Homework #2 is due the beginning of class on Tuesday, May 10, although you may submit an electronic version before then. Each person should turn in his or her own write-up. No late homeworks will be accepted.

Problem 1. Concepts

This homework section will serve as the notes for the univariate polynomial material of the course. Your assignment is to complete (i) Exercises 4 and 5, and (ii) two other exercises of your choice. If you complete all the exercises, you earn the title of Total Polynomial Ninja.

Consider a function $f : \mathcal{S} \to \mathbb{R}$, where the domain $\mathcal{S} \subset \mathbb{R}$. Choices for the domain include closed, semi-infinite, or infinite intervals, i.e. $[a, b]$, $[a, \infty)$, $(-\infty, b]$, or $(-\infty, \infty)$. Let $s \in \mathcal{S}$ be a point in the domain. We assume that the domain is equipped with a positive weight function $w : \mathcal{S} \to \mathbb{R}_+$ such that

$$\int_{\mathcal{S}} s^k w(s)\, ds < \infty, \qquad k = 1, 2, \ldots \tag{1}$$

We assume that $w(s)$ is normalized to integrate to 1, which allows the interpretation of $w(s)$ as a probability density function. In general, we consider functions which are square-integrable on $\mathcal{S}$, i.e.

$$\left\langle f^2 \right\rangle \equiv \int_{\mathcal{S}} f(s)^2 w(s)\, ds < \infty, \tag{2}$$

where the bracket notation denotes integration against the weight function. However, the polynomial approximation methods work best for smooth functions – particularly functions that are analytic in a region of the complex plane containing $\mathcal{S}$.

Orthogonal Polynomials and Gaussian Quadrature: Let $\mathbb{P}$ be the space of real polynomials defined on $\mathcal{S}$, and let $\mathbb{P}_n \subset \mathbb{P}$ be the space of polynomials of degree at most n. For any p, q in $\mathbb{P}$, we define the inner product as

$$\langle pq \rangle \equiv \int_{\mathcal{S}} p(s) q(s) w(s)\, ds. \tag{3}$$

We define a norm on $\mathbb{P}$ as $\|p\|_{L^2} = \sqrt{\langle p^2 \rangle}$, which is the standard L^2 norm for the given weight $w(s)$. Let $\{\pi_k(s)\}$ be the set of polynomials that are orthonormal with respect to $w(s)$, i.e. $\langle \pi_i \pi_j \rangle = \delta_{ij}$. It is known that $\{\pi_k(s)\}$ satisfy a three-term recurrence relation

$$\beta_{k+1}\pi_{k+1}(s) = (s - \alpha_k)\pi_k(s) - \beta_k \pi_{k-1}(s), \qquad k = 0, 1, 2, \ldots, \tag{4}$$

with $\pi_{-1}(s) = 0$ and $\pi_0(s) = 1$. If we consider only the first n equations, then we can rewrite (4) as

$$s\pi_k(s) = \beta_k \pi_{k-1}(s) + \alpha_k \pi_k(s) + \beta_{k+1}\pi_{k+1}(s), \qquad k = 0, 1, \ldots, n-1. \tag{5}$$

Setting $\boldsymbol{\pi}(s) = [\pi_0(s), \pi_1(s), \ldots, \pi_{n-1}(s)]^T$, we can write this conveniently in matrix form as

$$s\boldsymbol{\pi}(s) = \mathbf{J}\boldsymbol{\pi}(s) + \beta_n \pi_n(s)\mathbf{e}_n \tag{6}$$

where $\mathbf{e}_n$ is a vector of zeros with a one in the last entry, and $\mathbf{J}$ (known as the *Jacobi matrix*) is a symmetric, tridiagonal matrix defined as

$$\mathbf{J}_n = \begin{bmatrix} \alpha_0 & \beta_1 & & & \\ \beta_1 & \alpha_1 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-2} & \alpha_{n-2} & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_{n-1} \end{bmatrix}. \tag{7}$$

EXERCISE 1: Use (6) to show that $\alpha_i = \langle s\pi_i^2 \rangle$ and $\beta_i = \langle s\pi_{i-1}\pi_i \rangle$.

The zeros $\{\lambda_i\}$ of $\pi_n(s)$ are the eigenvalues of $\mathbf{J}$ and $\boldsymbol{\pi}(\lambda_i)$ are the corresponding eigenvectors; this follows directly from (6). Let $\mathbf{Q}$ be the orthogonal matrix of eigenvectors of $\mathbf{J}$; the elements of $\mathbf{Q}$ are given by

$$\mathbf{Q}(i,j) = \frac{\pi_i(\lambda_j)}{\|\boldsymbol{\pi}(\lambda_j)\|_2}, \qquad i,j = 0, \ldots, n-1, \tag{8}$$

where $\|\cdot\|_2$ is the standard 2-norm on $\mathbb{R}^n$. Then we write the eigenvalue decomposition of $\mathbf{J}$ as

$$\mathbf{J} = \mathbf{Q}\Lambda\mathbf{Q}^T. \tag{9}$$

It is known that the eigenvalues $\{\lambda_i\}$ are the familiar Gaussian quadrature points associated with the weight function $w(s)$. The quadrature weight ν_i corresponding to λ_i is equal to the square of the first component of the eigenvector associated with λ_i, i.e.

$$\nu_i \;=\; \mathbf{Q}(0,i)^2 \;=\; \frac{1}{\|\boldsymbol{\pi}(\lambda_i)\|_2^2}. \tag{10}$$

The weights $\{\nu_i\}$ are known to be strictly positive. For an integrable scalar function $f(s)$, we can approximate its integral by an n-point Gaussian quadrature rule, which is a weighted sum of function evaluations,

$$\int_{\mathcal{S}} f(s)w(s)\,ds = \sum_{i=0}^{n-1} f(\lambda_i)\nu_i + R(f). \tag{11}$$

If $f \in \mathbb{P}_{2n-1}$, then $R_n(f) = 0$; that is to say the *degree of exactness* of the Gaussian quadrature rule is $2n-1$. We use the notation

$$\langle f \rangle_n \equiv \sum_{i=0}^{n-1} f(\lambda_i)\nu_i \tag{12}$$

to denote the Gaussian quadrature rule. This is a discrete approximation to the true integral.

EXERCISE 2: Show that $\langle f \rangle_n = \mathbf{e}_1^T f(\mathbf{J}) \mathbf{e}_1$, where $\mathbf{e}_1$ is a vector of zeros with a 1 in the first component.

Fourier Series: The polynomials $\{\pi_k(s)\}$ form an orthonormal basis for the Hilbert space

$$L^2 \equiv L^2_w(\mathcal{S}) = \{f : \mathcal{S} \to \mathbb{R} \mid \|f\|_{L^2} < \infty\}. \tag{13}$$

Therefore, any $f \in L^2$ admits a convergent *Fourier series*

$$f(s) = \sum_{k=0}^{\infty} \langle f \pi_k \rangle \, \pi_k(s). \tag{14}$$

The coefficients $\langle f \pi_k \rangle$ are called the *Fourier coefficients*. If we truncate the series (14) after n terms, we are left with a polynomial of degree $n-1$ that is the best approximation polynomial in the L^2 norm. In other words, if we denote

$$P_n f(s) = \sum_{k=0}^{n-1} \langle f \pi_k \rangle \, \pi_k(s), \tag{15}$$

then

$$\|f - P_n f\|_{L^2} = \inf_{p \in \mathbb{P}_{n-1}} \|f - p\|_{L^2}. \tag{16}$$

In fact, the error made by truncating the series is equal to the sum of squares of the neglected coefficients,

$$\|f - P_n f\|^2_{L^2} = \sum_{k=n}^{\infty} \langle f \pi_k \rangle^2. \tag{17}$$

These properties of the Fourier series motivate the theory and practice of spectral methods.

Spectral Collocation and Pseudospectral Approximations: It will be notationally convenient to define the matrices $\mathbf{P}(i,j) = \pi_i(\lambda_j)$ and $\mathbf{W} = \mathrm{diag}([\sqrt{\nu_0}, \ldots, \sqrt{\nu_{n-1}}])$, and note that the orthogonal matrix of eigenvectors $\mathbf{Q}$ can be written $\mathbf{Q} = \mathbf{PW}$.

The spectral collocation approximation of $f(s)$ constructs a Lagrange interpolating polynomial through the Gaussian quadrature points. Since the points are distinct, the $n-1$ degree interpolating polynomial is unique. We write this approximation $f_c(s)$, where the subscript c is for *collocation*, as

$$f(s) \approx f_c(s) = \sum_{i=0}^{n-1} f(\lambda_i) \ell_i(s) \equiv \mathbf{f}^T \mathbf{l}(s). \tag{18}$$

The vector $\mathbf{f}$ contains the evaluations of $f(s)$ at the quadrature points, and the parameterized vector $\mathbf{l}(s)$ contains the Lagrange cardinal functions

$$\ell_i(s) = \prod_{j=0,\, j \neq i}^{n-1} \frac{s - \lambda_j}{\lambda_i - \lambda_j}. \tag{19}$$

By construction, the collocation polynomial $f_c(s)$ interpolates $f(s)$ at the Gaussian quadrature points.

The pseudospectral approximation of $f(s)$ is constructed by first truncating its Fourier series at n terms and approximating each Fourier coefficient with a quadrature rule. If we use the n-point Gaussian quadrature, then we can write

$$f(s) \;\approx\; f_p(s) \;=\; \sum_{i=0}^{n-1} \langle f\pi_i \rangle_n \, \pi_i(s) \;\equiv\; \hat{\mathbf{f}}^T \boldsymbol{\pi}(s), \tag{20}$$

where

$$\langle f\pi_i \rangle_n \equiv \sum_{j=0}^{n-1} f(\lambda_j)\pi_i(\lambda_j)\nu_j, \tag{21}$$

and the vector $\hat{\mathbf{f}}$ contains all coefficient approximations; the subscript p on $f_p(s)$ is for *pseudospectral*. We next state two lemmas about the relationship between the spectral collocation and pseudospectral approximations for future reference.

Lemma 1. *The vector of evaluations of f at the quadrature points $\mathbf{f}$ is related to the pseudospectral coefficients $\hat{\mathbf{f}}$ by*

$$\hat{\mathbf{f}} \;=\; \mathbf{QWf} \;=\; \mathbf{PW}^2\mathbf{f}. \tag{22}$$

Proof: *EXERCISE 3.* ∎

In the language of signal processing, this tranformation is called a *discrete Fourier transform*.

Lemma 2. *The pseudospectral approximation $f_p(s)$ is equal to the spectral collocation approximation $f_c(s)$ for all $s \in \mathcal{S}$.*

Proof: *EXERCISE 4.* (Hint: Use the fact that a Lagrange interpolant of the orthogonal basis evaluated at the Gaussian quadrature points exactly reproduces the orthogonal basis.) ∎

Note that Lemma 2 implies that the pseudospectral approximation $f_p(s)$ interpolates $f(s)$ at the Gaussian quadrature points.

Remark: We have restricted our attention to orthonormal polynomials and Gaussian quadrature rules for a given weight function. However, transformations similar to Lemma 1 apply for Chebyshev polynomials and Clenshaw-Curtis quadrature rules using an FFT. For an insightful discussion of the comparisions between these methods of integration and approximation, see Trefethen (SIAM Review, 2008).

Least Squares Approximation: Suppose we are given n points $s_j \in \mathcal{S}$ and function evaluations $f_j = f(s_j)$ with $j = 0, \ldots, n-1$. We want to find a polynomial of degree $m-1 \leq n-1$ that best approximates $f(s)$ at the points s_j. (The $m-1$ is used as

opposed to m to be consistent.) We know that any polynomial of degree $m-1$ can be written as a linear combination of the first m basis polynomials,

$$f(s) \approx \sum_{i=0}^{m-1} x_i \pi_i(s). \tag{23}$$

This yields the least squares approximation problem

$$\underset{x_i}{\text{minimize}} \quad \left(\sum_{i=0}^{m-1} x_i \pi_i(s_j) - f_j \right)^2, \qquad j = 0, \ldots, n-1. \tag{24}$$

Let $\mathbf{x}$ be the m-vector whose ith component is x_i, and let $\mathbf{f}$ be the n-vector whose jth component is f_j. Define the $n \times m$ matrix $\mathbf{P}$ with elements $\mathbf{P}(i,j) = \pi_i(s_j)$. Then we can write the least squares approximation problem as

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|\mathbf{P}^T\mathbf{x} - \mathbf{f}\|_2^2. \tag{25}$$

EXERCISE 5. Suppose that $s_j = \lambda_j$, i.e. that the points chosen to evaluate $f(s)$ were from the n-point Gaussian quadrature rule. Show that, if the jth equation in (24) is weighted by the Gaussian quadrature weight ν_j, then the pseudospectral coefficients $\hat{\mathbf{f}}$ from (21) satisfy the least squares approximation problem (25). (Hint: Use the normal equations).

Spectral Galerkin Approximation: Suppose that $f(s) = b(s)/a(s)$ is a rational function of s with $a(s) > 0$ for all $s \in \mathcal{S}$. The spectral Galerkin method computes a finite dimensional approximation to $f(s)$ such that each element of the equation residual is orthogonal to the approximation space. Define

$$r(y,s) = a(s)y(s) - b(s). \tag{26}$$

The finite dimensional approximation space for $f(s)$ will be the space of polynomials of degree at most $n-1$. We seek a polynomial $f_g(s)$ (the subscript g is for *Galerkin*) of maximum degree $n-1$ such that

$$\langle r(f_g)\pi_k \rangle = 0, \qquad k = 0, \ldots, n-1. \tag{27}$$

We can write equations (27) in matrix notation as

$$\left\langle r(f_g)\boldsymbol{\pi}^T \right\rangle = \mathbf{0}, \tag{28}$$

or equivalently

$$\left\langle a f_g \boldsymbol{\pi}^T \right\rangle = \left\langle b \boldsymbol{\pi}^T \right\rangle. \tag{29}$$

Since $f_g(s)$ is a polynomial of degree at most $n-1$, we can write its expansion in $\{\pi_i(s)\}$ as

$$f_g(s) \;=\; \sum_{i=0}^{n-1} g_i \pi_i(s) \;\equiv\; \mathbf{g}^T \boldsymbol{\pi}(s), \tag{30}$$

where $\mathbf{g}$ is an n-vector. Then equation (29) becomes

$$\left\langle a\mathbf{g}^T\boldsymbol{\pi}\boldsymbol{\pi}^T\right\rangle = \left\langle b\boldsymbol{\pi}^T\right\rangle. \tag{31}$$

Since $\mathbf{g}$ is constant, we can rewrite (31) as

$$\left\langle a\boldsymbol{\pi}\boldsymbol{\pi}^T\right\rangle \mathbf{g} = \left\langle b\boldsymbol{\pi}\right\rangle. \tag{32}$$

The (i,j) element of the $n \times n$ constant matrix $\left\langle a\boldsymbol{\pi}\boldsymbol{\pi}^T\right\rangle$ is equal to $\langle a\pi_i\pi_j\rangle$. More explicitly,

$$\left\langle a\boldsymbol{\pi}\boldsymbol{\pi}^T\right\rangle = \begin{bmatrix} \langle a\pi_0\pi_0\rangle & \cdots & \langle a\pi_0\pi_{n-1}\rangle \\ \vdots & \ddots & \vdots \\ \langle a\pi_{n-1}\pi_0\rangle & \cdots & \langle a\pi_{n-1}\pi_{n-1}\rangle \end{bmatrix}. \tag{33}$$

Similarly, the ith element of the n-vector $\langle b\boldsymbol{\pi}\rangle$ is equal to $\langle b\pi_i\rangle$, which is exactly the ith Fourier coefficient of $b(s)$. In the language of signal processing, equation (32) can be interpreted as a deconvolution.

Lemma 3. *Suppose that $a(s)$ is a polynomial of degree at most m. Then for $n > m+1$, the matrix $\left\langle a\boldsymbol{\pi}\boldsymbol{\pi}^T\right\rangle$ will have bandwidth $2m+1$.*

Proof: *EXERCISE 6.* ∎

Lemma 4. *Let $f(s)$ be analytic in a region of the complex plane containing $\mathcal{S}$. Then $\left\langle f\boldsymbol{\pi}\boldsymbol{\pi}^T\right\rangle_n = f(\mathbf{J})$.*

Proof: *EXERCISE 7.* ∎

Theorem 5. *The pseudospectral approximation $f_p(s)$ is equal to an approximation of the Galerkin solution where each integral in equation (32) is approximated by an n-point Gaussian quadrature formula. In other words, $\hat{\mathbf{f}}$ satisfies*

$$\left\langle a\boldsymbol{\pi}\boldsymbol{\pi}^T\right\rangle_n \hat{\mathbf{f}} = \left\langle b\boldsymbol{\pi}\right\rangle_n. \tag{34}$$

Proof: *EXERCISE 8.* ∎

Theorem 6. *Suppose that $a(s) = a_0 + a_1 s$ for some constants a_0 and a_1. If b is a polynomial of degree $n-1$, then the pseudospectral approximation $f_p(s)$ of $f(s) = b(s)/a(s)$ with an n-point Gaussian quadrature rule is equal to a spectral Galerkin approximation for all $s \in \mathcal{S}$.*

Proof: *EXERCISE 9.* ∎

Point	Weight
(-0.77,-0.58)	0.14
(0.00,-0.58)	0.22
(0.77,-0.58)	0.14
(-0.77,0.58)	0.14
(0.00,0.58)	0.22
(0.77,0.58)	0.14

Table 1: A bivariate quadrature rule corresponding to $\mathbf{n} = (3, 2)$.

Problem 2. Programming

In this exercise, you will write a program to compute the points and weights of a sparse grid quadrature rule given code to compute the points and weights of a tensor product quadrature rule. The code for computing the tensor product rules is given in the PMPack suite of Matlab tools, which is available on the website, along with a test script.

Consider a function $f : \mathcal{S} \to \mathbb{R}$ where $\mathcal{S} = [-1, 1]^d$ is a d-dimensional hypercube, and let $w(s)$ be the weight function on $\mathcal{S}$. Given a multi-index $\mathbf{n} = (n_1, \ldots, n_d) \in \mathbb{N}^d$, a tensor product quadrature rule is formed by the cross product of univariate n_k-point quadrature rules, $k = 1, \ldots, d$. For example, a 3-point rule in the first variable with points and weights

$$\begin{array}{rclrcl} \lambda_0 & = & -0.77 & \nu_0 & = & 0.28 \\ \lambda_1 & = & 0.00 & \nu_1 & = & 0.44 \\ \lambda_2 & = & 0.77 & \nu_2 & = & 0.28 \end{array} \tag{35}$$

can be combined with a 2-point rule in the second variable

$$\begin{array}{rclrcl} \lambda_0 & = & -0.58 & \nu_0 & = & 0.50 \\ \lambda_1 & = & 0.58 & \nu_1 & = & 0.50 \end{array} \tag{36}$$

to create a tensor product rule for bivariate integration given in Table 1. The weights of the tensor product rule are products of the weights for each univariate rule.

Let $I_{\mathbf{n}}(f)$ be the tensor product quadrature rule approximation of the integral

$$I_{\mathbf{n}}(f) \approx \int_{\mathcal{S}} f(s)w(s)\, ds.$$

A sparse grid quadrature formula can be written as a linear combination of specially chosen tensor product quadrature formulas. We can define a constraint $c : \mathbb{N}^d \to \{0, 1\}$ that takes a multi-index and returns 1 if the constraint is satisfied, and zero if it is not. A common constraint in the sparse grid quadrature construction is

$$c(\mathbf{n}) = \left(l + 1 \leq \sum_{k=1}^{d} n_k \leq l + d \right), \tag{37}$$

where the parameter l is called the *level*. Let $\mathcal{C}$ be the set of multi-indices that satisfies the constraints, i.e.

$$\mathcal{C} = \{\mathbf{n} \ : \ c(\mathbf{n}) = 1, \quad \mathbf{n} \in \mathbb{N}^d\}. \tag{38}$$

We also define a growth rule $\gamma : \mathbb{N}^d \to \mathbb{N}^d$, which dictates the number of points in the univariate quadrature rules given a multi-index. For example, an exponential growth rule may be given by

$$\gamma(n_k) = 2^{n_k} - 1, \qquad n_k \geq 1, \ \ k = 1, \ldots, d. \tag{39}$$

An exponential growth rule is useful if the univariate point sets are nested, i.e. the points of an n-point rule are a subset of the points of a $(2n+1)$-point rule; this happens for the Chebyshev points. A sparse grid built from univariate rules with a nesting property will also be nested.

Given a constraint $c = c(\mathbf{n})$ and growth rule $\gamma = \gamma(\mathbf{n})$, we can compute a sparse grid approximation to the integral as

$$\int_{\mathcal{S}} f(s)w(s)\,ds \approx \sum_{\mathbf{n}\in\mathcal{C}} a_{\mathbf{n}} I_{\gamma(\mathbf{n})}(f), \tag{40}$$

where $a_{\mathbf{n}}$ is the coefficient of the linear combination, which are typically given with the constraint.

Note that some of the function evaluations in the various $I_{\gamma(\mathbf{n})}(f)$ may occur at the same point in the parameter space. Therefore, a naive implementation of (40) is inefficient – particularly if the function evaluations are very expensive.

A. Write a program that takes the points and weights of each individual tensor product quadrature rule and produces a list of d-dimensional points and corresponding weights for the sparse grid quadrature rule. Use the constraint from (37) with coefficients

$$a_{\mathbf{n}} = (-1)^p \binom{d-1}{p}, \qquad p = d + l - (n_1 + \cdots + n_d).$$

Use the growth rule from (39). Plot the points for $d = 2$ and levels 3 and 6.

B. Perform a refinement study by increasing the level l from 1 to 7 on the following three bivariate functions: (i) $e^{s_1+s_2}$, (ii) $\sin(5(s_1 - 0.5)) + \cos(3(s_2 - 1))$, and (iii) $1/(2 + 16(s_1 - 0.1)^2 + 25(s_2 + 0.1)^2)$. Use a uniform weight function.

C. Divide each function above by the Chebyshev weight function $w(s_1, s_2) = (1 - s_1^2)^{1/2}(1 - s_2^2)^{1/2}$, and use the points from the Chebyshev measure, which are nested. Compare the number of function evaluations required to the number of function evaluations for each function from part B.

Problem 3. Reading

Read the following papers.

A. Paul G. Constantine, David F. Gleich, and Gianluca Iaccarino. *Spectral Methods for Parameterized Matrix Equations.* SIAM Journal of Matrix Analysis and Applications, 2010.

B. Dongbin Xiu and George Em Karniadakis. *The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations.* SIAM Journal of Scientific Computing, 2002.

C. Dongbin Xiu and Jan S. Hesthaven. *High-Order Collocation Methods for Differential Equations with Random Inputs.* SIAM Journal of Scientific Computing, 2005.

Write a one-page critical response to one of the papers. You should summarize the work in a paragraph and attempt to answer questions such as: How might you apply these ideas to your own research? What are the primary advantages and disadvantages of the methods? How might the methods or concepts be improved?