Set 17: Interpolation, Quadrature and GFS

Kyle A. Gallivan

Department of Mathematics

Florida State University

Foundations of Computational Math 2
Spring 2013

Generalized Fourier Series

- Let $\{P_k\}_{k=0}^{\infty}$ be a complete sequence of orthogonal polynomials for inner product $(f, g)_\omega$ on $\mathcal{L}^2_\omega[a, b]$.
- For any $f \in \mathcal{L}^2_\omega[a, b]$ the GFS representation is

$$f(x) = \sum_{k=0}^{\infty} \gamma_k P_k(x), \quad \gamma_k = \frac{(f, P_k)_\omega}{(P_k, P_k)_\omega}$$

- The truncated GFS is a polynomial of degree n

$$f_n(x) = \sum_{k=0}^{n} \gamma_k P_k(x) \in \mathbb{P}_n, \quad \gamma_k = \frac{(f, P_k)_\omega}{(P_k, P_k)_\omega}$$

Generalized Fourier Series

- Assume a linear quadrature method is defined and used to define a discrete inner product

$$(g, h)_\omega \approx (g, h)_n = \sum_{i=0}^{n} \alpha_i g(x_i) h(x_i)$$

(Note we have assumed a real space.)

- The quadrature method of interest here will be Gaussian or Gauss-Lobatto based on the sequence $\{P_k\}_{k=0}^{\infty}$.

Discrete Truncated Generalized Fourier Series

$$f(x) = \sum_{k=0}^{\infty} \gamma_k P_k(x), \quad \gamma_k = \frac{(f, P_k)_\omega}{(P_k, P_k)_\omega}$$

$$f_n(x) = \sum_{k=0}^{n} \gamma_k P_k(x) \in \mathbb{P}_n, \quad \gamma_k = \frac{(f, P_k)_\omega}{(P_k, P_k)_\omega}$$

$$f_n^*(x) = \sum_{k=0}^{n} \bar{\gamma}_k P_k(x) \in \mathbb{P}_n, \quad \bar{\gamma}_k = \frac{(f, P_k)_n}{(P_k, P_k)_n}$$

Gauss Quadrature

- If Gauss Quadrature is used then the degree of exactness is $2n+1$ and

$$(P_i, P_j)_\omega = (P_i, P_j)_n, \quad 0 \leq i,\ j \leq n$$

- In general, for the Gauss quadrature points x_i,

$$(f, P_j)_n = \sum_{i=0}^{n} \alpha_i f(x_i) P_i(x_i) \neq (f, P_j)_\omega$$

- For any $q_n \in \mathbb{P}_n$

$$(q_n, P_j)_\omega = (q_n, P_j)_n, \quad 0 \leq j \leq n$$

Gauss Quadrature and Interpolation

- Let $p_n^{(G)}(x) \in \mathbb{P}_n$ be the unique interpolating polynomial such that $f(x_i) = p_n^{(G)}(x_i), \;\; 0 \le i \le n.$

- It follows that

$$(f, P_k)_n = \sum_{i=0}^{n} \alpha_i f(x_i) P_i(x_i) = \sum_{i=0}^{n} \alpha_i p_n^{(G)}(x_i) P_i(x_i)$$

$$= (p_n^{(G)}, P_k)_n = (p_n^{(G)}, P_k)_\omega, \quad 0 \le k \le n+1$$

Gauss Quadrature and Interpolation

We therefore have,

$$
\begin{aligned}
f_n^*(x) &= \sum_{k=0}^{n} \frac{(f, P_k)_n}{(P_k, P_k)_n} P_k(x) \\
&= \sum_{k=0}^{n} \frac{(f, P_k)_n}{(P_k, P_k)_\omega} P_k(x) \\
&= \sum_{k=0}^{n} \frac{(p_n^{(G)}, P_k)_n}{(P_k, P_k)_\omega} P_k(x) \\
&= \sum_{k=0}^{n} \frac{(p_n^{(G)}, P_k)_\omega}{(P_k, P_k)_\omega} P_k(x) \\
&= p_n^{(G)}(x)
\end{aligned}
$$

by previous results and GFS of $p_n^{(G)}(x)$.

Gauss-Lobatto Quadrature

If Gauss-Lobatto quadrature is used the degree of exactness drops to $2n-1$ and

$$\begin{aligned} f_n^*(x) &= \sum_{k=0}^{n-1} \frac{(f, P_k)_n}{(P_k, P_k)_n} P_k(x) + \frac{(f, P_n)_n}{(P_n, P_n)_n} P_n(x) \\ &= \sum_{k=0}^{n-1} \frac{(p_n^{(GL)}, P_k)_n}{(P_k, P_k)_\omega} P_k(x) + \frac{(f, P_n)_n}{(P_n, P_n)_n} P_n(x) \\ &= \sum_{k=0}^{n-1} \frac{(p_n^{(GL)}, P_k)_\omega}{(P_k, P_k)_\omega} P_k(x) + \frac{(p_n^{(GL)}, P_n)_n}{(P_n, P_n)_n} P_n(x) \end{aligned}$$

So all terms except the last are verified as matching the GFS of $p_n^{(GL)}(x)$.

Gauss-Lobatto Quadrature and Interpolation

Lemma. *When Gauss-Lobatto quadrature with degree of exactness* $2n-1$ *defines the discrete inner product*

$$\frac{(p_n^{(GL)}, P_n)_\omega}{(P_n, P_n)_\omega} = \frac{(f, P_n)_n}{(P_n, P_n)_n}$$

and $p_n^{(GL)}(x) = f_n^*(x)$ *follows.*

Gauss-Lobatto Quadrature and Interpolation

Proof:

$$p_n^{(GL)}(x) = \sum_{j=0}^{n} \beta_j P_j(x), \quad \beta_j = \frac{(f, P_n)_\omega}{(P_n, P_n)_\omega}$$

$$\begin{aligned}
(p_n^{(GL)}, P_n) &= \sum_{i=0}^{n} \alpha_i p_n^{(GL)}(x_i) P_n(x_i) \\
&= \sum_{i=0}^{n} \alpha_i P_n(x_i) \left(\sum_{j=0}^{n} \beta_j P_j(x_i) \right) \\
&= \sum_{j=0}^{n} \beta_j \sum_{i=0}^{n} \alpha_i P_n(x_i) P_j(x_i)
\end{aligned}$$

Gauss-Lobatto Quadrature and Interpolation

Proof continued:

$$(p_n^{(GL)}, P_n) = \sum_{j=0}^{n} \beta_j (P_n, P_j)_n$$

$$= \sum_{j=0}^{n-1} \beta_j (P_n, P_j)_n + \beta_n (P_n, P_n)_n$$

$$= \sum_{j=0}^{n-1} \beta_j (P_n, P_j)_\omega + \beta_n (P_n, P_n)_n$$

$$(p_n^{(GL)}, P_n) = \beta_n (P_n, P_n)_n \qquad \square$$

Summary

- Gauss and Gauss-Lobatto quadrature methods applied to the coefficients in a truncated GFS yield the expansion of the unique interpolating polynomial of $f(x)$ at the points defining the quadrature method.
- Gauss-Lobatto quadrature produces these coefficients using lower degree of exactness than Gauss quadrature.
- The coefficients computed via Gauss-Lobatto quadrature define the discrete transform associated with the sequence of complete orthogonal polynomials, e.g., the discrete Legendre transform, the discrete Chebyshev transform, etc.

Summary

- The discrete Chebyshev is related to the discrete cosine transform family which along with the DFT are used extensively in image and signal processing.
- Very large literature on so-called number theoretic transforms, e.g., Winograd transform.
- FFT trick related to Winograd transform which is used to reduce the complexity of complex multiplication and matrix multiplication (Strassen's method).

Discrete Chebyshev/Cosine Transform

Source: Computational Frameworks for the Fast Fourier Transform, Van Loan, SIAM.

- Discrete Cosine Transform I:

$$\hat{f}_k = \frac{f_0}{2} + \frac{(-1)^k}{2} f_n + \sum_{j=1}^{n-1} \cos\left(\frac{kj\pi}{n}\right) f_j = \frac{n}{\pi} \left(T_k, f\right)_n^{(GL)}$$

- Discrete Cosine Transform II:

$$\hat{f}_k = \sum_{j=0}^{n} \cos\left(\frac{k(2j+1)\pi}{2(n+1)}\right) f_j = \frac{(n+1)}{\pi} \left(T_k, f\right)_n^{(G)}$$

where the (G) indicates Gauss-Chebyshev Quadrature and (GL) indicates Gauss-Chebyshev-Lobatto Quadrature.