CAAM 453 · NUMERICAL ANALYSIS I

Problem Set 3

Posted Friday 25 September 2009. Due Monday 5 October 2009.
CAAM 453 students should complete 100 points worth of problems.
CAAM 553 students should complete 125 points worth of problems (4 and 5 strongly recommended).
Students are welcome to attempt more problems if they wish.

1. [25 points]

   Recall that for $\mathbf{A} \in \mathbb{C}^{n\times n}$, the linear system $\mathbf{Ac} = \mathbf{f}$ has a unique solution for any $\mathbf{f}$ provided $\mathrm{Ker}(\mathbf{A}) = \{\mathbf{0}\}$, where $\mathrm{Ker}(\mathbf{A})$ denotes the kernel (null space) of $\mathbf{A}$.

   If the kernel of $\mathbf{A}$ is larger, i.e., if there is a nonzero vector $\mathbf{z} \in \mathrm{Ker}(\mathbf{A})$, then there are two possibilities:

   - If $\mathbf{f} \notin \mathrm{Ran}(\mathbf{A})$, then there is *no solution* $\mathbf{c}$ to the linear system $\mathbf{Ac} = \mathbf{f}$.
   - If $\mathbf{f} \in \mathrm{Ran}(\mathbf{A})$, then there are *infinitely many solutions* to the linear system $\mathbf{Ac} = \mathbf{f}$. In particular, if $\widehat{\mathbf{c}}$ satisfies $\mathbf{A}\widehat{\mathbf{c}} = \mathbf{f}$, then any $\mathbf{c}$ of the form $\mathbf{c} = \widehat{\mathbf{c}} + \gamma\mathbf{z}$ is also a solution, where γ is an arbitrary constant.

   With these facts in mind, please answer the following questions.

   (a) Suppose we wish to construct a polynomial $p_5 \in \mathcal{P}_5$ that interpolates a function $f \in \mathbb{C}^2[-1,1]$ in the following (somewhat unusual) manner: $p_5(-1) = f(-1)$; $p_5'(-1) = f'(-1)$; $p_5(0) = f(0)$; $p_5''(0) = f''(0)$; $p_5(1) = f(1)$; $p_5'(1) = f'(1)$. Write down the linear system to determine the coefficients $c_0, \ldots, c_5$ for p in the monomial basis: $p_5(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4 + c_5x^5$.

   (b) What is the kernel of the matrix $\mathbf{A}$ constructed in part (a)?
   (You may use the MATLAB command `null(A,'r')`.)
   What does your answer imply about the existence and uniqueness of the interpolant p_5?

   (c) Consider the data: $f(-1) = -1$, $f'(-1) = 0$, $f(0) = 1$, $f''(0) = -2$, $f(1) = 3$, $f'(1) = 4$. Show that there are infinitely many choices for the polynomial p_5 that interpolate this data. Plot six of them. (Superimpose all on the same plot.)

2. [25 points]

   The *Hermite interpolant* $h_n \in \mathcal{P}_{2n+1}$ of $f \in C^1[a,b]$ at the points $\{x_j\}_{j=0}^n$ can be written in the form

   $$h_n(x) = \sum_{j=0}^{n} \Big(A_j(x)f(x_j) + B_j(x)f'(x_j)\Big),$$

   where the functions A_j and B_j generalize the Lagrange basis functions:

   $$\begin{aligned} A_j(x) &= \big(1 - 2\,\ell_j'(x_j)(x - x_j)\big)\,\ell_j^2(x) \\ B_j(x) &= (x - x_j)\,\ell_j^2(x), \end{aligned}$$

   with $\ell_j(x) = \prod_{k=0,k\neq j}^{n}(x - x_k)/(x_j - x_k)$.

   (a) Verify that

   $$A_j(x_k) = \begin{cases} 1 & j = k \\ 0 & j \neq k, \end{cases} \qquad A_j'(x_k) = 0, \qquad B_j(x_k) = 0, \qquad B_j'(x_k) = \begin{cases} 1 & j = k \\ 0 & j \neq k. \end{cases}$$

   (b) The above expression for the Hermite interpolating polynomial mimics the *Lagrange form* of the the standard interpolating polynomial. Devise a scheme for constructing Hermite interpolants that generalizes the *Newton form*. What are your new Newton-like basis functions for $\mathcal{P}_{2n+1}$?

3. [25 points]

The one-dimensional interpolation scheme studied in class can be adapted to higher dimensions. For example, suppose we are given a scalar-valued function $f(x, y)$, such as

$$f(x, y) = \mathrm{e}^x \sin y,$$

and wish to construct a function of the form

$$p(x, y) = c_0 + c_1 x + c_2 y + c_3 xy + c_4 x^2 + c_5 y^2$$

that interpolates $f(x, y)$ at (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5).

(a) Set up a linear system $\mathbf{Ac} = \mathbf{f}$ to determine the coefficients $c_0, \ldots, c_5$.

(b) Write a MATLAB code to determine $\mathbf{c}$ when $f(x, y) = \mathrm{e}^x \sin y$ and the (x_j, y_j) pairs take the values listed in the following table.

j	0	1	2	3	4	5
x_j	0	0	1	1	2	2
y_j	0	2	0	2	1	3

Report your value for $\mathbf{c}$.

(c) Plot your model function $p(x, y)$ over $x \in [-1, 3]$, $y \in [-1, 3]$ using MATLAB's `surf` command. Compare this plot to the similar plot for $f(x, y)$, which can be obtained in the following manner.

```
f = inline('exp(x).*sin(y)','x','y');
[xx,yy] = meshgrid(linspace(-1,3,25),linspace(-1,3,25));
zz = f(xx,yy);
figure(1), clf
surf(xx,yy,zz)
```

Please submit plots of both $p(x, y)$ and $f(x, y)$.

4. [25 points]

When we have spoken about norms, projectors, and the like, we have usually been working with vectors and matrices, but these concepts generalize to a much broader setting. In this problem, you will apply these ideas to develop a bound on the accuracy of polynomial interpolation.

Let Π_n denote the linear operator that maps $f \in C[a, b]$ to the polynomial p_n that interpolates f at the distinct points $x_0, \ldots, x_n$, $\{x_j\}_{j=0}^n \subset [a, b]$. In other words, $\Pi_n f = p_n$, where p_n is the unique polynomial of degree n (or less) for which $f(x_j) = p_n(x_j)$ for $j = 0, \ldots, n$.

(a) Explain why Π_n is a *projector*.
(What does $\Pi_n p_n$ equal if p_n is a polynomial of degree n?)

For the rest of this problem, we use the following norm on $g \in C[a, b]$:

$$\|g\|_{L^\infty} = \max_{a \le x \le b} |g(x)|.$$

This norm obeys the three familiar norm axioms (e.g., the triangle inequality). It also induces the operator norm

$$\|\Pi_n\|_{L^\infty} = \max_{f \in C[a,b], f \not\equiv 0} \frac{\|\Pi_n f\|_{L^\infty}}{\|f\|_{L^\infty}} = \max_{\|f\|_{L^\infty}=1} \|\Pi_n f\|_{L^\infty}.$$

(b) Show that if $x_0 = a$ and $x_1 = b$, then $\|\Pi_0\|_{L^\infty} = \|\Pi_1\|_{L^\infty} = 1$.

(c) Recall that we can write the polynomial $p_n = \Pi_n f$ in the Lagrange form

$$\Pi_n f = \sum_{j=0}^{n} f(x_j)\ell_j(x),$$

where ℓ_k denotes the kth Lagrange basis polynomial.

Prove that $\|\Pi_n\|_{L^\infty} = \max_{x\in[a,b]} \sum_{j=0}^{n} |\ell_j(x)|$.

(d) Let p_* denote any polynomial of degree n (e.g., p_* minimizes $\|f - p\|_{L^\infty}$ over all $p \in \mathcal{P}_n$). Prove that $\|f - p_n\|_{L^\infty} \le (1 + \|\Pi_n\|_{L^\infty})\|f - p_*\|_{L^\infty}$.

(e) (Optional) Computationally estimate $\|\Pi_n\|_{L^\infty}$ for $n = 1, \ldots, 20$ with (i) uniformly spaced points $x_j = -1 + 2j/n$ and (ii) Chebyshev points $x_j = \cos(j\pi/n)$ over $[-1, 1]$.

5. [25 points]

Suppose the complex-valued function $f(z)$ of the variable $z \in \mathbb{C}$ is analytic in a region D of the complex plane whose boundary C is a simple closed contour. Furthermore, suppose the interpolation points $x_0, \ldots, x_n$ ($n \ge 1$) and the point x all lie in D.

(a) Let $p_n \in \mathcal{P}_n$ denote the polynomial that interpolates f at $x_0, \ldots, x_n$. For any $x \in D$, confirm the identity

$$f(x) - p_n(x) \;=\; \frac{1}{2\pi \mathrm{i}} \int_C \frac{f(z)}{z - x} \prod_{j=0}^{n} \frac{x - x_j}{z - x_j}\, \mathrm{d}z$$

by computing the integral on the right. (Hint: Consider the poles of the integrand, and use the Cauchy integral formula.)

For the rest of the problem, suppose that the real number x and the interpolation points $x_0, \ldots, x_n$ all lie in the real interval $[a, b]$, and define, for constant $K > 0$,

$$D = \{z \in \mathbb{C} : |z - t| < K \text{ for some } t \in [a, b]\}.$$

(b) Plot (or draw) the boundary C of D for $[a, b] = [-1, 1]$ and $K = 1$.

(c) Show that the length of the contour C is $2(b - a) + 2\pi K$, and that the integral formula in (a) leads to the bound

$$|f(x) - p_n(x)| < \frac{(b - a + \pi K)M}{\pi K}\Big(\frac{b - a}{K}\Big)^{n+1},$$

where M is such that $|f(z)| \le M$ on C.

(d) Deduce that if f is analytic on D for some $K > |b - a|$, then the sequence $\{p_n\}$ converges to f uniformly on $[a, b]$ as $n \to \infty$.

(e) Show that the requirements for the conclusion in (d) *are not satisfied* by Runge's function, $f(x) = 1/(1 + x^2)$ over $[a, b] = [-5, 5]$. For what values of α are the conditions satisfied by this f over $[a, b] = [-\alpha, \alpha]$?

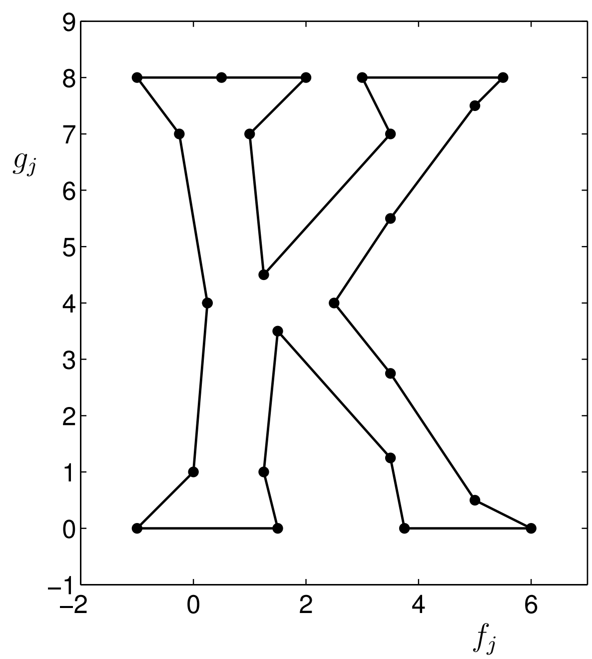
[Süli and Mayers, Problem 6.11]

6. [25 points]

Splines in font design. The font in which this text is set was designed by Donald Knuth using his remarkable METAFONT software. To make shapely letters, the font designer establishes fixed points that guide *Beziér curves*, defined via *Bernstein polynomials*. These curves do not interpolate the guide points, but a similar system based on spline functions, which do interpolate, has also been proposed. Here you will try your hand at spline font design: design a stylized 'K' character using cubic splines with natural boundary conditions. The craft would be the same if you were designing an airplane fuselage or a new sports car.

Consider the following table of data. The (f_j, g_j) values specify the skeleton for our 'K', as shown on the right. Our goal is to replace the straight lines by smooth curves generated from splines.

x_j	f_j	g_j
0	.25	4
1	0	1
2	-1	0
3	1.5	0
4	1.25	1
5	1.5	3.5
6	3.5	1.25
7	3.75	0
8	6	0
9	5	0.5
10	3.5	2.75
11	2.5	4
12	3.5	5.5
13	5	7.5
14	5.5	8
15	3	8
16	3.5	7
17	1.25	4.5
18	1	7
19	2	8
20	0.5	8
21	−1	8
22	−.25	7
23	.25	4



(a) Write a MATLAB routine

```
function S = cBspline(x, x0, h)
```

that computes the value of a natural cubic B-spline at a point $\mathtt{x} \in \mathbb{R}$, given the initial knot $\mathtt{x0} \in \mathbb{R}$ and uniform grid spacing $\mathtt{h}$, i.e., $f_j = x_0 + j\mathtt{h}$.

(b) Using your code from part (a), or otherwise, construct two *natural* cubic splines, one, called $S_1(t)$, interpolating the (x_j, f_j) values, the other, called $S_2(t)$, interpolating (x_j, g_j). (Each spline should be the linear combination of $n + 3 = 26$ B-splines. Further details will be provided in the lecture and lecture notes; the variables f_j and g_j are defined in the MATLAB file `Kdata.m` on the class website.)

(c) Produce a plot showing $S_1(t)$ and $S_2(t)$ for $t \in [0, 23]$, along with the points (x_j, f_j) and (x_j, g_j), to verify that your splines interpolate the data as desired.

(d) In a separate figure, plot $(S_1(t), S_2(t))$ for $t \in [0, 23]$. You should obtain a picture like the skeleton above, but with the straight lines replaced by more interesting curves. Superimpose the (f_j, g_j) points to verify that your splines interpolate the data points.