

Math 128a, Section 3 — Solutions to Problem Set 3

(2) We have three ways to represent the linear polynomial  $p(x)$  which interpolates values  $y_1$  at  $x_1$  and  $y_2$  at  $x_2$ : (a) the power form, (b) the Newton form, and (c) the Lagrange form. Carry out a backward error analysis of each form in floating-point arithmetic with machine precision  $\epsilon$ : in other words, *either* show that the computed result  $\hat{p}(x)$  is the exact value of the linear polynomial interpolating  $\hat{y}_1$  at  $\hat{x}_1$  and  $\hat{y}_2$  at  $\hat{x}_2$  and bound the relative error in the  $\hat{x}$ 's and  $\hat{y}$ 's by small multiples of  $\epsilon$ , *or* show that  $\hat{p}(x)$  is not in general the exact value of any such linear interpolating polynomial.

**Solution:** We will work these in reverse order, from easiest to hardest.

(c) The Lagrange form: The floating-point calculation  $\hat{p}(x)$  takes the following form:

$$\hat{p}(x) = \left[ \frac{(x-x_2)(1+\delta_1)}{(x_1-x_2)(1+\delta_2)}(1+\delta_3)y_1(1+\delta_4) + \frac{(x-x_1)(1+\delta_5)}{(x_2-x_1)(1+\delta_6)}(1+\delta_7)y_2(1+\delta_8) \right] (1+\delta_9).$$

By making the substitutions

$$(1+\Delta_5) = \frac{(1+\delta_1)}{(1+\delta_2)}(1+\delta_3)(1+\delta_4)(1+\delta_9)$$

and

$$(1+\Delta'_5) = \frac{(1+\delta_5)}{(1+\delta_6)}(1+\delta_7)(1+\delta_8)(1+\delta_9),$$

this simplifies to

$$\hat{p}(x) = \frac{x-x_2}{x_1-x_2}y_1(1+\Delta_5) + \frac{x-x_1}{x_2-x_1}y_2(1+\Delta'_5),$$

which is the exact value at  $x$  of the polynomial interpolating  $y_1(1+\Delta_5)$  at  $x_1$  and  $y_2(1+\Delta'_5)$  at  $x_2$ . Each of these relative errors is bounded by  $10\epsilon$ .

(b) The Newton form: We saw in class that there was some difficulty in assigning the errors consistently to the input variables, but the problem asks for either a solution, or a proof that none exists. What we need, then, is some specific choice of inputs, and a proof that no small perturbation of those inputs produces an error as large as the error in  $\hat{p}(x)$ , the floating-point approximation of the Newton form interpolating polynomial. One approach to such a proof relies on the bad behavior of floating-point arithmetic when adding numbers of opposite sign to try to compute 0. To be specific, let  $x_1 = y_1 = -1$  and  $x_2 = y_2 = 0$ , and interpolate at the point  $x = 0$ . The floating-point approximation is

$$\hat{p}(x) = \left[ y_1 + \frac{(y_2 - y_1)(1 + \delta_1)}{(x_2 - x_1)(1 + \delta_2)}(1 + \delta_3)(x - x_1)(1 + \delta_4)(1 + \delta_5) \right] (1 + \delta_6).$$

By making the substitution

$$(1+\Delta_5) = \frac{(1+\delta_1)}{(1+\delta_2)}(1+\delta_3)(1+\delta_4)(1+\delta_5)$$

and plugging in the data points we have chosen, this simplifies to

$$\begin{aligned} \hat{p}(0) &= [-1 + (1 + \Delta_5)](1 + \delta_6) \\ &= \Delta_5(1 + \delta_6) \\ &= \Delta_5 + O(\epsilon^2). \end{aligned}$$

Is this the same as the exact value on approximate inputs? Let us define a new linear polynomial  $\tilde{p}(x)$  which interpolates the perturbed data

$$\begin{aligned}\hat{y}_1 &= y_1(1 + \varepsilon_1), \\ \hat{y}_2 &= y_2(1 + \varepsilon_2), \\ \hat{x}_1 &= x_1(1 + \varepsilon_3), \text{ and} \\ \hat{x}_2 &= x_2(1 + \varepsilon_4),\end{aligned}$$

and evaluate it at the point  $\hat{x} = x(1 + \varepsilon_5)$ . The resulting calculation

$$\tilde{p}(\hat{x}) = -(1 + \varepsilon_1) + \frac{0(1 + \varepsilon_2) + (1 + \varepsilon_1)}{0(1 + \varepsilon_4) + (1 + \varepsilon_3)}(0(1 + \varepsilon_5) + (1 + \varepsilon_3))$$

simplifies to exactly 0, showing that the floating point error, which might be as large as  $5\epsilon$ , cannot possibly be attributed to a small relative error in the input data.

Note: In actuality,  $\Delta_5(1 + \delta_6) = 0$  for the input data we have chosen, since floating point calculations involving only a few 1's and 0's are exact. If this bothers you, play around with numbers close to  $-1$  and  $0$  until you come up with a genuine counterexample.

(a) The power form (also called Van der Monde): We have seen in a previous homework assignment that this form behaves badly when the size of the  $x_i$ 's is large compared to their difference, so it is reasonable to try to make backwards error analysis fail by setting  $x_1 = 1$  and  $x_2 = 1 + t$ , where  $t$  is some small number, but larger than  $\epsilon$ . It is convenient to use  $t = \sqrt{\epsilon}$ , so that  $O(\epsilon) = O(t^2)$ . We will set  $y_1 = y_2 = t$ , so the correct interpolant is the constant function  $p(x) = t$ , and then evaluate the floating-point approximation  $\hat{p}(x)$  of the power form interpolant at the value  $x = 1$ . The resulting mess

$$\hat{p}(x) = \left[ \begin{array}{l} \frac{(y_1 x_2(1+\delta_1) - y_2 x_1(1+\delta_2))(1+\delta_3)}{(x_2 - x_1)(1+\delta_4)}(1 + \delta_5) \\ + \frac{(y_2 - y_1)(1+\delta_6)}{(x_2 - x_1)(1+\delta_4)}(1 + \delta_7)x(1 + \delta_8) \end{array} \right] (1 + \delta_9)$$

simplifies to

$$\begin{aligned}\hat{p}(1) &= \frac{t(1+t)(1+\delta_1) - t(1+\delta_2)}{t}(1 + \Delta_4) + 0 \\ &= (\delta_1 - \delta_2 + t(1 + \delta_1))(1 + \Delta_4) \\ &= p(1) + (\delta_1 - \delta_2) + O(t^3).\end{aligned}$$

This is supposed to equal  $\tilde{p}(\hat{x})$ , the exact interpolant on perturbed data. (Since the exact interpolant is the same using any method, we use the Newton form which we have already worked out.)

$$\tilde{p}(\hat{x}) = t(1 + \varepsilon_1) + \frac{t(1 + \varepsilon_2) - t(1 + \varepsilon_1)}{(1 + t)(1 + \varepsilon_4) - (1 + \varepsilon_3)}((1 + \varepsilon_5) - (1 + \varepsilon_3))$$

simplifies to

$$\begin{aligned}\tilde{p}(1 + \varepsilon_5) &= t + t\varepsilon_1 + \left( \frac{t}{t + \varepsilon_4 - \varepsilon_3 + t\varepsilon_4} \right) (\varepsilon_2 - \varepsilon_3)(\varepsilon_5 - \varepsilon_3) \\ &= t + O(t^3) + \left( \frac{t}{t + O(t^2)} \right) (O(t^4)) \\ &= p(1) + O(t^3).\end{aligned}$$

This shows that perturbing these input data by a relative error of size  $\epsilon = O(t^2)$  produces an absolute error in the output of size at most  $O(t^3)$ , which is not enough to account for the floating-point error of size  $(\delta_1 - \delta_2) = O(t^2)$ .