Assignment2 Key for CSCE/Math 441

Thomas Shores
Department of Mathematics
University of Nebraska
Fall 2009

Points: 35
Due: Friday, October 2

Exercise 1.5. Let $\mathcal{A}$ be the set of real continuous functions on the interval $[a,b]$ that are composed of straight line segments. Hence $\mathcal{A}$ is a subspace of $C[a,b]$. Prove that, for any $f \in C[a,b]$ and for any $\epsilon > 0$, there exists an element $a \in \mathcal{A}$ such that $\|f - z\|_\infty < \epsilon$.

Solution. (6) Let $\{p_n(x)\}$ be any sequence of piecewise linear polynomials that interpolates $f(x)$on a grid of knots such that the norm of the grid (size of the largest subinterval) tends to zero. If $\|f - p_n\|_\infty \geq \epsilon$ for all n, then there is a sequence of points in $[a,b]$ for which the inequality holds. Choose a subsequence that converges to a point in the interval, say x^*. Then there must be a sequence of knots that converges to x^*. The corresponding subsequence of piecewise polynomials interpolates $f(x)$ at those knots, so there is a sequence of points converging to $x^*$for which $f(x)$ is no closer than ϵ to $f(x^*)$. This contradicts the continuity of f at x^*, so there must be an index n such that $\|f - z\|_\infty < \epsilon$.

Exercise 2.7. Let the set $\mathcal{A}$ in $C[-1,1]$ consist of continuous functions consisting of one or two straight line segements. Show there is more than one best approximation from $\mathcal{A}$ to the function $f(x) = x^3$ in $C[-1,1]$ with respect to the infinity norm.

Solution. (6) The function $f(x) = x^3$ is skew-symmetric, so if there is a best approximation $p(x)$ that is not skew symmetric, its reflection across the x-axis, $-p(-x)$ is also a best approximation. If not, then the best approximation is skew symmetric and and any knot would have to be at the origin. But then the best approximation must pass through the origin and be a straight line (else replace the further segment by the reflection of the better segment). But one can do better than this best approximation by moving the knot up the y-axis slightly and moving the knot at $x = 1$ up slightly to get an approximation at least as good as the straight line.

Exercise 3.9. Let s be the cubic spline function

$$s(x) = x^3 - 4(x-1)_+^3 + 6(x-2)_+^3 - 4(x-3)_+^3 + (x-4)_+^3$$

on the interval $0 \leq x \leq 100$. Show that s is identically zero if $x \geq 4$, but severe cancellation occurs if $s(100)$ is evaluated from definition of s.

Solution. (5) Evaluate the expression for $s(x)$ for $x \geq 4$, so that all of the "+" signs disappear, along with its derivatives:

$$s(x) = x^3 - 4(x-1)^3 + 6(x-2)^3 - 4(x-3)^3 + (x-4)^3 .$$

Now you can either expand this polynomial or simply evaluate it at any four points, say $x = 1, 2, 3,4$, to obtain that $0 = s(1) = s(2) = s(3) = s(4) = 0$. But the only polynomial of degree at most 4 that interpolates these points is $s(x) = 0$.

The choice by the author of $x = 100$ was a poor one, since you can do this exactly in 6-digit arithmetic to get 0. A much better choice would be something like $99 + 18/19$, where you would see that the computed answer is nonzero and something like $-6.9849e{-}10$, which is certainly not 0.

Exercise 4.8. Suppose that one has to calculate $p(x)$ from Equations (4.7) and (4.3) for many millions of value of x, where n is about 20. Show that, by calculating in advance some auxillary quantities that depend on the data polnts $\{x_i\}_{i=0}^{n}$ and the function values $\{f(x_i)\}_{i=0}^{n}$, the number of computer operations in each evaluation of $p(x)$ can be reduced to a small multiple of n.

Solution. (6) Let's say that k evaluations of $p(x)$ will be performed with $k \gg n$. Here is the strategy: Precompute the quantities

$$c_j = \frac{f(x_j)}{\prod_{0 \le k \le n,\, k \ne j} (x_j - x_k)}.$$

This will require $(n+1)(2n+1)$ flops.

Now for the evaluation of $p(x)$ for a given x:

(1) Check to see if $x = x_j$ for some j. This will cost at most $n+1$ flops, since a comparison requires the test $x - x_j == 0$. If equality, set $p(x) = f(x_j)$ and output $p(x)$. Otherwise
(2) Evaluate the the terms $d_k = (x - x_k)$, $k = 0, \ldots, n$, at a cost of $n+1$ flops.
(3) Evaluate the product

$$L = \prod_{0 \le k \le n} d_k$$

at a cost of $n+1$ flops.
(4) Compute

$$p(x) = \sum_{j=0}^{n} c_j \frac{L}{d_j}$$

at a cost of $3(n+1)$ flops.
(5) Output $p(x)$.

Total cost of evaluating $p(x)$ is at most $6(n+1)$ flops.

Exercise N1. With the assumptions and notation of Theorem 3.1 in the text, if one has a good estimate of of $d^*(f)$, one can find a lower bound for $\|X\|$. Show this lower bound. Then assume that Chebyshev interpolation gives a good estimate to the best approximation to $f \in C[-5,5]$ from $\mathcal{B} = \mathcal{P}_{20}$ and estimate $d^*(f)$ for $f = 1/(1+x^2)$ and one other function of your choice. Use these to give lower bounds for $\|X\|$, where X is the interpolation operator using 21 equally spaced interpolation points. (This corrects a typo in the original problem – X should map into the interpolating space $\mathcal{B} = \mathcal{P}_{20}$, so we need 21, not 12, interpolating points.)

Solution. (6) According to Theorem 3.1 we have

$$\|f - X(f)\| \le \{1 + \|X\|\}\, d^*(f).$$

Subtract $d^*(f)$ from both sides and then divide by this positive number to obtain

$$\frac{\|f - X(f)\|}{d^*(f)} - 1 \le \|X\|.$$

Now use Matlab to obtain the numerator of the left-hand side via this transcript:

```
octave:2> xnodes = -5:0.01:5;
octave:3> xinterp = linspace(-5,5,21);
octave:4> polyinterp = vander(xinterp)\(1./(1+xinterp'.^2));
octave:5> nrmfmX = norm(1./(1+xnodes.^2)-polyval(polyinterp,xnodes),inf) nrmfmX
= 59.768
octave:6> xnodes = cos((2*(20:-1:0)+1)*pi/(2*21));
octave:7> polyinterp = vander(xinterp)\(1./(1+xinterp'.^2));
octave:8> dstar = norm(1./(1+xnodes.^2)-polyval(polyinterp,xnodes),inf) dstar
= 0.0032753
octave:9> nrmfmX/dstar - 1 ans = 1.8247e+04
```

So we estimate that the norm of the equally spaced interpolation operator is at least 18247.

Exercise N2. Write a Matlab function with calling form Lnorm(a,b,x), where inputs a and b are left and right endpoints of an interval containing the coordinates of the input vector x, and the output is an approximation to the norm of the Lagrange interpolation operator with interpolation points in the vector x.

Solution. (6) We simply have to implement Theorem 4.2, which says that

$$\|X\| = \max_{a\leq x\leq b}\sum_{k=0}^{n}|l_k(x)|.$$

Here is a straightforward implementation that uses no Matlab shortcuts nor the more sophisticated algorithm of Exercise 4.8.

```
function retval = Lnorm(a,b,x)
% usage:  y = Lnorm(a,b,x)
% description:  given input of left endpoint a, right
% endpoint b and vector of interpolation nodes x in
% increasing order between a and b, return the approximate
% infinity norm of the Lagrange interpolation operator
% with interpolation points specified by x.

x = x(:); % make x a column
n = length(x);
dx = min(abs(diff(x)))/100; % choose a spacing on [a,b]
m = round((b-a)/dx);
xnodes = linspace(a,b,m); % row of sampling nodes for norm
Lx = ones(n,m);
for j = 1:n
  for k = [1:j-1,j+1:n]
    Lx(j,:)  = Lx(j,:).*(xnodes-x(k))/(x(j)-x(k));
  end
end
retval = sum(abs(Lx));
```

Check this against the table on page 42 of text and get the same results.