GROUP HOMEWORK 7, CPSC 303, SPRING 2024

JOEL FRIEDMAN

Please note:

(1) You must justify all answers; no credit is given for a correct answer without justification.
(2) Proofs should be written out formally.
(3) You do not have to use LaTeX for homework, but **homework that is too difficult to read will not be graded.**
(4) You may work together on homework in groups of up to four, **but you must submit a single homework as a group submission under Gradescope.**
(5) At times we may only grade part of the homework set. The number of points per problem (at times indicated) may be changed.

For this problem set, "the handout" refers to the article "CPSC 303: What the Condition Number Does and Does Not Tell Us."

(1) (0 to -8 points) Who are your group members? Please print if writing by hand. [See (4) above.]

(2) The point of this exercise is to compare monomial interpolation (Section 10.2 of [A&G]) with Lagrange interpolation (Section 10.3).
   (a) Let $p(x) = c_0 + c_1 x$ be the unique polynomial of degree at most 1 such that
   $$p(2) = \sqrt{2}, \quad p(2.01) = \sqrt{3},$$
   In exact arithmetic,
   $$p(2.005) = \frac{\sqrt{2} + \sqrt{3}}{2},$$
   since 2.005 is the midpoint between 2 and 2.01. Hence one can also write:
   $$p(2.005) = c_0 + c_1(2.005).$$

Research supported in part by an NSERC grant.

Solve for c_0, c_1, using the Vandermonde matrix and the formula derived in class (see also page 300 of [A&G]). [Hint: you may find the following MATLAB commands useful:

```
A = fliplr( vander([2 2.01]))
y = [sqrt(2);sqrt(3)]
c = A^(-1)*y
trueVal = (y(1)+y(2))/2
monoVal = c(1) + c(2) * 2.005
```

What does MATLAB report for the absolute error in

$$\big(c_0 + c_1(2.005)\big)$$

as an approximation for

$$\frac{\sqrt{2}+\sqrt{3}}{2}$$

(in absolute value)? What about the relative error?

(b) Same question, where

$$p(2) = \sqrt{2}, \quad p(2+10^{-6}) = \sqrt{3},$$

and you want to compute $p(2+10^{-6}/2)$. [Hint: Recall 5×10^{-7} in MATLAB notation is `5e-7` or `5.0e-7`.]

(c) Same question, where

$$p(2) = \sqrt{2}, \quad p(2+10^{-10}) = \sqrt{3},$$

and you want to compute $p(2+10^{-10}/2)$. [Hint: Recall 5×10^{-11} in MATLAB notation is `5e-11` or `5.0e-11`.]

(d) What is the L^p-condition number of A in part (c) for $p=\infty$? Do this FIRST by typing `cond(A,Inf)`, and SECOND check this by examining the values of A and A^{-1} and using the formula

$$\left\| \begin{bmatrix} a & b \\ c & d \end{bmatrix} \right\|_\infty = \max\big(|a|+|b|, |c|+|d|\big).$$

(i.e., given in class and proven on the previous homework).

(e) Double precision for standard numbers has a relative precision error after rounding of roughly $2^{-53} = 1.1102\ldots\times 10^{-16}$ in the worst case.[1] If you multiply this by the condition number of A (and this is only a very rough indication of the precision you'd expect to lose in $\mathbf{c}\ldots$), what do you get?

(f) Now use the Lagrange formula for $p(x)$ in part (c):

$$p(x) = y_0 \frac{x-x_1}{x_0-x_1} + y_1 \frac{x-x_0}{x_1-x_0}$$

to calculate $p(2+10^{-10}/2)$; what are the absolute and relative errors in this calculation compared with the true value?

[1] This reason is that a true value of $1+2^{-53}$ has to stored as either 1 or $1+2^{-52}$ or a number farther away, resulting in a relative error of $2^{-53}/(1+2^{-53})$; of course, in the best case the relative error is 0.

(g) Now use the Lagrange formula for $p(x)$ in part (c):

$$p(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}$$

to calculate $p(2 + 10^{-10}/3)$, and compute the true value of

$$p(2 + 10^{-10}/3) = (2/3)\sqrt{2} + (1/3)\sqrt{3}$$

via the MATLAB line `(2/3)*sqrt(2)+(1/3)*sqrt(3)`. What are the absolute and relative errors in the Lagrange formula computation as compared with the true value?

(3) (a) Let $p(x) = c_0 + c_1 x + c_2 x^2$ be the unique polynomial of degree at most 2 such that

$$p(2) = \sqrt{2}, \quad p(2.01) = \sqrt{3}, \quad p(2.02) = \sqrt{5}.$$

Let

$$\alpha_2 = p(2.005)$$

(we will explain the subscript 2 in the notation α_2 below). Approximate α as follows: first solve for $\mathbf{c} = (c_0, c_1, c_2)$ as $\mathbf{c} = A^{-1}\mathbf{y}$ using the formula derived in class (see also page 300 of [A&G]) $A\mathbf{c} = \mathbf{y}$ where $\mathbf{y} = (y_0, y_1, y_2)$ and A is a Vandermonde matrix.

(i) What value do you get for α_2? Report this as a base 10 number $1.d_1d_2d_3d_4d_5d_6d_7\ldots$ (so drop the remaining digits, rather than round up/down, and make sure you type `format long` into MATLAB if you aren't seeing enough decimal places).

(ii) What does MATLAB report for the ∞-condition number of A? (Here a few decimal places suffice, e.g., $5.37\ldots \times 10^5$.)

You may find some of the following lines of MATLAB code helpful:

```
help format
format long
A = fliplr( vander([2, 2.01, 2.02]))
y = [sqrt(2);sqrt(3);sqrt(5)]
c = A^(-1)*y
%     For the result below, note that MATLAB indexing
%     begins with 1, not 0
monoVal = c(1) + c(2) * 2.005 + c(3) * (2.005)^2
cond(A,Inf)
```

(b) Let $q(x)$ be the unique polynomial of degree at most 2 such that

$$q(2) = \sqrt{2}, \quad q(2 + 10^{-6}) = \sqrt{3}, \quad q(2 + 10^{-6} \cdot 2) = \sqrt{5}.$$

Let

$$\alpha_6 = q(2 + 10^{-6}/2).$$

Approximate α_6 in the same way as you did α_2 in part (a).

(i) What value do you get for α_6? Report this as a base 10 number $1.d_1d_2d_3d_4d_5d_6d_7\ldots$ (so drop the remaining digits, rather than round up/down, and make sure you type `format long` into MATLAB if you aren't seeing enough decimal places).

(ii) What does MATLAB report for the ∞-condition number of A?

(c) Same question in part (b), with $q(x), 10^{-6}, \alpha_6$ respectively replaced with $r(x), 10^{-7}, \alpha_7$.
(d) Same question in part (b), with $q(x), 10^{-6}, \alpha_6$ respectively replaced with $s(x), 10^{-8}, \alpha_8$.
(e) Let p be the polynomial in part (a), and q that in part (b). Show that $f(y) = p(2 + y10^{-2}) - q(2 + y10^{-6})$ is a polynomial in y of degree 2 such that $f(y) = 0$ for $y = 0, 1, 2$.
(f) Use the previous part to show that (in an exact computation) $\alpha_2 = \alpha_6$.
(g) Use the ideas of the two previous parts to argue that in exact computations, $\alpha_6 = \alpha_7$.
(h) Now use the Lagrange formula for quadratic polynomials,

$$p(x) = y_0 \frac{x - x_1}{x_0 - x_1} \frac{x - x_2}{x_0 - x_2} + y_1 \frac{x - x_0}{x_1 - x_0} \frac{x - x_2}{x_1 - x_2} + y_2 \frac{x - x_0}{x_2 - x_0} \frac{x - x_1}{x_2 - x_1}$$

to calculate $\alpha_2, \alpha_7, \alpha_8$ and report all the decimal places that MATLAB's `format long` reports. You may find the following MATLAB lines helpful for the α_2 calculation (note: an earlier version had 2's instead of 10's below):

```
n=2
x0 = 2 ; x1 = 2 + 10^(-n) ; x2 = 2 + 10^(-n) * 2;
x = 2 + 10^(-n)/2;
y0 = sqrt(2); y1 = sqrt(3); y2 = sqrt(5);
L0 = (x-x1) * (x-x2) / ( (x0-x1) * (x0-x2) );
L1 = (x-x0) * (x-x2) / ( (x1-x0) * (x1-x2) );
L2 = (x-x0) * (x-x1) / ( (x2-x0) * (x2-x1) );
p = y0 * L0 + y1 * L1 + y2 * L2
```

For the next problem(s), recall that if A is a square, invertible matrix, and if $A\mathbf{x}_{\text{true}} = \mathbf{b}_{\text{true}}$ (representing the "true values" of vector $\mathbf{x}, \mathbf{b}$) and $A\mathbf{x}_{\text{approx}} = \mathbf{b}_{\text{approx}}$ (representing the "approximate values" or "observed values by some experiment"), in class we defined the p-norm relative error (for $1 \le p \le \infty$)

$$\text{RelError}_p(\mathbf{x}_{\text{approx}}, \mathbf{x}_{\text{true}}) \stackrel{\text{def}}{=} \frac{\|\mathbf{x}_{\text{approx}} - \mathbf{x}_{\text{true}}\|_p}{\|\mathbf{x}_{\text{true}}\|_p} \tag{1}$$

(assuming $\mathbf{x}_{\text{true}} \neq \mathbf{0}$) and similarly with $\mathbf{x}$ replaced with $\mathbf{b}$. (See also [A&G], pages 3 and Section 5.8.) In class we proved that

$$\text{RelError}_p(\mathbf{x}_{\text{approx}}, \mathbf{x}_{\text{true}}) \le \kappa_p(A)\, \text{RelError}_p(\mathbf{b}_{\text{approx}}, \mathbf{b}_{\text{true}}) \tag{2}$$

where

$$\kappa_p(A) = \|A\|_p \left\|A^{-1}\right\|_p,$$

and, moreover, that for any A there are $\mathbf{x}_{\text{true}}, \mathbf{b}_{\text{true}}, \mathbf{x}_{\text{approx}}, \mathbf{b}_{\text{approx}}$ for which equality holds in (2). Equivalently, if $\mathbf{x}_{\text{error}} = \mathbf{x}_{\text{approx}} - \mathbf{x}_{\text{true}}$ and similarly for $\mathbf{b}_{\text{error}}$, then (2) is equivalent to

$$\frac{\|\mathbf{x}_{\text{error}}\|_p}{\|\mathbf{b}_{\text{error}}\|_p} \frac{\|\mathbf{b}_{\text{true}}\|_p}{\|\mathbf{x}_{\text{true}}\|_p} \le \kappa_p(A),$$

or equivalently

$$\frac{\|A^{-1}\mathbf{b}_{\text{error}}\|_p}{\|\mathbf{b}_{\text{error}}\|_p}\ \frac{\|A\mathbf{x}_{\text{true}}\|_p}{\|\mathbf{x}_{\text{true}}\|_p} \le \kappa_p(A). \tag{3}$$

([A&G] refer to $\mathbf{b}_{\text{error}}$ as the *residual*, and denote it $\hat{\mathbf{r}}$.)

Conversely, for any A, p, here is a recipe for producing cases where (2) holds with equality: let $\mathbf{b}_{\text{error}}$ and $\mathbf{x}_{\text{true}}$ be arbitrary (nonzero) vectors such that

$$\frac{\|A^{-1}\mathbf{b}_{\text{error}}\|_p}{\|\mathbf{b}_{\text{error}}\|_p} = \|A^{-1}\|_p, \quad \frac{\|A\mathbf{x}_{\text{true}}\|_p}{\|\mathbf{x}_{\text{true}}\|_p} = \|A\|_p \tag{4}$$

(such vectors do exist); then (3) holds, and so working backwards we set

$$\mathbf{x}_{\text{error}} = A^{-1}\mathbf{b}_{\text{error}}, \quad \mathbf{b}_{\text{true}} = A\mathbf{x}_{\text{true}}, \tag{5}$$

and

$$\mathbf{x}_{\text{approx}} = \mathbf{x}_{\text{true}} + \mathbf{x}_{\text{error}}, \quad \mathbf{b}_{\text{approx}} = \mathbf{b}_{\text{true}} + \mathbf{b}_{\text{error}}, \tag{6}$$

yielding an example for which (2) holds with equality.

(4) Let $\epsilon > 0$ be a real number (which we think of as small), and let

$$A = \begin{bmatrix} 1 & 2 \\ 1 & 2+\epsilon \end{bmatrix}, \tag{7}$$

and hence

$$A^{-1} = \frac{1}{\epsilon}\begin{bmatrix} 2+\epsilon & -2 \\ -1 & 1 \end{bmatrix},$$

(a) What are $\|A\|_\infty$ and $\|A^{-1}\|_\infty$?

(b) Show that

$$\left\| A\begin{bmatrix}1\\1\end{bmatrix}\right\|_\infty = \|A\|_\infty \left\|\begin{bmatrix}1\\1\end{bmatrix}\right\|_\infty,$$

and for any $\delta \in \mathbb{R}$

$$\left\| A^{-1}\begin{bmatrix}\delta\\-\delta\end{bmatrix}\right\|_\infty = \|A^{-1}\|_\infty \left\|\begin{bmatrix}\delta\\-\delta\end{bmatrix}\right\|_\infty.$$

(c) Use the previous part to show that

$$\mathbf{b}_{\text{error}} = \begin{bmatrix}\delta\\-\delta\end{bmatrix}, \quad \mathbf{x}_{\text{true}} = \begin{bmatrix}1\\1\end{bmatrix}$$

satisfy (4); then let $\mathbf{x}_{\text{error}}$ satisfying (5), and show that the resulting $\mathbf{x}_{\text{approx}}$ is

$$\mathbf{x}_{\text{approx}}(\delta) = \begin{bmatrix}1\\1\end{bmatrix} + \begin{bmatrix}4+\epsilon\\-2\end{bmatrix}\frac{\delta}{\epsilon} \tag{8}$$

(d) Show that $\mathbf{x}_{\text{approx}}(0)$ equals x_{true} above.

(e) Now check your work: let $\mathbf{x}_{\text{approx}}(\delta)$ be as in (8), and let $\delta \neq 0$.

(i) Evaluate

$$\text{RelError}_\infty(\mathbf{x}_{\text{approx}}, \mathbf{x}_{\text{true}}) = \frac{\|\mathbf{x}_{\text{approx}}(\delta) - \mathbf{x}_{\text{approx}}(0)\|_\infty}{\|\mathbf{x}_{\text{approx}}(0)\|_\infty}.$$

(ii) Evaluate

$$\mathrm{RelError}_\infty(A\mathbf{x}_{\mathrm{approx}}, A\mathbf{x}_{\mathrm{true}}) = \frac{\|A\mathbf{x}_{\mathrm{approx}}(\delta) - A\mathbf{x}_{\mathrm{approx}}(0)\|_\infty}{\|A\mathbf{x}_{\mathrm{approx}}(0)\|_\infty}.$$

(iii) Divide the result in (i) by (ii) and show that the result is equal to

$$\kappa_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty$$

(which you should find to be $(3+\epsilon)(4+\epsilon)/\epsilon$, using part (a)).

(5) Let $\epsilon > 0$ be fixed, and let A be given by (7). Consider the function of a real η given by

$$\mathbf{x}(\eta) \stackrel{\mathrm{def}}{=} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 2 \\ -1 \end{bmatrix} \eta.$$

(a) Show that for any $\eta \neq 0$ we have

$$\frac{\mathrm{RelError}_\infty\big(\mathbf{x}(\eta), \mathbf{x}(0)\big)}{\mathrm{RelError}_\infty\big(A\mathbf{x}(\eta), A\mathbf{x}(0)\big)} = \frac{6+2\epsilon}{\epsilon}. \tag{9}$$

(b) Let $\mathbf{x}_{\mathrm{approx}}(\delta)$ be as in (8) with $\delta \neq 0$. From Problem (4) we know that

$$\frac{\mathrm{RelError}_\infty\big(\mathbf{x}_{\mathrm{approx}}(\delta), \mathbf{x}_{\mathrm{approx}}(0)\big)}{\mathrm{RelError}_\infty\big(A\mathbf{x}_{\mathrm{approx}}(\delta), A\mathbf{x}_{\mathrm{approx}}(0)\big)} = \kappa_\infty(A) = \frac{12+7\epsilon+\epsilon^2}{\epsilon}, \tag{10}$$

which is roughly twice as large as the quantity in (9) for small $\epsilon > 0$. Nonetheless, show that if $\eta = 2\delta/\epsilon$, and $|\eta| \le 1/4$

$$\mathrm{RelError}_\infty\big(\mathbf{x}_{\mathrm{approx}}(\delta), \mathbf{x}(2\delta/\epsilon)\big) \le |\delta|.$$

[Hint: show that $\|\mathbf{x}(\eta)\|_\infty \ge 1$ for all $\eta \in \mathbb{R}$, by considering $\eta \ge 0$ and $\eta < 0$ separately.]

[Hence $\mathbf{x}_{\mathrm{approx}}(\delta), \mathbf{x}(2\delta/\epsilon)$ can be arbitrarily relatively close, and $\mathbf{x}_{\mathrm{approx}}(0) = \mathbf{x}(0) = \mathbf{x}_{\mathrm{true}}$, but their loss of precision in solving $A\mathbf{x} = \mathbf{b}$ to the same "true solution" can differ by roughly a factor of two (i.e., the right-hand-sides of (9) and (10)).]

(6) Fix $\epsilon > 0$. In interpolation with a line through the data points (x_0, y_0) and (x_1, y_1), we get a system of equation $A\mathbf{c} = \mathbf{y}$ where A is as in (7) whern $x_0 = 2$ and $x_0 = 2+\epsilon$. But this system, namely

$$\begin{bmatrix} 1 & 2 \\ 1 & 2+\epsilon \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \end{bmatrix}, \tag{11}$$

is equivalent to

$$\begin{bmatrix} 1 & 2 \\ 0 & \epsilon \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 - y_0 \end{bmatrix},$$

which is equivalent to

$$\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} y_0 \\ (y_1 - y_0)/\epsilon \end{bmatrix}. \tag{12}$$

(See the page 2 of "CPSC 303: What the Conidition Number Does and Does Not Tell Us," and note a similar matrix on Homework 5.) But we easily compute the condition numbers

$$\kappa_\infty(A) = \kappa_\infty\left(\begin{bmatrix}1 & 2\\ 1 & 2+\epsilon\end{bmatrix}\right) = \frac{12+7\epsilon+\epsilon^2}{\epsilon}, \quad \kappa_\infty\left(\begin{bmatrix}1 & 2\\ 0 & 1\end{bmatrix}\right) = 9. \tag{13}$$

Doesn't it seem strange that (11) involves a condition number tending to infinity as $\epsilon \to 0$, and the equivalent (12) has a fixed condition number??? (This is a rhetorical question, not part of Problem 6.) The point of this exercise is to explain make this phenomenon seem less strange. [Recall that the fact that $\kappa_\infty(A) = (12 + O(\epsilon))/\epsilon$ was an important indication of a type of degeneracy in interpolation when $x_0 = 2$ and $x_1 = 2 + \epsilon$ and $\epsilon \to 0$.]

If $\mathbf{x}(\eta)$ is as in Problem 5, and for some $\eta \neq 0$ we set

$$\mathbf{c} = \mathbf{x}(0) = \begin{bmatrix}1\\1\end{bmatrix} \quad \mathbf{c}_{\text{approx}} = \mathbf{x}(\eta), \tag{14}$$

then Problem 5 shows that we have

$$\text{RelError}_\infty\big(\mathbf{c}_{\text{approx}}, \mathbf{c}\big) = \frac{6+2\epsilon}{\epsilon}\text{RelError}_\infty\big(A\mathbf{c}_{\text{approx}}, A\mathbf{c}\big). \tag{15}$$

On the other hand, (12) and (13) implies that

$$\text{RelError}_\infty\big(\mathbf{c}_{\text{approx}}, \mathbf{c}\big) \leq 9\ \text{RelError}_\infty\big(B\mathbf{c}_{\text{approx}}, B\mathbf{c}\big) \tag{16}$$

where $B\mathbf{c}$ takes $A\mathbf{c}$ and applies the function $(y_0, y_1) \mapsto (y_0, (y_1 - y_0)/\epsilon)$. Hence the last two equations imply that

$$9\ \text{RelError}_\infty\big(B\mathbf{c}_{\text{approx}}, B\mathbf{c}\big) \geq \frac{6+2\epsilon}{\epsilon}\text{RelError}_\infty\big(A\mathbf{c}_{\text{approx}}, A\mathbf{c}\big). \tag{17}$$

In other words passing from A to B in the above introduces a factor of order $1/\epsilon$.

(Here is what you are asked to do for this problem:)

(a) Verify (17) directly.

(b) If $x(\eta)$ is replaced with $x_{\text{approx}}(\delta)$, write down the inequality that results in place of (17) (but you don't have to verify it directly).

Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, CANADA.

E-mail address: jf@cs.ubc.ca

URL: http://www.cs.ubc.ca/~jf