Theory and Computation for Bilinear Quadratures

Christopher A. Wong

University of California, Berkeley

15 March 2015

Research supported by NSF, AFOSR, and SIAM

Classical and bilinear quadratures

Classical quadrature (approximates linear functionals):

$$\int_\Omega f(x)\, d\mu = \begin{bmatrix} w_1 & \dots & w_n \end{bmatrix} \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}$$

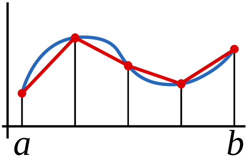


Image from Wikimedia Commons

Classical and bilinear quadratures

Classical quadrature (approximates linear functionals):

$$\int_\Omega f(x)\,d\mu = \begin{bmatrix} w_1 & \dots & w_n \end{bmatrix} \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}$$

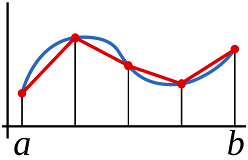


Image from Wikimedia Commons

Bilinear quadrature (approximates bilinear forms):

$$\int_\Omega f(x)g(x)\,d\mu = \begin{bmatrix} f(y_1) & \dots & f(y_m) \end{bmatrix} \begin{bmatrix} w_{11} & \dots & w_{1n} \\ \vdots & & \vdots \\ w_{m1} & \dots & w_{mn} \end{bmatrix} \begin{bmatrix} g(x_1) \\ \vdots \\ g(x_n) \end{bmatrix}.$$

Classical and bilinear quadratures

Classical quadrature (approximates linear functionals):

$$\int_\Omega f(x)\, d\mu = \begin{bmatrix} w_1 & \dots & w_n \end{bmatrix} \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}$$

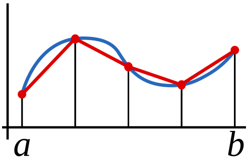


Image from Wikimedia Commons

Bilinear quadrature (approximates bilinear forms):

$$\int_\Omega f(x)g(x)\, d\mu = \begin{bmatrix} f(y_1) & \dots & f(y_m) \end{bmatrix} \begin{bmatrix} w_{11} & \dots & w_{1n} \\ \vdots & & \vdots \\ w_{m1} & \dots & w_{mn} \end{bmatrix} \begin{bmatrix} g(x_1) \\ \vdots \\ g(x_n) \end{bmatrix}.$$

- Useful for finite element/Galerkin method.

Classical and bilinear quadratures

Classical quadrature (approximates linear functionals):

$$\int_\Omega f(x)\,d\mu = \begin{bmatrix} w_1 & \dots & w_n \end{bmatrix} \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}$$

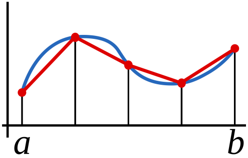


Image from Wikimedia Commons

Bilinear quadrature (approximates bilinear forms):

$$\int_\Omega f(x)g(x)\,d\mu = \begin{bmatrix} f(y_1) & \dots & f(y_m) \end{bmatrix} \begin{bmatrix} w_{11} & \dots & w_{1n} \\ \vdots & & \vdots \\ w_{m1} & \dots & w_{mn} \end{bmatrix} \begin{bmatrix} g(x_1) \\ \vdots \\ g(x_n) \end{bmatrix}.$$

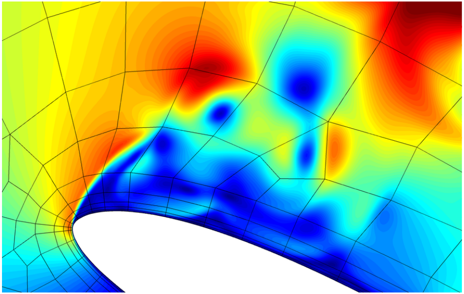
- Useful for finite element/Galerkin method.
- Previous work: Boland and Duris (1972), McGrath (1979), Gribble (1980), Chen (2012)

Advantages

- f's and g's can belong to different spaces.
- Can do any continuous bilinear form, e.g.

$$\langle f, g\rangle_{H^1} = \int_\Omega fg + Du \cdot Dv, \quad \langle f, g\rangle = \int_\Omega fg + \int_{\partial\Omega} fg.$$

  Very useful in Galerkin methods!



- The **right method** for computing an orthogonal projection onto a fixed function space.

Image courtesy of P.-O. Persson

Looking for a bilinear quadrature

- Choose L^2 inner product on some finite-dim. function space X_0 (e.g. polynomials)

Looking for a bilinear quadrature

- Choose L^2 inner product on some finite-dim. function space X_0 (e.g. polynomials)
- Pick orthonormal basis $\{f_1, \ldots, f_k\}$ for X_0. Solve for $W = (w_{ij})$ and $\mathbf{x} = \{x_i\}$ such that:

$$F(\mathbf{x})^T W F(\mathbf{x}) = I_k$$

where

$$F(\mathbf{x}) := \begin{bmatrix} f_1(x_1) & \ldots & f_k(x_1) \\ \vdots & & \vdots \\ f_1(x_n) & \ldots & f_k(x_n) \end{bmatrix}$$

Looking for a bilinear quadrature

- Choose L^2 inner product on some finite-dim. function space X_0 (e.g. polynomials)
- Pick orthonormal basis $\{f_1, \ldots, f_k\}$ for X_0. Solve for $W = (w_{ij})$ and $\mathbf{x} = \{x_i\}$ such that:

$$F(\mathbf{x})^T W F(\mathbf{x}) = I_k$$

  where

$$F(\mathbf{x}) := \begin{bmatrix} f_1(x_1) & \ldots & f_k(x_1) \\ \vdots & & \vdots \\ f_1(x_n) & \ldots & f_k(x_n) \end{bmatrix}$$

- Approximate coefficients of $\mathrm{Proj}_{X_0}(\phi)$ using matvec $F(\mathbf{x})^T W \phi(\mathbf{x})$

Nonlinear optimization

- Find points **x** by minimizing an objective. Inspired by Chen, Rokhlin, and Yarvin (1999), Bremer, Gimbutas, Rokhlin (2010), Xiao and Gimbutas (2010)

Nonlinear optimization

- Find points **x** by minimizing an objective. Inspired by Chen, Rokhlin, and Yarvin (1999), Bremer, Gimbutas, Rokhlin (2010), Xiao and Gimbutas (2010)
- Choose orthonormal set $\{g_1, \ldots, g_p\}$ orthogonal to X_0 and minimize the pairing with X_0.

Nonlinear optimization

- Find points $\mathbf{x}$ by minimizing an objective. Inspired by Chen, Rokhlin, and Yarvin (1999), Bremer, Gimbutas, Rokhlin (2010), Xiao and Gimbutas (2010)
- Choose orthonormal set $\{g_1, \ldots, g_p\}$ orthogonal to X_0 and minimize the pairing with X_0.

$$G(\mathbf{x}) = \begin{bmatrix} g_1(x_1) & \ldots & g_p(x_1) \\ \vdots & & \vdots \\ g_1(x_n) & \ldots & g_p(x_n) \end{bmatrix},$$

then must solve

$$\text{Find } \mathbf{x}, W \text{ minimizing } \|F(\mathbf{x})^T W G(\mathbf{x})\|_2 \text{ subject to } F(\mathbf{x})^T W F(\mathbf{x}) = I_k$$

Nonlinear optimization

- Find points $\mathbf{x}$ by minimizing an objective. Inspired by Chen, Rokhlin, and Yarvin (1999), Bremer, Gimbutas, Rokhlin (2010), Xiao and Gimbutas (2010)
- Choose orthonormal set $\{g_1, \ldots, g_p\}$ orthogonal to X_0 and minimize the pairing with X_0.

$$G(\mathbf{x}) = \begin{bmatrix} g_1(x_1) & \ldots & g_p(x_1) \\ \vdots & & \vdots \\ g_1(x_n) & \ldots & g_p(x_n) \end{bmatrix},$$

then must solve

$$\text{Find } \mathbf{x}, W \text{ minimizing } \|F(\mathbf{x})^T W G(\mathbf{x})\|_2 \text{ subject to } F(\mathbf{x})^T W F(\mathbf{x}) = I_k$$

- Don't have access to derivatives. Use a quasi-Newton method (e.g. BFGS).
- Need many initial guesses to get close to the minimum.

Connection with familiar classical quadratures

Theorem

Given:

- *n-point bilinear quadrature on an interval,*
- *exact for L^2 inner products on $\mathbb{P}_{n-1} \times \mathbb{P}_{n-1}$,*
- *minimized over degree n polynomials,*

then $W = \mathrm{diag}(Gauss\ weights)$ and $\mathbf{x}$ are the Gauss points from Gaussian quadrature!

Connection with familiar classical quadratures

Theorem

Given:

- *n-point bilinear quadrature on an interval,*
- *exact for L^2 inner products on $\mathbb{P}_{n-1} \times \mathbb{P}_{n-1}$,*
- *minimized over degree n polynomials,*

then $W = \operatorname{diag}(Gauss\ weights)$ and $\mathbf{x}$ are the Gauss points from Gaussian quadrature!

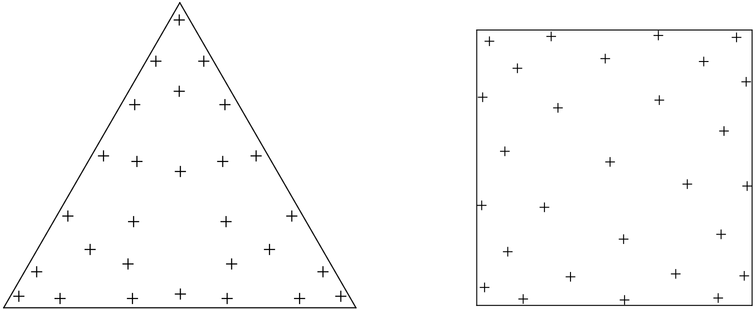
Theorem

Given:

- *n-point bilinear quadrature on $[0, 2\pi]$, n odd,*
- *exact for L^2 inner products on $T_{(n-1)/2} \times T_{(n-1)/2}$,*
- *minimized over* $\operatorname{span}\{\sin \frac{n+1}{2}x, \cos \frac{n+1}{2}x\}$,

then $W = \operatorname{diag}(\frac{\pi}{n}, \frac{2\pi}{n}, \ldots, \frac{2\pi}{n}, \frac{\pi}{n})$ and $\mathbf{x}$ are uniformly spaced points on $[0, 2\pi]$!

Two numerical results

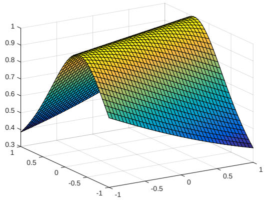


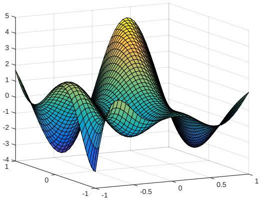
The nodes of 28-point bilinear quadratures on $\mathbb{P}_6 \times \mathbb{P}_6$, minimized against $\mathbb{P}_7$, for both the triangle and the square.
Notice symmetry!

Accuracy

- Compare with two very good 28-point classical quadratures on triangles.
- Project onto $\mathbb{P}_6$.
- Calculate relative 2-norm error in projection coefficients for random functions from four different spaces.

	$\mathbb{P}_5$	$\mathbb{P}_6$	Cauchy	Trig
Dunavant (1985)	9.38e-14	3.97e-01	4.96e-05	9.12e-03
Xiao et al. (2010)	3.29e-15	2.73e-01	1.91e-05	4.76e-03
Bilinear	3.92e-15	3.99e-15	6.70e-06	1.71e-03





Conclusions

Advantages:

- Ideal for computing orthogonal projections.
- Independent of domain or choice of functions.

Conclusions

Advantages:

- Ideal for computing orthogonal projections.
- Independent of domain or choice of functions.

Disadvantages

- A bilinear quadrature does less than a general-purpose classical quadrature.
- Expensive to construct; optimization is slow.