COL726 Problem set 3: Polynomials and interpolation

January 16, 2017

1 First some more rounding and chopping

1. Consider the numbers

$$\begin{aligned} x_1 &= 0.1234 \times 10^1 \\ x_2 &= 0.3429 \times 10^0 \\ x_3 &= 0.1289 \times 10^{-1} \\ x_4 &= 0.9895 \times 10^{-3} \\ x_5 &= 0.9763 \times 10^{-5} \end{aligned}$$

   Add these numbers using four-decimal-digit chopped floating point arithmetic in both forward and reverse. Which is more accurate? Why?

2. Suggest methods for evaluating each of

   (a) $e^x \simeq 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$

   (b) $\cos x \simeq 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \dots$

   (c) $\sin x \simeq x - \frac{x^3}{3!} + \frac{x^5}{5!} + \dots$

   for $x = 25$ up to 12 digits of accuracy. Pay attention to both truncation and round-off errors. Try out with Matlab programs.

3. Suppose you have to evaluate

$$F(x) = x \sin x / (1 - \cos x)$$

   near $x = 0$. Do you foresee any problems? Suggest a method to overcome the problem. Again try out on Matlab.

2 Interpolation

1. Derive the *divided difference* formula again (not to be done in the tute class).

2. Consider the following algorithm for computing $d_k = f[x_1, \ldots, x_k]$, $k = 1, 2, \ldots, n+1$ given the data points $x_1, x_2, \ldots, x_n$ and function values $f_1, f_2, \ldots, f_n$.

   Algorithm:

   Input $\{x_1, x_2, \ldots, x_n, f_1, f_2, \ldots, f_n\}$
   for $j = 1, 2, \ldots, n+1$
   $\quad d_j \leftarrow f_j$
   $\quad$**for** $k = 1, 2, \ldots, n$
   $\quad\quad$**for** $j = n+1, n, \ldots, k+1$
   $\quad\quad\quad d_j \leftarrow (d_j - d_{j-1})/(x_j - x_{j-k})$
   Output $\{d_1, d_2, \ldots, d_{n+1}\}$

   Prove the correctness and determine the time and space requirements.

3. Prove the following result.
   If $p(x)$ is the interpolating polynomial which agrees with $f(x)$ at $n+1$ points in $[a, b]$ and if f is $n+1$ times continuously differentiable in $[a, b]$ then for any $\bar{x} \in [a, b]$ there is a value $\eta \in [a, b]$ such that the *truncation error* is given by

   $$E_T(\bar{x}) = f(\bar{x}) - p(\bar{x}) = [f^{(n+1)}(\eta)/(n+1)!](\bar{x} - x_1)((\bar{x} - x_2) \ldots (\bar{x} - x_{n+1})$$

   What conclusion can you draw about extrapolation/truncation errors?

4. Show that under the conditions of the previous problem, and with P_k the polynomial that interpolates f at $x_1, x_2, \ldots, x_{k+1}$, for $k = 1, 2, \ldots n$, the difference

   $$P_{k+1}(\bar{x}) - P_k(\bar{x})$$

   is an estimate of the truncation error in $P_k(\bar{x})$. This estimate is usable whenever $f^{(k+1)}(x)$ does not change greatly in the interval containing $x_1, \ldots, x_{k+2}$ and $\bar{x}$.

5. Consider Horner's method for evaluating a polynomial

Input $\{n, a_0, a_1, \ldots, a_n\}$
Input x
$p \leftarrow a_n$
for $k = n-1, n-2, \ldots, 0$
  $p \leftarrow x * p + a_k$
Output $\{x, p\}$

Show that the **backward error estimate** is given by

$$\hat{p} = \hat{a}_n x^n + \ldots + \hat{a}_1 x + \hat{a}_0$$

where,

$$\hat{a}_k = \begin{cases} a_k\langle 2n\rangle & k = n \\ a_k\langle 2k+1\rangle & k = n-1, \ldots, 0 \end{cases}$$

Conclude that if $nr\mu \leq 0.1$, then the relative error bound can be obtained as

$$\frac{\mid \hat{a}_k - a_k \mid}{\mid a_k \mid} \leq \begin{cases} 2n\mu' & k = n \\ (2k+1)\mu' & k = n-1, \ldots, 0 \end{cases}$$

Also, show that the forward error estimate is given by

$$\mid \hat{p} - p(x) \mid = [2n \mid a_n x^n \mid + (2n-1) \mid a_{n-1} x^{n-1} \mid + 3 \mid a_1 x] \mid + \mid a_0 \mid]\mu'$$

6. Given Algorithm in 2 for computing the divided difference coefficients, an algorithm for evaluating Newton's formula can be given as

Input $\{x_1, \ldots, x_{n+1}, d_1, \ldots, d_{n+1}\}$
Input x
$p \leftarrow d_{n+1}$
for $i = n, n-1, \ldots, 1$
  $p \leftarrow p * (x - x_i) + d_i$
Output $\{x, p\}$

Show that the Algorithms in 2 and 6 evaluate a polynomial $\hat{p}$ that interpolates a function $\hat{f}$ at $x_1, x_2, \ldots, x_{n+1}$. For all x in an interval of length l that contain $x_1, x_2, \ldots, x_{n+1}$, the following bound holds:

$$\mid f(x) - \hat{f}(x) \mid \leq 9(n^3 + n^2 + 1)(l^n/m^n)\bar{f}\mu'$$

where,

$$\bar{f} = max\{\mid f(x_i) \mid: i = 1, 2, \ldots, n + 1\}$$

and

$$m = min\{\mid x_i - x_j \mid: i, j = 1, 2, \ldots, n + 1, i \neq j\}$$

[**Note:** This one will be hard! Perhaps you should split it up into several sub-problems?]

7. Conclude from the previous problems that the rounding errors in the two algorithms are equivalent to a change in the finction f. This change will be small provided that

   (a) n is not too large

   (b) the interval length is not large

   (c) the data points $x_1, x_2, \ldots, x_{n+1}$ are not too close together, and

   (d) the maximum value $\bar{f}$ is not much larger than any other $f(x)$ for x in the interval.

8. Suppose that each of the following functions has been tabulated at $x_1 = 0$, $x_2 = 0.1$, $x_3 = 2$ and $x_4 = 0.3$. Estimate the truncation error at $\bar{x} = 0.15$ for the polynomial of degree 3 that interpolates at the given tabulated points.

   (a) $f(x) = \sin x$

   (b) $f(x) = 2x^3$

   (c) $f(x) = 1/(x + 1)$

9. Use Algorithm 6 to approximate $f(-0.5)$ where f is the function $f(x) = 1/(1 + 25x^2)$. Print out a table of $(n, f(-0.5), \epsilon)$ (ϵ being the absolute error) for $n = 1, 2, 3, 4$. Also print out the truncation error estimates $P_{n+1} - P_n$. Verify that the derivatives of this function change so rapidly that these error estimates are not so accurate.

10. Suppose f is a function that has been tabulated at $x = 0, 0.1, 0.2, 0.3, 0.4, \ldots$ and suppose $\mid f^{(k)}(x) \mid \leq k, k = 0, 1, 2, \ldots$. What is the best value of n to use for interpolating f at $\bar{x} = 0.25$? That is, what value of n will result in the sum of truncation error and rounding error to be as small as possible?