Numerical Analysis Homework #4

due 2020 APR 14, 9:50 a.m.

1 Assignments

Caution:

- To get full credit, *you must write down sufficient intermediate steps*, only giving the final answer earns you no credit!
- Please make sure that your handwriting is recognizable, otherwise you only get partial credit for the recognizable part.

I. Consider $s \in \mathbb{S}_3^2$ on $[0,2]$:

$$s(x) = \begin{cases} p(x) & \text{if } x \in [0,1], \\ (2-x)^3 & \text{if } x \in [1,2]. \end{cases}$$

Determine $p \in \mathbb{P}_3$ such that $s(0) = 0$. Is $s(x)$ a natural cubic spline?

II. Given $f_i = f(x_i)$ of some scalar function at points $a = x_1 < x_2 < \cdots < x_n = b$, we consider interpolating f on $[a,b]$ with a quadratic spline $s \in \mathbb{S}_2^1$.

(a) Why an additional condition is needed to determine s uniquely?

(b) Define $m_i = s'(x_i)$ and $p_i = s|_{[x_i, x_{i+1}]}$. Determine p_i in terms of f_i, f_{i+1}, and m_i for $i = 1, 2, \ldots, n-1$.

(c) Suppose $m_1 = f'(a)$ is given. Show how $m_2, m_3, \ldots, m_{n-1}$ can be computed.

III. Let $s_1(x) = 1 + c(x+1)^3$ where $x \in [-1,0]$ and $c \in \mathbb{R}$. Determine $s_2(x)$ on $[0,1]$ such that

$$s(x) = \begin{cases} s_1(x) & \text{if } x \in [-1,0], \\ s_2(x) & \text{if } x \in [0,1] \end{cases}$$

is a natural cubic spline on $[-1,1]$ with knots $-1, 0, 1$. How must c be chosen if one wants $s(1) = -1$?

IV. Consider $f(x) = \cos\left(\frac{\pi}{2}x\right)$ with $x \in [-1,1]$.

(a) Determine the natural cubic spline interpolant to f on knots $-1, 0, 1$.

(b) As discussed in the class, natural cubic splines have the minimal total bending energy. Verify this by taking $g(x)$ be (i) the quadratic polynomial that interpolates f at $-1, 0, 1$, and (ii) $f(x)$.

V. The quadratic B-spline $B_i^2(x)$.

(a) Derive the same explicit expression of $B_i^2(x)$ as that in the notes from the recursive Definition of B-splines and the hat function.

(b) Verify that $\frac{\mathrm{d}}{\mathrm{d}x} B_i^2(x)$ is continuous at t_i and t_{i+1}.

(c) Show that only one $x^* \in (t_{i-1}, t_{i+1})$ satisfies $\frac{\mathrm{d}}{\mathrm{d}x} B_i^2(x^*) = 0$. Express x^* in terms of the knots within the interval of support.

(d) Consequently, show $B_i^2(x) \in [0,1)$.

(e) Plot $B_1^2(x)$ for $t_i = i$.

VI. We proved a theorem on constructing B-splines from truncated power functions,

$$B_i^n(x) = (t_{i+n} - t_{i-1}) \cdot [t_{i-1}, \ldots, t_{i+n}](t-x)_+^n. \quad (1)$$

Verify it algebraically for the case of $n = 2$, i.e.

$$(t_{i+2} - t_{i-1})[t_{i-1}, t_i, t_{i+1}, t_{i+2}](t-x)_+^2 = B_i^2.$$

The above six problems weigh 4, 9, 4, 8, 10, and 5 points, respectively.

2 `C++` programming

(a) Write a `C++` subroutine for cubic-spline interpolation of the function

$$f(x) = \frac{1}{1+25x^2}$$

on evenly spaced nodes within the interval $[-1,1]$ with $N = 6, 11, 21, 41, 81$. Compute the max-norm of interpolation errors at the nodes for each N and report the errors and convergence rates with respect to the number of subintervals.

Your algorithm should follow the example of interpolating the natural logarithm in the notes and your program must use an implementation of `lapack`.

Plot the interpolating spline against the exact function to observe that spline interpolation is free of the wide oscillations in the Runge phenomenon.

(b) (*) Write a `Matlab` subroutine to illustrate (1) by plotting the truncated power functions and building a table of divided difference where the entries are figures instead of numbers. See the hints in Section 4.

The two programming assignments weigh 14 and 6 points, respectively. Problem (b) is for extra credit. The total number of points of this homework is thus 54(+6).

3 Extra credits

Additional 10% credits will be given to you if you typeset your solutions in LaTeX. You are welcome to use the LaTeX template available on my webpage. You can also get partial extra credit for typesetting solutions of *some* problems.

Note: If you choose to typeset your solutions in LaTeX, you still need to turn in a hard copy in class. In addition, please upload your latex source (.tex), supporting files, and `C++` program in a single zip file (**format**: `YourName_Homework1.zip`) to the course email `NumApproximation@163.com`.

4 Hints on the extra-credit assignment

For simplicity, take $t_{i-1} = 1$ and $t_{i+j} = t_i + j$. You only need to worry about the cases of $n = 1, 2$.

Your matlab program may use the following signature.

```
function plotTruncatedPowerFunc2Bsplines(n)
% Illustrate the relation between truncated power functions and B-splines.
%
% INPUT
%   n : the degree of B-splines.
% Preconditions:
%   CHECK: n==1 or n==2, i.e. the code only works for linear and quadratic cases.
```

The pictures you generated for $n = 1$ should be the same as those in the notes. Those for $n = 2$ should be the same as follows.