764870e-02	-3.9360776151315504e-03	-2.3637808700689445e-04
+2.5490019851030759e-03	+1.5140173293347400e-02	+1.6016441277526564e-02	-2.1093305685189914e-03
+2.8733112940335288e-03	+4.3478354933652328e-03	-5.8312136438493350e-04	-3.8913956458129484e-03
-1.9923309428261211e-03	+1.7751769121762682e-03	-5.1681377393799483e-04	+3.1351565756690224e-04
-1.5757525927789485e-03	-5.4539824824209621e-03	+3.7425295626600701e-03	-9.3778288850261213e-03
-1.2176258784445358e-02	-2.5333967898782599e-02	-7.8733260939605094e-04	+6.2420017316622773e-03
+1.4771343608492539e-02	+7.2166229409807107e-03	+2.6465053138225608e-03	+6.0626365889634611e-03
-6.3057088243058555e-03	+1.5142330291537895e-02	-1.0374683146656160e-04	+1.3881149714164039e-03
-4.4377064327032869e-03	-7.2825295795245187e-02	-8.7258151966791101e-03	-1.0206645605165480e-03
+6.8723332794520097e-03	-2.5336152395517033e-02	+8.8033205620127492e-04	+2.2167089052018848e-03
-7.1013868686791289e-05	+1.6026824394762306e-03	-3.8908928602809840e-03	+2.6975600108272929e-03
+9.0489130029407477e-02	+1.7509858878709886e-02	+1.3432545675811980e-04	-9.0932916864862583e-04
+1.7509858878709886e-02	+2.1941070810566402e-01	+4.3457816977721647e-03	+1.6020200912494158e-03
+1.3432545675811980e-04	+4.3457816977721647e-03	+5.6931358875070344e-02	-1.4030281120998933e-03
-9.0932916864862583e-04	+1.6020200912494158e-03	-1.4030281120998933e-03	+5.2880383937390626e-02
-1.7450272820222921e-03	-8.6573370582547510e-03	+1.4485194302293505e-03	-9.7184811750515793e-04
+1.8260066423985533e-02	-8.0178848647190805e-03	+5.8614185722962256e-04	-1.5109418870899341e-03
-2.4243970170675536e-02	+2.1663708780308499e-02	-3.1233782151539762e-03	+2.2897606233060783e-03
-1.3462829261395427e-02	-8.6546092579064728e-03	-1.0997717101408075e-03	-3.4853134092069940e-04
-3.0267263362940701e-02	-7.2821182540636686e-02	+6.7533741652275468e-04	-3.8551581971681794e-03
+2.1977533946374280e-03	+1.7513397993017363e-02	+2.8692684702810722e-03	-7.1218574210089248e-05
+4.6516903759111191e-03	+7.2192389847409210e-03	-7.8183127978842405e-03	+1.6832790572332847e-03
+1.5215454454745208e-03	+2.8355248530772740e-03	-8.7051820029380531e-03	+1.5401894144168950e-02
-4.8821979978840306e-04	-5.4546669184612172e-03	+4.6033711792367602e-03	-4.1012410878727137e-03
+2.4565883462770153e-03	+4.9213729365323190e-03	+1.8364514547890222e-03	+1.0686099629055119e-03
+1.3192659685276214e-03	-8.0157853247957555e-03	+3.8803174994664650e-03	-4.5375951253616691e-04
-1.1171171912215167e-04	-1.2731066575300063e-03	-1.2086491667815322e-02	-2.6411407049255339e-04
Column 17	Column 18	Column 19	Column 20
+6.5511873943400480e-04	-2.2498175365009234e-03	+4.3426833351178015e-03	+5.1963115856037620e-03
+1.3652664125799817e-02	-2.8357451606117151e-03	+8.1132891074484504e-03	+2.593787632222270e-03
+4.4699533711241796e-03	+5.2116923076925411e-04	+1.2143124795273399e-03	+2.0560513909241279e-03
-1.0996567755392084e-03	+3.8816288818211055e-03	-3.9396773476290748e-03	+1.4429471099068902e-03
+1.1456752173759615e-03	+7.6786792282342140e-04	+2.1580998365854477e-03	+1.1464617111831782e-03
+3.1777990688408170e-03	+9.0811558285750318e-04	-7.8545805953266845e-04	+3.0280116900032201e-03
+1.2273710229922002e-02	+5.3949511890072387e-04	-6.6872802391815204e-03	+7.6747854788147504e-04
-4.0273708117700589e-04	+6.7769794557552701e-03	-1.6343590849122843e-02	-1.8933436347747513e-02
+2.0579545911177158e-03	+1.0759192268402608e-03	+1.1993699851011436e-05	+4.4730760349496659e-03
+1.1222636276208288e-02	+2.1781088976186971e-04	-3.5182151032975245e-03	+4.4545976137271692e-03
+7.6860296959120980e-04	-1.7769611584289465e-03	+3.7368983084950700e-03	+1.2273036566816957e-02
-3.5092356448932686e-04	-4.5373914191862723e-04	-2.3650363397358754e-04	-9.7242927402815642e-04
-1.7450272820222921e-03	+1.8260066423985533e-02	-2.4243970170675536e-02	-1.3462829261395427e-02

-8.6573370582547510e-03	-8.0178848647190805e-03	+2.1663708780308499e-02	-8.6546092579064728e-03
+1.4485194302293505e-03	+5.8614185722962256e-04	-3.1233782151539762e-03	-1.0997717101408075e-03
-9.7184811750515793e-04	-1.5109418870899341e-03	+2.2897606233060783e-03	-3.4853134092069940e-04
+7.1720962658457302e-02	-3.0915422832229376e-04	+2.5935890552019037e-03	+1.3350479204900371e-03
-3.0915422832229376e-04	+6.1080263401282163e-02	-2.9561072638645047e-02	-9.2045934354439748e-03
+2.5935890552019037e-03	-2.9561072638645047e-02	+9.9070587355209086e-02	+1.3654877852922591e-02
+1.3350479204900371e-03	-9.2045934354439748e-03	+1.3654877852922591e-02	+7.1727695838429847e-02
+4.4558273821584601e-03	+3.3155035089640441e-03	-1.3151854350546943e-02	+1.1220602873023872e-02
-1.3463211425769129e-02	+1.3210644645904089e-03	-3.9837851403479035e-03	-1.7444589027715040e-03
-1.8928462958090319e-02	+7.2231965698779503e-04	+3.6304300343973739e-03	-4.0078909546427074e-04
-5.8561702602715629e-03	-4.4421583805941281e-03	+3.8069614492952648e-03	-5.8560485313790666e-03
+3.0272205908358661e-03	+4.7106325553801574e-04	+5.1124776450855124e-04	+3.1764359435937075e-03
-3.0869616416602515e-03	+5.9606335463268666e-05	-6.7303737896572419e-03	-3.0877670397543317e-03
-9.2032416550986238e-03	+2.0013023345901086e-03	-2.8336956690468685e-03	-3.0813886691625147e-04
+5.1940819560416388e-03	-8.9376115251838949e-05	+1.2189190157949523e-03	+6.5553863647545383e-04
Column 21	Column 22	Column 23	Column 24
+2.7462446959654603e-03	-1.1150008943409057e-04	-1.0000885940160162e-03	-3.5908459426406446e-04
-3.5177319654263782e-03	-2.4240229731561168e-02	-1.6342342782367299e-02	+3.8061333411611429e-03
-1.0156231894151418e-02	-6.3056513563078233e-03	-4.7639026595685648e-02	-1.1029297294524851e-02
-8.7254610426178558e-03	+1.3469220785489441e-04	+2.6460109617397610e-03	-8.7037801242963331e-03
-5.5411503290900537e-04	-1.9907917184942624e-03	-1.3388450385362193e-03	+1.3426370693467838e-03
+6.3429083662466003e-03	-4.8939965266633066e-04	+2.3589084887936724e-04	-7.7588230215530268e-02
+2.5359701166147830e-02	+6.8723282230390348e-03	-4.7270562108486312e-02	+3.1680294681871807e-02
+2.3531878578830656e-02	+4.6519824833307181e-03	+5.8727779877887918e-04	+8.1280961505820638e-03
+1.0055844273724917e-02	+2.5489679513823363e-03	-9.4973890686204706e-03	-1.1028382910976355e-02
+2.6723499160260997e-02	-3.0271794047340664e-02	+2.3533992826739725e-02	-1.1715686941448603e-02
-2.9197282134496030e-02	-1.2177916385341606e-02	-1.5513686044932869e-02	+3.1680128592864315e-02
-1.0212833892929579e-03	-9.0731571334624959e-04	+6.0643308948611840e-03	+1.5404149965457045e-02
-3.0267263362940701e-02	+2.1977533946374280e-03	+4.6516903759111191e-03	+1.5215454454745208e-03
-7.2821182540636686e-02	+1.7513397993017363e-02	+7.2192389847409210e-03	+2.8355248530772740e-03
+6.7533741652275468e-04	+2.8692684702810722e-03	-7.8183127978842405e-03	-8.7051820029380531e-03
-3.8551581971681794e-03	-7.1218574210089248e-05	+1.6832790572332847e-03	+1.5401894144168950e-02
+4.4558273821584601e-03	-1.3463211425769129e-02	-1.8928462958090319e-02	-5.8561702602715629e-03
+3.3155035089640441e-03	+1.3210644645904089e-03	+7.2231965698779503e-04	-4.4421583805941281e-03
-1.3151854350546943e-02	-3.9837851403479035e-03	+3.6304300343973739e-03	+3.8069614492952648e-03
+1.1220602873023872e-02	-1.7444589027715040e-03	-4.0078909546427074e-04	-5.8560485313790666e-03
+1.8763949229377111e-01	-4.4384748899201319e-03	-1.0867758726257094e-02	-1.1715375503362107e-02
-4.4384748899201319e-03	+9.0490522284564612e-02	+1.4770366680297404e-02	+1.5230319673185493e-03
-1.0867758726257094e-02	+1.4770366680297404e-02	+1.6687209950977414e-01	+8.1273128215416952e-03
-1.1715375503362107e-02	+1.5230319673185493e-03	+8.1273128215416952e-03	+1.8346089279093944e-01
-5.5991787017199300e-04	-1.5763938961853843e-03	-5.4985119505743399e-03	-7.7585646982112427e-02
+7.8381612667565940e-03	+2.4549565554986001e-03	-4.4745967872273284e-03	+6.9475386242994399e-03
+2.1849147743545214e-04	+1.8258486705486091e-02	+6.7761653587042848e-03	-4.4417658967491709e-03
+1.1473688334353320e-05	-5.0534356089028346e-03	+1.3394592012297584e-03	-3.6518022932886818e-04
Column 25	Column 26	Column 27	Column 28
-7.9617151046552177e-04	-1.1915496881594817e-03	-8.9202917253479989e-05	-4.0875495755872041e-05
-7.8551271746690233e-04	-6.7290195567423711e-03	-2.9556785861392754e-02	+4.3410924269056501e-03
+7.9059785951979406e-03	-1.1677018738359246e-03	+1.0756759604372702e-03	-5.0204798748365916e-03
+3.7414858824861104e-03	+1.8372132901049263e-03	+5.8526337954289335e-04	+4.5693011578868862e-04
-1.5063282884769061e-04	+9.6936790581991002e-04	+7.7077140113945050e-04	+3.3947804455467033e-04
+3.9473024602858967e-02	-1.3335674837075999e-03	+4.7079179626933543e-04	-7.9141909075068943e-04
-1.9768872247157850e-02	+6.2489228729137021e-04	-1.7769173294606563e-03	+1.9251769686691553e-04
+2.3661820379152637e-04	-4.4737979393178057e-03	+7.2303050043401405e-04	-1.0011572200731103e-03
-1.7790971272262864e-02	-1.1686031360187091e-03	+5.2175006371765254e-04	+4.2775803380074164e-04

+6.3436203663348604e-03	+7.8386189861750933e-03	+3.3146242597728161e-03	+2.7478416328565471e-03
+5.5724824191627879e-03	+6.2523852599089696e-04	+5.3725228672955665e-04	-2.1201601292624699e-03
-9.3783071753730917e-03	+1.0689248517750721e-03	-1.5103039456872169e-03	+1.6192294575884465e-03
-4.8821979978840306e-04	+2.4565883462770153e-03	+1.3192659685276214e-03	-1.1171171912215167e-04
-5.4546669184612172e-03	+4.9213729365323190e-03	-8.0157853247957555e-03	-1.2731066575300063e-03
+4.6033711792367602e-03	+1.8364514547890222e-03	+3.8803174994664650e-03	-1.2086491667815322e-02
-4.1012410878727137e-03	+1.0686099629055119e-03	-4.5375951253616691e-04	-2.6411407049255339e-04
+3.0272205908358661e-03	-3.0869616416602515e-03	-9.2032416550986238e-03	+5.1940819560416388e-03
+4.7106325553801574e-04	+5.9606335463268666e-05	+2.0013023345901086e-03	-8.9376115251838949e-05
+5.1124776450855124e-04	-6.7303737896572419e-03	-2.8336956690468685e-03	+1.2189190157949523e-03
+3.1764359435937075e-03	-3.0877670397543317e-03	-3.0813886691625147e-04	+6.5553863647545383e-04
-5.5991787017199300e-04	+7.8381612667565940e-03	+2.1849147743545214e-04	+1.1473688334353320e-05
-1.5763938961853843e-03	+2.4549565554986001e-03	+1.8258486705486091e-02	-5.0534356089028346e-03
-5.4985119505743399e-03	-4.4745967872273284e-03	+6.7761653587042848e-03	+1.3394592012297584e-03
-7.7585646982112427e-02	+6.9475386242994399e-03	-4.4417658967491709e-03	-3.6518022932886818e-04
+1.1088977614545878e-01	-1.3333310423014107e-03	+9.0785053455405107e-04	+7.5351960916686118e-04
-1.3333310423014107e-03	+1.0992402273789459e-01	+5.5962798894973088e-05	-1.1916246115777080e-03
+9.0785053455405107e-04	+5.5962798894973088e-05	+6.1081676902560680e-02	-2.2491651888216736e-03
+7.5351960916686118e-04	-1.1916246115777080e-03	-2.2491651888216736e-03	+2.7579526739210745e-02

5.2 Bilinear quadratures on the square

Table 5.11: Coordinates of the 3-point bilinear quadrature on $[-1, 1]^2$

x	y
-2.4803243723465021e-01	-5.9796034989446800e-01
-4.0109988408604075e-01	+4.0119797073707941e-01
+5.9774538304268865e-01	+2.4806649013783522e-01

Table 5.12: Coordinates of the 6-point bilinear quadrature on $[-1, 1]^2$

x	y
+1.9362968103654435e-01	+1.6289142454415681e-01
+6.5840783235372691e-01	+8.7995086520853971e-02
+2.6071992986599457e-01	+6.8646052253093115e-01
+3.9141902400612705e-01	-6.8656458158788303e-01
-7.1224040091623397e-01	+5.1139613133676276e-01
-6.7956043103432617e-01	-6.0772490035836779e-01

Table 5.13: Coordinates of the 10-point bilinear quadrature on $[-1, 1]^2$

x	y
-7.3646461785710782e-01	-7.6824876442298107e-01
+4.8814102094439910e-01	-1.8604734870142037e-01
+8.1504072025126684e-01	+4.6081371693389944e-01
-2.3492420324981320e-01	-1.9145064767847084e-01
-6.5631548965136155e-01	+7.9623356838907611e-01
+2.0590697858403903e-01	-8.3773867400987534e-01

-1.2362992940930762e-01	+3.9201161274391572e-01
+8.0867547242867499e-01	-5.9724811973450032e-01
-8.5414063246829253e-01	+9.7067254015967802e-02
+3.7951216307386409e-01	+8.2518032883456061e-01

Table 5.14: Coordinates of the 15-point bilinear quadrature on $[-1, 1]^2$

x	y
+8.8333852320353312e-01	-6.6565627173429232e-01
+7.9165662303815598e-01	+8.9126995576285373e-01
-8.9646560177706613e-01	-7.5082152089281562e-01
-7.8131937165837018e-01	+8.2943387774980337e-01
+5.0217326004941754e-01	+6.2489803313285319e-01
-1.8879672669672917e-01	-1.6222398351434997e-01
+9.1655677307203332e-01	+3.2410921058422371e-01
-3.6155557623174836e-02	-4.2330769996468193e-01
-9.2025596683061306e-01	+2.2676270204088014e-01
+6.7386780436691363e-01	-2.0253888084742230e-01
-4.5777395511184926e-01	+4.7903393824433238e-01
-4.3308728612410236e-01	-9.0857778225659958e-01
-6.7721981848682788e-01	-3.1734443530730200e-01
-3.7092485696762628e-02	+8.8406519231107084e-01
+5.0592293638334507e-01	-8.9014862151497598e-01

Table 5.15: Coordinates of the 21-point bilinear quadrature on $[-1, 1]^2$

x	y
+9.4227394157715594e-01	+5.8165113141750113e-01
+8.0736208741237814e-01	+9.1774352340772425e-01
+9.0842259015941718e-01	-8.7155500290648014e-01
+2.6615550334588078e-01	-2.9332213113286776e-01
-7.6505311353385541e-01	+9.2117128745044619e-01
-5.0541749075890485e-01	-9.4394253852777299e-01
+4.3064784675224316e-01	-9.4657252861264640e-01
+4.5603359547721857e-02	+9.5025044635031553e-01
-4.0238842736911618e-01	+7.4595247393789665e-01
-7.2094883217612926e-01	-5.9871481744196897e-01
+6.6383029473792843e-01	-6.1707204822386374e-01
+5.7008204592928560e-02	+4.5445462235324646e-01
-6.7186856468946399e-01	+2.3876343640591510e-01
+9.3106492559145981e-01	-2.5348024394092572e-01
-3.3217576147906042e-01	-2.2629805366228117e-01
-9.3723416926135850e-01	-1.7971117937661563e-01
-3.6509074470560871e-02	-7.6078556272299980e-01
-9.3037993912696193e-01	-8.6328534576403770e-01
+7.0849938759992026e-01	+2.0524740571838546e-01
-9.3115703939414474e-01	+6.3455635411794364e-01
+4.7160312498783902e-01	+7.3339372612282339e-01

Table 5.16: Coordinates of the 28-point bilinear quadrature on $[-1, 1]^2$

x	y
+3.9456060055363124e-01	-7.7787426646692281e-01
+3.7133096641423036e-01	+9.5367067725901167e-01
-7.0445860166326124e-01	-9.6365528345461626e-01
-7.3135592299956516e-06	-1.1236372863861321e-01
+9.5602839975068299e-01	+5.9324271029700659e-01
-7.8400004789501332e-01	-5.5182341907192223e-01
+9.4398412463724202e-01	-8.4715958391295809e-01
-1.7763040043034239e-06	-4.4394239512829442e-01
-3.9455580210491420e-01	-7.7788273035090227e-01
-3.7133766152269726e-01	+9.5367090280864331e-01
+7.1974667481011334e-01	+7.9995321439977463e-01
+9.2520097426686254e-01	+9.6134266394483769e-01
-7.1974467775006579e-01	+7.9995331913713397e-01
-9.6618908821496907e-01	-2.1878442912217189e-01
-9.2520473289336136e-01	+9.6135037588270200e-01
-8.2133127981090881e-01	+2.0049246872336202e-01
+9.6618626480836645e-01	-2.1879210626697063e-01
+5.4741609600437424e-01	-1.8812179531368672e-01
+8.2134094733167984e-01	+2.0048012279147406e-01
-5.4741068856843711e-01	-1.8811457273567517e-01
+4.2391530537631067e-01	+4.8044062467493948e-01
-9.5602935157035951e-01	+5.9326006498766692e-01
-6.4579782743485223e-06	-9.5128914373344298e-01
+7.8400320691098424e-01	-5.5181962872070944e-01
+7.0445697840475108e-01	-9.6365778269391300e-01
-2.6910708755941710e-06	+7.5936772099998928e-01
-9.4398576042891014e-01	-8.4716256880679153e-01
-4.2391000780133092e-01	+4.8043918817755255e-01

Table 5.17: Coordinates of the 36-point bilinear quadrature on $[-1, 1]^2$

x	y
+8.8708256808445940e-01	+9.5498237577554601e-01
+6.8125610831729477e-01	-9.6553591847870790e-01
+8.1063940920682542e-01	-1.8305788020112687e-01
+6.6548191386350164e-01	+8.3241090583337829e-01
+9.6536356586087335e-01	+1.6587513419418262e-01
+2.5788624839514024e-01	-2.5009189074396121e-01
-9.6477828367434393e-01	+1.7422931346504314e-01
-6.5314487101537810e-01	+8.2969889226496407e-01
+4.7962450694679429e-03	+8.5919867451497711e-01
-8.1881004373159083e-01	-7.4787331816402647e-01
-8.1345315305263011e-01	-1.7010656908035651e-01
-5.5972154395533713e-01	+1.6580785277087276e-01
+9.4945593140232332e-01	-9.1802941219731793e-01
-5.5611305247545806e-01	-4.9945130078760913e-01
-3.5302429045297434e-01	+6.3996580965368000e-01
-4.1096664688772032e-03	-9.7672945018769641e-01
+5.7066677597643056e-01	+1.7575993450658495e-01
-5.8280799087355364e-03	-6.3502820640868551e-01

-6.7301694638713632e-04	+3.3982521260536580e-01
+8.2097870038877396e-01	-7.5764227808171980e-01
+3.5181784071292810e-01	-8.5195006737622014e-01
-9.6046242432041462e-01	-9.2705459843577487e-01
-8.8317286889020608e-01	+9.5461279747332417e-01
-9.6878557501932472e-01	+7.6722774462874233e-01
-3.7005847221123256e-01	+9.7298888299224640e-01
+9.6738842949187498e-01	-4.9169201815731689e-01
-8.2883230936696650e-01	+5.0345104978031274e-01
+3.7989628459400487e-01	+9.7337330806431743e-01
+8.3208057521733148e-01	+4.9915783945156511e-01
-3.6109131721749899e-01	-8.5081461758966503e-01
+5.6471187753736429e-01	-5.1232003243066271e-01
-2.5870940179817586e-01	-2.4432374286370465e-01
-9.6746814204499421e-01	-4.8297594750543466e-01
+3.6050127598152165e-01	+6.3437759545824846e-01
-6.8989590660354116e-01	-9.6646303451678395e-01
+9.6773659470479656e-01	+7.5780543587019167e-01

Table 5.18: Coordinates of the 45-point bilinear quadrature on $[-1, 1]^2$

x	y
+1.1683700817996905e-02	+3.9177365327395119e-01
+3.0425994044120097e-01	+9.7445316104931701e-01
-9.4233919212948347e-01	-9.6505415782614445e-01
-6.9159469515814209e-01	+4.5888228631546396e-01
+8.1541194968580344e-01	+9.7599910777029075e-01
+8.7451393050709325e-01	-4.7420706352575376e-01
+5.1770921191597666e-03	+8.5399285806678171e-01
-3.3265895758969366e-01	-3.7141923773015795e-01
-8.8299592144098615e-01	+1.6891575003989778e-01
-4.3210341579808825e-01	+1.6396195931506816e-01
+3.1791035415999086e-01	+6.5078984050132183e-01
+9.7030547511183751e-01	-7.4428542471942427e-01
+9.7601654425940654e-01	+4.9437415279723018e-01
+3.1601921536795224e-01	-3.6494237388445966e-01
-3.0670455780361827e-01	+9.7430690137788922e-01
+6.8818209442177103e-01	+4.6446136116130315e-01
-8.0144632456638021e-01	-8.6202252358600218e-01
-9.6761497700332888e-01	+9.0669281802543633e-01
+8.8248843932153187e-01	+1.6279908317088193e-01
-8.1359486544310855e-01	+9.7820162034676472e-01
+4.4192149758228133e-01	+1.5363620692068924e-01
+9.6268446621470849e-01	+8.9778475593408602e-01
-5.9009663568846005e-01	+8.6687376767094293e-01
-9.7013224083361360e-01	-7.3704677449857003e-01
-8.5280256159500434e-01	+7.1620013796917947e-01
+3.1239306220220403e-01	-8.5608707517776306e-01
-5.8343056363311130e-01	-9.7650443034770895e-01
-9.7706623651713176e-01	+4.9541929408774360e-01
-5.9047656244888336e-01	-6.4853908404003124e-01
-3.1441913996125198e-01	+6.4708178517281645e-01
+8.0484141013686239e-01	-8.6502951186703281e-01

+6.9302521529658390e-01	-1.5302403277469082e-01
+9.4652157671374670e-01	-9.6751536083100831e-01
-3.1447247300587267e-01	-8.5780780675891910e-01
-8.7115488475974023e-01	-4.7023249153457108e-01
-1.3832918646163215e-03	-6.6117516291088685e-01
+5.8059161447888208e-01	-9.7567672110078596e-01
+4.5439304171163734e-03	-9.7242975469465764e-01
+5.8752549844482926e-01	+8.6706139490421297e-01
-2.8002946253877087e-02	+3.0926681468413375e-02
-9.8109250574924556e-01	-1.7838528733511325e-01
+9.8144574173519872e-01	-1.6984110107916187e-01
+8.4735416878456560e-01	+7.1525506778922265e-01
-6.9560332834584770e-01	-1.5810487353892611e-01
+5.9944517649370688e-01	-6.5667637856389327e-01

Table 5.19: Weight matrix for the 15-point bilinear quadrature on $[-1, 1]^2$

Column 1	Column 2	Column 3	Column 4
+2.2503774223868975e-01	-7.2566555190170783e-03	-4.1266245501584350e-03	+1.6782566824635810e-02
-7.2566555190170783e-03	+1.5797681823700574e-01	+2.8371612263303758e-03	-6.0108539543197136e-03
-4.1266245501584350e-03	+2.8371612263303758e-03	+1.7589212092296491e-01	+2.3965271449690752e-03
+1.6782566824635810e-02	-6.0108539543197136e-03	+2.3965271449690752e-03	+2.4009208079875943e-01
+1.5359862012282717e-02	-5.9374783281753232e-02	+1.2568137767892671e-02	+1.1202417592958855e-02
+1.1205080827111555e-01	-6.1497137498490347e-02	+1.2896927633499394e-01	+1.2069520748055682e-01
+1.7088817565026677e-02	+9.6386996694840070e-03	+1.0194017673745433e-03	+1.6180662269742498e-03
-1.0385020373178899e-01	+2.6348216680570229e-02	-7.5136940556851245e-02	-8.6319669571834928e-02
+5.0983181030424983e-03	-5.9954063872428602e-03	+1.4633219047293550e-02	+8.7520809139574676e-03
-7.4695268881646057e-02	-1.3980379477847062e-03	+1.3010912335661694e-02	-2.7545894697892936e-02
-5.2201133794095141e-02	+1.3570121968087785e-02	-5.4582706114208509e-03	-8.9587501273896308e-02
+4.7729889243611889e-03	+1.1990637847570651e-03	+2.8377096609345272e-04	+1.3765692497203746e-03
-6.4265223723825177e-03	+7.9375924843380265e-03	-6.9140229310243981e-02	-2.3268755424705818e-02
-8.1690117245786700e-03	+7.6461660408274033e-03	-5.0438129426908125e-03	+8.4650111280500772e-03
+5.8353158105131859e-03	-5.5222709057087399e-03	+7.4193236449113780e-03	+7.8089374599462401e-03
Column 5	Column 6	Column 7	Column 8
+1.5359862012282717e-02	+1.1205080827111555e-01	+1.7088817565026677e-02	-1.0385020373178899e-01
-5.9374783281753232e-02	-6.1497137498490347e-02	+9.6386996694840070e-03	+2.6348216680570229e-02
+1.2568137767892671e-02	+1.2896927633499394e-01	+1.0194017673745433e-03	-7.5136940556851245e-02
+1.1202417592958855e-02	+1.2069520748055682e-01	+1.6180662269742498e-03	-8.6319669571834928e-02
+4.4947520352010339e-01	+2.6908306086281181e-01	-2.5982734185669262e-02	-9.8336994552217200e-02
+2.6908306086281181e-01	+1.9063002944134930e+00	-4.7627584064748015e-03	-1.2822132245501625e+00
-2.5982734185669262e-02	-4.7627584064748015e-03	+2.0909927832299291e-01	+2.7399111761456270e-02
-9.8336994552217200e-02	-1.2822132245501625e+00	+2.7399111761456270e-02	+1.4998146349862287e+00
+2.4029105917997434e-02	+4.6762309497214535e-02	+4.4881690998403575e-03	+1.4420253735965850e-02
-9.6510535927637579e-03	-4.0845414065727129e-02	-3.5027279105869516e-02	+4.7009627024788551e-02
-3.3363911750169145e-03	-2.8947202916559900e-01	+2.0641512864757912e-03	+2.5906214268489952e-01
-1.1943841139135953e-02	+3.5536620318186215e-05	-5.5923957293814710e-03	+3.1310320361200286e-03
-8.9251000932479715e-02	-3.7784640849672957e-01	-3.2564390240799801e-03	+1.6591840390189111e-01
-2.8122947613111540e-02	+2.8078644619398897e-02	+1.6443322469146787e-03	-4.0814068726742182e-02
+1.4086915888183449e-02	+1.3415304618564886e-01	+4.7624565365058501e-03	-5.6369708034323729e-02
Column 9	Column 10	Column 11	Column 12
+5.0983181030424983e-03	-7.4695268881646057e-02	-5.2201133794095141e-02	+4.7729889243611889e-03
-5.9954063872428602e-03	-1.3980379477847062e-03	+1.3570121968087785e-02	+1.1990637847570651e-03
+1.4633219047293550e-02	+1.3010912335661694e-02	-5.4582706114208509e-03	+2.8377096609345272e-04

+8.7520809139574676e-03	-2.7545894697892936e-02	-8.9587501273896308e-02	+1.3765692497203746e-03
+2.4029105917997434e-02	-9.6510535927637579e-03	-3.3363911750169145e-02	-1.1943841139135953e-02
+4.6762309497214535e-02	-4.0845414065727129e-02	-2.8947202916559900e-01	+3.5536620318186215e-05
+4.4881690998403575e-03	-3.5027279105869516e-02	+2.0641512864757912e-03	-5.5923957293814710e-03
+1.4420253735965850e-02	+4.7009627024788551e-02	+2.5906214268489952e-01	+3.1310320361200286e-03
+2.1221744955356964e-01	-4.9567593168968757e-03	-3.3638715824112479e-02	-3.6983544567632793e-04
-4.9567593168968757e-03	+4.7619532903546596e-01	+4.6060667228283995e-02	+2.1876672993445881e-03
-3.3638715824112479e-02	+4.6060667228283995e-02	+5.3597429841768585e-01	-7.4546016384838508e-03
-3.6983544567632793e-04	+2.1876672993445881e-03	-7.4546016384838508e-03	+2.0350895793552837e-01
-4.0091433364902743e-02	-2.3812764969601129e-02	+3.4694055555053373e-02	-8.4093678315338167e-03
-5.3037921676367243e-04	+2.6493080522985945e-02	-2.9625419016900745e-02	+5.4836865103632115e-03
+6.0549115272901053e-05	-2.5215514871189098e-02	-3.5349953352878730e-02	+6.1661005861473753e-03
Column 13	Column 14	Column 15	
-6.4265223723825177e-03	-8.1690117245786700e-03	+5.8353158105131859e-03	
+7.9375924843380265e-03	+7.6461660408274033e-03	-5.5222709057087399e-03	
-6.9140229310243981e-02	-5.0438129426908125e-03	+7.4193236449113780e-03	
-2.3268755424705818e-02	+8.4650111280500772e-03	+7.8089374599462401e-03	
-8.9251000932479715e-02	-2.8122947613111540e-02	+1.4086915888183449e-02	
-3.7784640849672957e-01	+2.8078644619398897e-02	+1.3415304618564886e-01	
-3.2564390240799801e-03	+1.6443322469146787e-03	+4.7624565365058501e-03	
+1.6591840390189111e-01	-4.0814068726742182e-02	-5.6369708034323729e-02	
-4.0091433364902743e-02	-5.3037921676367243e-04	+6.0549115272901053e-05	
-2.3812764969601129e-02	+2.6493080522985945e-02	-2.5215514871189098e-02	
+3.4694055555053373e-02	-2.9625419016900745e-02	-3.5349953352878730e-02	
-8.4093678315338167e-03	+5.4836865103632115e-03	+6.1661005861473753e-03	
+5.2084639171955860e-01	+2.0630867977143742e-02	-2.2726746961323274e-02	
+2.0630867977143742e-02	+2.7296643746521371e-01	+9.6997371805306720e-03	
-2.2726746961323274e-02	+9.6997371805306720e-03	+2.3274179345497420e-01	

Table 5.20: Weight matrix for the 21-point bilinear quadrature on $[-1, 1]^2$

Column 1	Column 2	Column 3	Column 4
+1.0428677805625614e-01	+8.1735110525941058e-04	-3.9937516312635020e-04	-3.1235697919607158e-03
+8.1735110525941058e-04	+1.0134142833592957e-01	-8.9626107869664493e-05	+1.3420070139632355e-02
-3.9937516312635020e-04	-8.9626107869664493e-05	+8.0072418382530264e-02	+1.0100183393839781e-02
-3.1235697919607158e-03	+1.3420070139632355e-02	+1.0100183393839781e-02	+4.2386106206805546e-01
+3.0412905804133816e-04	+1.7386193974571408e-03	-4.3523281536733439e-03	-1.9616140205739055e-02
+1.6447488068908305e-03	+5.7519446680864687e-04	+3.8352505466188262e-05	+1.1165192476551919e-02
-1.4181568254800442e-03	+2.7592469846975881e-03	+1.8361101011635395e-03	+2.6502393756915109e-03
-1.1000269660140988e-03	+5.0264466407620098e-03	-1.0010784843144139e-04	-1.3993879281430739e-03
+3.4612380999905789e-04	-5.8233901149865744e-03	+6.2644814477400487e-03	+1.0837995122182734e-02
+2.9639767108169050e-03	+5.8591046016828819e-03	+2.2417053602709985e-03	+1.7313917788657093e-02
+8.6610182536656284e-04	-3.8915583678078439e-03	-8.3519072602030158e-03	-2.6055393291949218e-02
-5.6771023728759695e-03	+3.1317878742268573e-02	-4.2069201681225379e-03	+8.2745899052749858e-03
+1.7579049844597696e-03	-9.7630251001447831e-04	+5.5073925982121966e-03	+5.0318752224924261e-02
+3.2493124956227129e-03	+6.5103686807648430e-04	+2.7094806756096518e-03	-1.9468051803170120e-03
-8.4062661373352186e-03	-5.9303229451181938e-03	-6.3952796095407020e-03	-8.9014608718835278e-02
-1.1161955010617075e-03	+7.8811861051327598e-04	+1.1468897324417813e-04	-5.3908058456910123e-03
-1.1567993252649751e-03	-5.5833063472688514e-03	-1.8609874107353573e-03	-2.0494449237684346e-02
-7.1601928988652215e-04	-2.7774516543124694e-03	-8.1741570860489204e-04	-5.2221707091473302e-03
-9.1309534290861458e-03	-3.9821129818619324e-03	-1.6240960982915318e-04	+3.0051867376724710e-02
+3.6229881176918144e-04	+2.6129208168366774e-04	-2.0103209190624051e-03	-1.8942885042661103e-02
-5.8815820329493826e-03	-2.6014892857167494e-02	+1.8462669369761508e-03	-1.3655448504823108e-02

Column 5	Column 6	Column 7	Column 8
+3.0412905804133816e-04	+1.6447488068908305e-03	-1.4181568254800442e-03	-1.1000269660140988e-03
+1.7386193974571408e-03	+5.7519446680864687e-04	+2.7592469846975881e-03	+5.0264466407620098e-03
-4.3523281536733439e-03	+3.8352505466188262e-05	+1.8361101011635395e-03	-1.0010784843144139e-04
-1.9616140205739055e-02	+1.1165192476551919e-02	+2.6502393756915109e-03	-1.3993879281430739e-03
+1.1086008548998363e-01	+2.2313049820838668e-03	-2.6392965145548877e-04	+6.3799106372646455e-03
+2.2313049820838668e-03	+1.0967372509469020e-01	+4.2249052962319254e-03	-8.6596691153166712e-04
-2.6392965145548877e-04	+4.2249052962319254e-03	+1.1243965034130493e-01	-9.8339710913082916e-04
+6.3799106372646455e-03	-8.6596691153166712e-04	-9.8339710913082916e-04	+1.1973918150507552e-01
-3.0763126763290419e-02	-2.6932999821430506e-03	-5.9710541021555565e-04	-6.9206313639364175e-03
-9.1533180485108157e-03	-9.3453356687278246e-03	-2.1350166661749338e-03	-2.5209135013040751e-03
+1.3549767434015205e-02	-1.3081950122066914e-03	-9.8451711777223384e-03	-2.3057900949855289e-04
+3.3680550006632004e-02	+3.6965589012664284e-04	+1.3582203865723961e-03	-4.6151991759277320e-03
-1.9624133149996628e-02	+2.5165869126146023e-03	-8.2378294294361328e-03	+4.7135193126041693e-03
-1.90432807495669721e-03	-1.1699482538389530e-03	-6.8431098731518451e-04	-7.5325212267468491e-04
+2.3720286616973396e-02	+4.1162470318043546e-04	+5.5472646871195519e-03	+3.4127789165530431e-03
+2.4202244139289062e-03	+4.7041061521584278e-04	-6.6535676539509938e-04	-3.6381710474365711e-04
-3.9678767260957620e-03	-7.1117454399198098e-03	-8.8371281351254504e-03	-1.9722765705640008e-04
+1.3398572936354990e-03	+2.0956805320406931e-03	+2.2751696893985701e-04	+5.5178911020032609e-04
-1.3135809729647291e-03	-2.0113358316685621e-03	+6.1355736992399958e-03	+6.2963435882512310e-03
+3.3520492133808496e-03	-4.3814185018853770e-04	+1.9035561259625059e-03	+8.3478915960086839e-04
-9.1409659787286095e-03	-1.4634026536475549e-03	-3.4362921206744185e-03	-4.7285526492311276e-03
Column 9	Column 10	Column 11	Column 12
+3.4612380999905789e-04	+2.9639767108169050e-03	+8.6610182536656284e-04	-5.6771023728759695e-03
-5.8233901149865744e-03	+5.8591046016828819e-03	-3.8915583678078439e-03	+3.1317878742268573e-02
+6.2644814477400487e-03	+2.2417053602709985e-03	-8.3519072602030158e-03	-4.2069201681225379e-03
+1.0837995122182734e-02	+1.7313917788657093e-02	-2.6055393291949218e-02	+8.2745899052749858e-03
-3.0763126763290419e-02	-9.1533180485108157e-03	+1.3549767434015205e-02	+3.3680550006632004e-02
-2.6932999821430506e-03	-9.3453356687278246e-03	-1.3081950122066914e-03	+3.6965589012664284e-04
-5.9710541021555565e-04	-2.1350166661749338e-03	-9.8451711777223384e-03	+1.3582203865723961e-03
-6.9206313639364175e-03	-2.5209135013040751e-03	-2.3057900949855289e-04	-4.6151991759277320e-03
+2.8258442978151499e-01	+1.0915670179968561e-02	-1.1214183960023527e-02	-5.1067921037138218e-02
+1.0915670179968561e-02	+2.3055561605996910e-01	-6.6435375267499576e-03	-2.6611460048015109e-02
-1.1214183960023527e-02	-6.6435375267499576e-03	+2.4315248240953899e-01	+5.1283861501354531e-03
-5.1067921037138218e-02	-2.6611460048015109e-02	+5.1283861501354531e-03	+4.4891622947192783e-01
+1.7826874622000846e-02	-3.8205431713892275e-03	-1.8775375093060181e-02	-7.8940362038350739e-03
-7.8179589279703636e-03	+1.2140205861595774e-03	-5.2813993167365875e-03	+3.2069717204267038e-03
-3.0582287032970334e-02	-8.6632019002984770e-03	+2.6587958190941216e-02	+6.8322082751560251e-02
-5.8866673287558732e-04	-4.4645720141377595e-03	+2.5877945970498857e-03	+3.9851054631707796e-03
+1.1870304286905487e-02	+5.6208478531819522e-03	+1.6105226847191716e-03	-1.4119056682730967e-02
-1.0963234539339574e-03	-8.1954623126893663e-03	+1.4572630304951911e-03	+4.2141221549425872e-03
+4.5550790827137699e-03	-2.8165990110738865e-03	-8.1452397273625229e-03	+1.8812542723697024e-02
-1.3387560153689249e-02	-1.5447833181820745e-03	+8.1377691140239374e-03	+4.0745534256812314e-03
+4.6453202987622993e-03	+2.9349572876425003e-03	+6.9501126482742165e-04	-6.8969091332209159e-02
Column 13	Column 14	Column 15	Column 16
+1.7579049844597696e-03	+3.2493124956227129e-03	-8.4062661373352186e-03	-1.1161955010617075e-03
-9.7630251001447831e-04	+6.5103686807648430e-04	-5.9303229451181938e-03	+7.8811861051327598e-04
+5.5073925982121966e-03	+2.7094806756096518e-03	-6.3952796095407020e-03	+1.1468897324417813e-04
+5.0318752224924261e-02	-1.9468051803170120e-03	-8.9014608718835278e-02	-5.3908058456910123e-03
-1.9624133149996628e-02	+1.90432807495669721e-03	-2.3720286616973396e-02	+2.4202244139289062e-03
+2.5165869126146023e-03	-1.1699482538389530e-03	+4.1162470318043546e-04	+4.7041061521584278e-04
-8.2378294294361328e-03	-6.8431098731518451e-04	+5.5472646871195519e-03	-6.6535676539509938e-04
+4.7135193126041693e-03	-7.5325212267468491e-04	+3.4127789165530431e-03	-3.6381710474365711e-04
+1.7826874622000846e-02	-7.8179589279703636e-03	-3.0582287032970334e-02	-5.8866673287558732e-04

-3.8205431713892275e-03	+1.2140205861595774e-03	-8.6632019002984770e-03	-4.4645720141377595e-03
-1.8775375093060181e-02	-5.2813993167365875e-03	+2.6587958190941216e-02	+2.5877945970498857e-03
-7.8940362038350739e-03	+3.2069717204267038e-03	+6.8322082751560251e-02	+3.9851054631707796e-03
+3.2780479264523205e-01	+5.3154208382045704e-03	-3.9056096689397690e-02	-1.9949230687025510e-03
+5.3154208382045704e-03	+1.4015232507639622e-01	-9.3577306499390903e-03	-2.1513812705550432e-03
-3.9056096689397690e-02	-9.3577306499390903e-03	+4.2623292425170206e-01	-3.5743078145701059e-03
-1.9949230687025510e-03	-2.1513812705550432e-03	-3.5743078145701059e-03	+1.3377064203716987e-01
+3.7946228833065161e-03	+7.5331086246365996e-03	+2.3065260140822147e-03	+5.0191681632136007e-03
-5.5831489637585307e-04	+7.7157702081318631e-04	+6.9299312438204251e-03	+1.9392227172016819e-03
-4.6274356362006139e-03	-7.0822339900064165e-03	+1.8379024967733647e-02	+3.5583343264079912e-03
-3.2786211435375761e-03	-2.6254055977638002e-03	+1.3491706886964588e-02	+3.6868907789579606e-03
+9.1908898650429737e-03	+1.6028326497713559e-03	-1.6865625628206222e-02	-6.0169541496489431e-03
Column 17	Column 18	Column 19	Column 20
-1.1567993252649751e-03	-7.1601928988652215e-04	-9.1309534290861458e-03	+3.6229881176918144e-04
-5.5833063472688514e-03	-2.7774516543124694e-03	-3.9821129818619324e-03	+2.6129208168366774e-04
-1.8609874107353573e-03	-8.1741570860489204e-04	-1.6240960982915318e-04	-2.0103209190624051e-03
-2.0494449237684346e-02	-5.2221707091473302e-03	+3.0051867376724710e-02	-1.8942885042661103e-02
-3.9678767260957620e-03	+1.3398572936354990e-03	-1.3135809729647291e-03	+3.3520492133808496e-03
-7.1117454399198098e-03	+2.0956805320406931e-03	-2.0113358316685621e-03	-4.3814185018853770e-04
-8.8371281351254504e-03	+2.2751696893985701e-04	+6.1355736992399958e-03	+1.9035561259625059e-03
-1.9722765705640008e-04	+5.5178911020032609e-04	+6.2963435882512310e-03	+8.3478915960086839e-04
+1.1870304286905487e-02	-1.0963234539339574e-03	+4.5550790827137699e-03	-1.3387560153689249e-02
+5.6208478531819522e-03	-8.1954623126893663e-03	-2.8165990110738865e-03	-1.5447833181820745e-03
+1.6105226847191716e-03	+1.4572630304951911e-03	-8.1452397273625229e-03	+8.1377691140239374e-03
-1.4119056682730967e-02	+4.2141221549425872e-03	+1.8812542723697024e-02	+4.0745534256812314e-03
+3.7946228833065161e-03	-5.5831489637585307e-04	-4.6274356362006139e-03	-3.2786211435375761e-03
+7.5331086246365996e-03	+7.7157702081318631e-04	-7.0822339900064165e-03	-2.6254055977638002e-03
+2.3065260140822147e-03	+6.9299312438204251e-03	+1.8379024967733647e-02	+1.3491706886964588e-02
+5.0191681632136007e-03	+1.9392227172016819e-03	+3.5583343264079912e-03	+3.6868907789579606e-03
+2.8833495590029634e-01	-3.0741154404783982e-03	-8.3970833380550263e-03	-1.5415978706536067e-03
-3.0741154404783982e-03	+6.8296465300533876e-02	-5.974932653396874e-04	-1.1770984484832142e-04
-8.3970833380550263e-03	-5.974932653396874e-04	+3.0593628027845260e-01	+1.6992243827301031e-03
-1.5415978706536067e-03	-1.1770984484832142e-04	+1.6992243827301031e-03	+1.1171627510426622e-01
+1.3871880477143101e-02	+3.0228545976519875e-03	-3.2125573083310248e-03	-2.7659237977446308e-03
Column 21			
-5.8815820329493826e-03			
-2.6014892857167494e-02			
+1.8462669369761508e-03			
-1.3655448504823108e-02			
-9.1409659787286095e-03			
-1.4634026536475549e-03			
-3.4362921206744185e-03			
-4.7285526492311276e-03			
+4.6453202987622993e-03			
+2.9349572876425003e-03			
+6.9501126482742165e-04			
-6.8969091332209159e-02			
+9.1908898650429737e-03			
+1.6028326497713559e-03			
-1.6865625628206222e-02			
-6.0169541496489431e-03			
+1.3871880477143101e-02			
+3.0228545976519875e-03			
-3.2125573083310248e-03			

-2.7659237977446308e-03
+2.7432544337994830e-01

5.3 Bilinear quadratures on the disk

Table 5.21: Coordinates of the 3-point bilinear quadrature on the unit disk

x	y
-1.5323207788123461e-01	-5.3040157303421887e-01
-3.8272519755320089e-01	+3.9790365865896538e-01
+5.3595727539120253e-01	+1.3249791434264588e-01

Table 5.22: Coordinates of the 6-point bilinear quadrature on the unit disk

x	y
-6.6705512920468113e-02	-6.8845317149603569e-01
-3.5069751189426057e-01	+5.9617573336874374e-01
+1.4858252938998915e-03	+1.1282501702100057e-03
+4.5862855527552032e-01	+5.1776459408821018e-01
-6.7537141076129503e-01	-1.4930815564266747e-01
+6.3414595157300435e-01	-2.7617913592716464e-01

Table 5.23: Coordinates of the 10-point bilinear quadrature on the unit disk

x	y
+1.9025606165724379e-01	-7.6593278985613267e-01
+2.2224710136406561e-01	-3.6976841907709618e-01
-4.4082380131630772e-01	-6.6625410997530363e-06
-5.0425429937664668e-01	+6.2032912235536475e-01
+2.2225197020942067e-01	+3.6976966360537894e-01
-5.0425577107546882e-01	-6.2031938378596108e-01
+1.9026068682157246e-01	+7.6593635878110566e-01
+7.0979144904329117e-01	-3.5267522094547427e-01
-8.0440582294392471e-01	+3.8424876904033961e-06
+7.0979612540301151e-01	+3.5266776321918586e-01

Table 5.24: Coordinates of the 15-point bilinear quadrature on the unit disk

x	y
+2.2850521668568057e-02	-3.4268832784783030e-09
+5.0386162644143151e-01	+1.9652095051959602e-08
+1.7397944525559805e-01	+4.9857820853038776e-01
-4.6033047427628332e-01	+3.4794353553468399e-01
-4.6033052767597854e-01	-3.4794351110912292e-01
+1.7397937471390673e-01	-4.9857800935291663e-01
+8.0407524691546794e-01	+2.8403940029479746e-01
+4.2200902320360772e-01	+7.4376436986793260e-01
-1.4141100157516312e-01	+8.3557483255348308e-01

-6.5876551617661983e-01	+5.4917996336948127e-01
-8.4302256262529973e-01	-5.9042245946769213e-08
-6.5876533043771635e-01	-5.4918009247558430e-01
-1.4141073947867205e-01	-8.3557500237990479e-01
+4.2200935803189199e-01	-7.4376422942118281e-01
+8.0407534700381222e-01	-2.8403930415341333e-01

Table 5.25: Coordinates of the 21-point bilinear quadrature on the unit disk

x	y
+1.9352329990126532e-01	+2.1497297639976878e-01
-2.4808006284862219e-01	+1.2447280303791241e-06
+1.9352462050015137e-01	-2.1497980143830644e-01
+5.8990212425842281e-01	+4.8693508224651075e-06
+4.6573025156390035e-01	+5.6921945633476956e-01
-1.9333724309824624e-01	+6.0260515483416011e-01
-5.5433978697188790e-01	+3.0415381662561852e-01
-5.5433455281042388e-01	-3.0414939139681357e-01
-1.9333932551994643e-01	-6.0260847947756269e-01
+4.6572936774765106e-01	-5.6921755259847961e-01
+8.5723879799720548e-01	+2.4510324627851177e-01
+5.8442487544578192e-01	+6.7901479389963526e-01
+1.2586613571053265e-01	+8.8301365128310894e-01
-3.7124818993706060e-01	+8.2398019289617774e-01
-7.6260455183891973e-01	+4.8716111189725980e-01
-8.9024173491039182e-01	-5.3541426320469245e-06
-7.6260572795676229e-01	-4.8715953509660814e-01
-3.7125784986056298e-01	-8.2397865634106426e-01
+1.2586213678726077e-01	-8.8301598494353728e-01
+5.8442339490826278e-01	-6.7902756221399596e-01
+8.5724436683636429e-01	-2.4509635138382413e-01

Table 5.26: Coordinates of the 28-point bilinear quadrature on the unit disk

x	y
+3.4534673484373557e-01	-5.5839656230721487e-07
-8.9779198056740783e-01	+2.1760647017682530e-01
+5.3016824533437701e-01	-7.7026978609082164e-01
+1.2888358761931537e-01	-3.3429559815988344e-06
-1.3632099377117890e-01	+6.7011458806102586e-01
+4.0427627790921317e-01	+6.1894088918254131e-01
+6.5299218130055747e-01	-2.7228924066407983e-01
+1.0530420776958886e-01	-9.1078815035966787e-01
+9.1410788896300610e-01	+1.6551952383804518e-06
+1.0529876810695317e-01	+9.1078661081346035e-01
-1.3632001172851091e-01	-6.7011444573443923e-01
-8.9778997840847463e-01	-2.1760613325077011e-01
-5.7321773412317778e-01	+4.9001870976585216e-01
-3.2151087042450976e-01	+8.6403194784049042e-01
+8.2183455187297039e-01	+4.2720847613805141e-01
-3.2151459246393932e-01	-8.6402090374422624e-01
-3.8329228985111313e-01	+2.9487329408334001e-01

-5.7321305360882202e-01	-4.9001753322415109e-01
+4.0427580546210612e-01	-6.1893906347999084e-01
-6.9356375369792345e-01	-6.1528075172136487e-01
-6.4969197735531437e-01	+6.7230796110162985e-06
-6.9357147563294219e-01	+6.1527814532564518e-01
+8.2183766003611070e-01	-4.2721100539884976e-01
+6.5299398951765453e-01	+2.7229131876058627e-01
+5.3016666614872565e-01	+7.7026811670541817e-01
+1.1554977009210694e-01	-3.3363461815187934e-01
-3.8328948174317617e-01	-2.9486837242972591e-01
+1.1555257753705599e-01	+3.3363479187124956e-01

Table 5.27: Coordinates of the 36-point bilinear quadrature on the unit disk

x	y
+3.3906906300376022e-01	-6.8772923898696359e-01
-4.5239761846689641e-01	+6.4213620341802418e-08
+4.6974364887158315e-01	+8.1977590211775708e-01
+3.3906893082128109e-01	+6.8772927976652376e-01
+4.8573182493560074e-01	+2.8579888476576537e-01
+7.6747360829223110e-01	-5.5696180478501467e-01
+6.5262119740929125e-01	+4.4850716793782069e-01
-7.0243584911759549e-01	-2.2652028205019620e-01
-7.0243586651145851e-01	+2.2652035939209109e-01
+9.3395153092823896e-02	+1.9991167175291485e-01
+9.7004212909067619e-02	+9.3238508213949500e-01
-8.6466018699184199e-01	+3.8395979905979710e-01
-3.8469569353760141e-01	-4.6376076499111224e-01
+4.8573188532450140e-01	-2.8579884406326095e-01
-6.3363743433635211e-01	+7.0326890772374440e-01
+9.1993294175802010e-01	-1.9078231984947755e-01
+9.7004188084988119e-02	-9.3238503972890796e-01
-2.6589792744088248e-01	-2.4753881266531579e-08
-5.0312419990635382e-01	+5.6484569859276901e-01
+7.4126177746051758e-01	+5.6066207241968200e-08
-2.8733367565295143e-01	-8.9790569242198714e-01
-5.0312422317270766e-01	-5.6484565007076826e-01
+6.5262119144632602e-01	-4.4850718523968641e-01
-8.6466014056802010e-01	-3.8395963345725231e-01
+9.3395125753967884e-02	-1.9991165349929357e-01
-3.8469566178415837e-01	+4.6376082121183732e-01
+1.6948807433605242e-01	+4.7404459134663968e-01
+7.6747371562333111e-01	+5.5696189037921151e-01
-9.3698602559153610e-01	-1.7256911621040512e-08
-1.2081257815216616e-01	+7.2704452790481855e-01
+4.6974376301662257e-01	-8.1977601703921721e-01
-1.2081260423868900e-01	-7.2704435829324809e-01
-6.3363734937115412e-01	-7.0326880912722800e-01
+1.6948811004145398e-01	-4.7404455921621991e-01
-2.8733370242560097e-01	+8.9790571738847913e-01
+9.1993302284426071e-01	+1.9078232460004199e-01

Table 5.28: Coordinates of the 45-point bilinear quadrature on the unit disk

x	y
-2.7378737378099488e-01	+7.3051929527712567e-01
-3.4090562281866250e-02	-1.0599860352605416e-09
-5.6140782590172067e-01	+1.7236884866026417e-01
+7.9320001167747578e-01	-3.2801644165308568e-08
+9.1348823500438991e-02	-7.9329109479131821e-01
-7.7451594197157159e-01	-1.8856648789044489e-01
-5.8925942429211020e-01	+5.2525758849702953e-01
-2.5792890506695820e-01	+5.2895323536352379e-01
+9.5598472657284328e-01	-2.2011169427966683e-07
+8.8633081859579840e-02	-9.4786032835519574e-01
+8.8960999420395481e-01	-3.4904487779288274e-01
-9.3763796909518016e-01	+1.7902664737354662e-01
+7.1056866639440430e-01	+3.7136088916458160e-01
+7.0594210495784260e-01	+6.4113497683476772e-01
+9.1348826016286480e-02	+7.9329109232968131e-01
-9.3763793458815459e-01	-1.7902664274360522e-01
-8.1315289011277481e-01	-5.0257545198573217e-01
-5.7275546078373329e-01	-7.6233589236858568e-01
+5.4795036653372309e-01	-4.5063333847054631e-09
-2.6145219791726632e-01	-9.1579613338642596e-01
+3.2237865216713946e-01	-2.4655859405228683e-01
+8.8633063586273042e-02	+9.4786034785229978e-01
+4.4665607661485446e-01	-6.4740634665577868e-01
-5.8925937121042038e-01	-5.2525765097848653e-01
+9.4863628211315096e-02	+6.3139930233111019e-01
+4.5526452928532241e-01	+3.2041889169971466e-01
-1.1169144973052872e-01	+3.4256873494343243e-01
-2.7378742921830829e-01	-7.3051932507622730e-01
+4.2395311658694612e-01	-8.5620303860285263e-01
+4.2395308101797369e-01	+8.5620295001626801e-01
-7.7451593565452859e-01	+1.8856642485115810e-01
-3.3276102792045620e-01	+2.1364615424209584e-08
-5.727555297384306e-01	+7.6233595389277220e-01
-8.1315286003180598e-01	+5.0257545551705218e-01
+8.8960994738944010e-01	+3.4904484472281722e-01
+4.4665609856204774e-01	+6.4740631251255709e-01
-5.6140781130803630e-01	-1.7236885845409677e-01
+3.2237865113962821e-01	+2.4655860484217768e-01
-1.1169145133414077e-01	-3.4256873745701161e-01
-2.6145216457672155e-01	+9.1579603525185294e-01
+9.4863625701928828e-02	-6.3139930683711321e-01
+4.5526452949865998e-01	-3.2041889214161628e-01
+7.1056863409448812e-01	-3.7136092432541995e-01
+7.0594211013903507e-01	-6.4113498476494590e-01
-2.5792889056264784e-01	-5.2895324241051689e-01

Table 5.29: Weight matrix for the 21-point bilinear quadrature on the unit disk

Column 1	Column 2	Column 3	Column 4
+3.5178649002468348e-01	-5.1462994126100954e-02	-9.3571938287814210e-02	-6.5654526945022773e-02

-5.1462994126100954e-02	+3.4775562206410388e-01	-5.1455443094206403e-02	+7.5908176527152066e-02
-9.3571938287814210e-02	-5.1455443094206403e-02	+3.5177913281590395e-01	-6.5647479991474328e-02
-6.5654526945022773e-02	+7.5908176527152066e-02	-6.5647479991474328e-02	+3.0490990438889942e-01
-5.9909261753550262e-02	+1.0886427965257496e-02	+5.5212813380646202e-02	+1.7128621804192003e-03
-3.2093839369095482e-02	+2.5639032203037506e-02	+2.6544873559448741e-02	+7.0462189815350251e-03
+4.6497345484708033e-02	-8.7230368289290705e-02	+2.2998481753211408e-02	-4.2118205413039164e-02
+2.3005414979577279e-02	-8.7236545856891223e-02	+4.6493549899043703e-02	-4.2116928194639393e-02
+2.6541266379486533e-02	+2.5643160629166519e-02	-3.2090540164146905e-02	+7.0429970906878666e-03
+5.5211199808457040e-02	+1.0882363824825447e-02	-5.9904560482515676e-02	+1.7159590370502188e-03
+3.4806040125372646e-02	-3.2314797920640473e-02	+2.1020784983566906e-02	-5.0017185473336410e-02
+4.8208891843716649e-02	-1.1316644683681818e-02	-3.1843395317634186e-02	+1.1429943324144714e-02
+3.8032682256686282e-02	-1.6915475237385827e-03	-2.5524219308606936e-02	-4.2448069559297957e-03
+2.1083426285713771e-02	+6.3377304255732587e-03	-2.2331272798461494e-02	-6.8783828838456739e-04
-7.1561420444092643e-03	+3.5217347022012062e-02	-1.7336771714607360e-02	+9.4066492788841775e-03
-1.7443358776948421e-02	+4.4047440724384893e-02	-1.7441094583753106e-02	+1.776210252320291e-02
-1.7336888072855189e-02	+3.5216967437725434e-02	-7.1552024984311404e-03	+9.4054578823279215e-03
-2.2329779307623587e-02	+6.3361228257126515e-03	+2.1081753299626436e-02	-6.8616449813446788e-04
-2.5521427554637974e-02	-1.6922734668210389e-03	+3.8029992920188771e-02	-4.2453485038532008e-03
-3.1840438590447592e-02	-1.1311995859315352e-02	+4.8200510477256570e-02	+1.1428087527974945e-02
+2.1020182108785410e-02	-3.2312214109826362e-02	+3.4801826542723770e-02	-5.0009783132674124e-02
Column 5	Column 6	Column 7	Column 8
-5.9909261753550262e-02	-3.2093839369095482e-02	+4.6497345484708033e-02	+2.3005414979577279e-02
+1.0886427965257496e-02	+2.5639032203037506e-02	-8.7230368289290705e-02	-8.7236545856891223e-02
+5.5212813380646202e-02	+2.6544873559448741e-02	+2.2998481753211408e-02	+4.6493549899043703e-02
+1.7128621804192003e-03	+7.0462189815350251e-03	-4.2118205413039164e-02	-4.2116928194639393e-02
+2.9749229050627618e-01	+3.5630706803545298e-02	-6.3787867935564879e-03	+1.1034703761356182e-03
+3.5630706803545298e-02	+2.6911306328829593e-01	-6.5384425624863698e-02	+9.7205840944118438e-03
-6.3787867935564879e-03	-6.5384425624863698e-02	+2.8395496323410241e-01	+4.0154524430235865e-02
+1.1034703761356182e-03	+9.7205840944118438e-03	+4.0154524430235865e-02	+2.8395933773472498e-01
-1.0409148359349757e-02	-3.4399997140954749e-02	+9.7193776678726491e-03	-6.5384875213700455e-02
-1.8714768922847333e-02	-1.0411884979502264e-02	+1.1072479281192169e-03	-6.3782818421823859e-03
-1.0205346947767610e-02	-7.7760834869640173e-03	+1.0524793600036938e-02	+1.7844594366834509e-02
-1.4397581581844968e-01	-4.6882901858742564e-03	-1.2711484253506162e-03	+1.7345993021658326e-03
-2.9675480022213024e-02	-2.9763920752182826e-02	+1.2737818590263829e-02	-5.2399683561085448e-03
-5.9931764131721929e-03	-5.2142011858343773e-02	+1.8036340967179453e-02	-1.0479604011896310e-02
-5.0934452353546388e-03	+2.4334832502542366e-02	-5.9348155965414623e-02	-4.1366460716491284e-03
+2.3835751068160978e-03	+8.3536728755033169e-03	-2.9927388402376076e-02	-2.9927446402655079e-02
+6.0818986957006278e-03	-4.3228590157866151e-03	-4.1353744429005624e-03	-5.9345701537517115e-02
+1.2787443484827530e-02	+9.7982579858059832e-03	-1.0479726361582290e-02	+1.8036368145606999e-02
+8.7878249722790503e-03	+1.3252000620743965e-02	-5.2389613802831716e-03	+1.2737712403867405e-02
+8.0819438421911791e-03	+3.2456866789550339e-03	+1.7316385625871563e-03	-1.2717742556467250e-03
+3.3215154347675780e-03	-4.4000320127931952e-03	+1.7842821924598455e-02	+1.0524691808853622e-02
Column 9	Column 10	Column 11	Column 12
+2.6541266379486533e-02	+5.5211199808457040e-02	+3.4806040125372646e-02	+4.8208891843716649e-02
+2.5643160629166519e-02	+1.0882363824825447e-02	-3.2314797920640473e-02	-1.1316644683681818e-02
-3.2090540164146905e-02	-5.9904560482515676e-02	+2.1020784983566906e-02	-3.1843395317634186e-02
+7.0429970906878666e-03	+1.7159590370502188e-03	-5.0017185473336410e-02	+1.1429943324144714e-02
-1.0409148359349757e-02	-1.8714768922847333e-02	-1.0205346947767610e-02	-1.4397581581844968e-01
-3.4399997140954749e-02	-1.0411884979502264e-02	-7.7760834869640173e-03	-4.6882901858742564e-03
+9.7193776678726491e-03	+1.1072479281192169e-03	+1.0524793600036938e-02	-1.2711484253506162e-03
-6.5384875213700455e-02	-6.3782818421823859e-03	+1.7844594366834509e-02	+1.7345993021658326e-03
+2.6910927770876286e-01	+3.5630009413562584e-02	-4.4006054578055046e-03	+3.2442688938478649e-03
+3.5630009413562584e-02	+2.9746058087110855e-01	+3.3234587123135034e-03	+8.0819387559338211e-03
-4.4006054578055046e-03	+3.3234587123135034e-03	+1.5272610736522654e-01	+3.5044189084596192e-03

+3.2442688938478649e-03	+8.0819387559338211e-03	+3.5044189084596192e-03	+2.0851157526510983e-01
+1.3253141214813132e-02	+8.7892051698581584e-03	+1.5513198011338476e-03	+1.0432546172406116e-02
+9.7977621090807095e-03	+1.2788043837774413e-02	+6.8820413889827633e-04	+1.4597420902803572e-03
-4.3218848811481135e-03	+6.0819500038463824e-03	-2.4269637804707925e-03	+3.3041365104688018e-03
+8.3531012113095514e-03	+2.3822365169752966e-03	-3.2175894124836157e-03	+6.6092099324850564e-04
+2.4334998100141132e-02	-5.0920854331598141e-03	-4.5225369986724986e-03	-3.9147248668796363e-03
-5.2138912147128340e-02	-5.9951386223933032e-03	+2.3752180430243303e-03	-5.6460174475074397e-03
-2.9762705984982697e-02	-2.9669156004371776e-02	-6.1074866329934344e-04	-3.0602205400174806e-03
-4.6879542719983642e-03	-1.4393955483011472e-01	-2.4349446886157071e-03	-2.8214483885151773e-03
-7.7764335389149909e-03	-1.0205568556107446e-02	+3.1110652112962237e-03	-2.4309600017835119e-03
Column 13	Column 14	Column 15	Column 16
+3.8032682256686282e-02	+2.1083426285713771e-02	-7.1561420444092643e-03	-1.7443358776948421e-02
-1.6915475237385827e-02	+6.3377304255732587e-03	+3.5217347022012062e-02	+4.4047440724384893e-02
-2.5524219308606936e-02	-2.2331272798461494e-02	-1.7336771714607360e-02	-1.7441094583753106e-02
-4.2448069559297957e-03	-6.8783828838456739e-04	+9.4066492788841775e-03	+1.7762102523230291e-02
-2.9675480022213024e-02	-5.9931764131721929e-03	-5.0934452353546388e-03	+2.3835751068160978e-03
-2.9763920752182826e-02	-5.2142011858343773e-02	+2.4334832502542366e-02	+8.3536728755033169e-03
+1.2737818590263829e-02	+1.8036340967179453e-02	-5.9348155965414623e-02	-2.9927388402376076e-02
-5.2399683561085448e-03	-1.0479604011896310e-02	-4.1366460716491284e-03	-2.9927446402655079e-02
+1.3253141214813132e-02	+9.7977621090807095e-03	-4.3218848811481135e-03	+8.3531012113095514e-03
+8.7892051698581584e-03	+1.2788043837774413e-02	+6.0819500038463824e-03	+2.3822365169752966e-03
+1.5513198011338476e-03	+6.8820413889827633e-04	-2.4269637804707925e-03	-3.2175894124836157e-03
+1.0432546172406116e-02	+1.4597420902803572e-03	+3.3041365104688018e-03	+6.6092099324850564e-04
+1.4888131214676584e-01	+3.5469690799706218e-03	-3.2149321842454303e-03	-2.1459685586229533e-03
+3.5469690799706218e-03	+1.3959588327944064e-01	-4.7723352276889241e-03	-1.526040493340231e-03
-3.2149321842454303e-03	-4.7723352276889241e-03	+1.4062957588228689e-01	+4.5254571699987270e-03
-2.1459685586229533e-03	-1.5260404933430231e-03	+4.5254571699987270e-03	+1.5048010029077155e-01
+2.2937640997742958e-03	+2.8753261018137789e-03	+7.2336902072357376e-04	+4.5216938189447907e-03
-3.6188590015140566e-03	-2.9098561598644951e-03	+2.8745487655900499e-03	-1.5230848931115882e-03
-2.6970111275473811e-03	-3.6180100009132908e-03	+2.2936133286365502e-03	-2.1474368601492040e-03
-3.0612106230388948e-03	-5.6459462694993185e-03	-3.9149048250687827e-03	+6.6339288622346555e-04
-6.1241893863240573e-04	+2.3766359898783240e-03	-4.5232018857295029e-03	-3.2175368186765385e-03
Column 17	Column 18	Column 19	Column 20
-1.7336888072855189e-02	-2.2329779307623587e-02	-2.5521427554637974e-02	-3.1840438590447592e-02
+3.5216967437725434e-02	+6.3361228257126515e-03	-1.6922734668210389e-03	-1.1311995859315352e-02
-7.1552024984311404e-03	+2.1081753299626436e-02	+3.8029992920188771e-02	+4.8200510477256570e-02
+9.4054578823279215e-03	-6.8616449813446788e-04	-4.2453485038532008e-03	+1.1428087527974945e-02
+6.0818986957006278e-03	+1.2787443484827530e-02	+8.7878249722790503e-03	+8.0819438421911791e-03
-4.3228590157866151e-03	+9.7982579858059832e-03	+1.3252000620743965e-02	+3.2456866789550339e-03
-4.1353744429005624e-03	-1.0479726361582290e-02	-5.2389613802831716e-03	+1.7316385625871563e-03
-5.9345701537517115e-02	+1.8036368145606999e-02	+1.2737712403867405e-02	-1.2717742556467250e-03
+2.4334998100141132e-02	-5.2138912147128340e-02	-2.9762705984982697e-02	-4.6879542719983642e-03
-5.0920854331598141e-03	-5.9951386223933032e-03	-2.9669156004371776e-02	-1.4393955483011472e-01
-4.5225369986724986e-03	+2.3752180430243303e-03	-6.1074866329934344e-04	-2.4349446886157071e-03
-3.9147248668796363e-03	-5.6460174475074397e-03	-3.0602205400174806e-03	-2.8214483885151773e-03
+2.2937640997742958e-03	-3.6188590015140566e-03	-2.6970111275473811e-03	-3.0612106230388948e-03
+2.8753261018137789e-03	-2.9098561598644951e-03	-3.6180100009132908e-03	-5.6459462694993185e-03
+7.2336902072357376e-04	+2.8745487655900499e-03	+2.2936133286365502e-03	-3.9149048250687827e-03
+4.5216938189447907e-03	-1.5230848931115882e-03	-2.1474368601492040e-03	+6.6339288622346555e-04
-1.4062823133784924e-01	-4.7740506450450306e-03	-3.2146984950020070e-03	+3.3025441686825728e-03
-4.7740506450450306e-03	+1.3959151594869748e-01	+3.5482418433188316e-03	+1.4608907515844002e-03
-3.2146984950020070e-03	+3.5482418433188316e-03	+1.4887811328389319e-01	+1.0427379177488250e-02
+3.3025441686825728e-03	+1.4608907515844002e-03	+1.0427379177488250e-02	+2.0847404414081527e-01
-2.4277669039013492e-03	+6.9015902960833713e-04	+1.5492614770181482e-03	+3.5100455872968681e-03

Column 21

+2.1020182108785410e-02
-3.2312214109826362e-02
+3.4801826542723770e-02
-5.0009783132674124e-02
+3.3215154347675780e-03
-4.4000320127931952e-03
+1.7842821924598455e-02
+1.0524691808853622e-02
-7.7764335389149909e-03
-1.0205568556107446e-02
+3.1110652112962237e-03
-2.4309600017835119e-03
-6.1241893863240573e-04
+2.3766359898783240e-03
-4.5232018857295029e-03
-3.2175368186765385e-03
-2.4277669039013492e-03
+6.9015902960833713e-04
+1.5492614770181482e-03
+3.5100455872968681e-03
+1.5271986969758364e-01