# FINAL PRACTICE, CPSC 303, SPRING 2024

JOEL FRIEDMAN



The following problems are sample final exam problems, largely based on the homework. There are a lot on material after the midterm, i.e., starting with divided differences, but some cover earlier material.

(1) Answer true or false:
  (a) In the central force problem, $m\ddot{\mathbf{x}} = -u(\|\mathbf{x}\|)\mathbf{x}/\|\mathbf{x}\|$, the momentum $m\mathbf{v} = m\dot{\mathbf{x}}$ is conserved.
  (b) In the central force problem, $m\ddot{\mathbf{x}} = -u(\|\mathbf{x}\|)\mathbf{x}/\|\mathbf{x}\|$, the energy $(1/2)m\|\mathbf{v}\|^2 + U(\|\mathbf{x}\|)$ is conserved, where $\mathbf{v} = \dot{\mathbf{x}}$ and $U$ satisfies $U' = u$.
  (c) The central force problem, $m\ddot{\mathbf{x}} = -u(\|\mathbf{x}\|)\mathbf{x}/\|\mathbf{x}\|$ is reversable in time, i.e., if $\mathbf{x}(t)$ is a solution, then $\mathbf{x}(T-t)$ is also a solution.
  (d) If $A$ is a $2\times 2$ matrix, then there exists an $\mathbf{x}$ each of whose components are all either $1, -1$, such that $\|A\mathbf{x}\|_\infty = \|A\|_\infty \|x\|_\infty$.
  (e) If $A$ is a $2\times 2$ matrix, and $\mathbf{x} \neq \mathbf{0}$ satisfies $\|A\mathbf{x}\|_\infty = \|A\|_\infty \|x\|_\infty$, then all components of $\mathbf{x}$ are either $1, -1$.
  (f) We have an algorithm for monomial interpolation on $n$ points that takes order $n^2$ floating point operations to form the polynomial and evaluate it at one point.
  (g) We have an algorithm for Lagrange interpolation on $n$ points that takes order $n^2$ floating point operations to form the polynomial and evaluate it at one point.
  (h) We have an algorithm for Newton's form of interpolation (i.e., via divided differences) on $n$ points that takes order $n^2$ floating point operations to form the polynomial and evaluate it at one point.
  (i) There exist functions $f\colon \mathbb{R} \to \mathbb{R}$ such that $f[0,1,2] \neq f[0,2,1]$.
  (j) MORE EXERCISES MAY APPEAR

Note: for a cubic spline $v(x)$ with abscissae $A = x_0 < \cdots < x_n = B$ that approximates $f(x)$, the following formulas are useful regarding the piecewise cubic spline $s_i(x) = a_i + b_i(x-x_i) + c_i(x-x_i)^2 + d_i(x-x_i)^3$ satisfies (this piece is for $v$ on $[x_i, x_{i+1}]$:

$$a_i = f(x_i), \quad b_i = f[x_i, x_{i+1}] - \frac{h_i}{3}(2c_i + c_{i+1}), \quad d_i = \frac{c_{i+1} - c_i}{3h_i}$$

and that for each $i = 1, \ldots, n-1$ we have

$$\frac{h_i}{h_i + h_{i+1}} c_{i-1} + 2c_i + \frac{h_{i+1}}{h_i + h_{i+1}} c_{i+1} = 3f([x_{i-1}, x_i, x_{i+1}]).$$

where $h_i = x_{i+1} - x_i$, and where $c_0 = v''(A)$ and $c_n = v''(B)$ ($c_n$ is not really defined since $s_n(x)$ isn't defined, but it is convenient to define it so that the above formulas hold).

(2) Say that $x_0 < x_1 < \cdots < x_n$ are reals, and $v(x)$ is a cubic spline with abscissae $x_0, \ldots, x_n$ (i.e., $v(x)$ is a piecewse cubic polynomial in the intervals $(x_i, x_{i+1})$ for $i = 0, \ldots, n-1$). How many parameters describe $v(x)$? If we insist that $v(x_0), \ldots, v(x_n)$ take on $n+1$ given values, how many equations to we get on the parameters describing $v$? If we insist that $v$ is twice continuously differentiable at $x_1, \ldots, x_{n-1}$, show many equations do we get?

(3) Say that $A = x_0 < x_1 < \cdots < x_n = B$, and $v(x)$ is a cubic spline approximating $f(x)$, with abscissae $x_0, \ldots, x_n$. Say that instead of the natural spline (i.e., $v''(A) = v''(B) = 0$), we fix $v'(A) = f'(A)$ (which we presume is known), but we keep the condition $v''(B) = 0$. As usual, say that $h_1 = h_2 = \cdots = h_{n-1}$.
   (a) What does $v'(A) = f'(A)$ tell you about the coefficients of the cubic polynomial $s_0(x) = a_0 + b_0(x - x_0) + c_0(x - x_0)^2 + d_0(x - x_0)^3$ that agrees with $v(x)$ in $[x_0, x_1]$?
   (b) Show that the above information allows you to determine $2c_0 + c_1$.
   (c) If we now write a system of equations for $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$, do we still get a system of equations where $(2I + M)\mathbf{c}$ is determined for some matrix with $\|M\|_\infty \le 1$?

For divided differences the following formulas are useful: $f[x_0, \ldots, x_n] = f^{(n)}(\xi)/n!$ for some $\xi$ contained in any interval containing $x_0, \ldots, x_n$ (we called this the "generalized mean-value" theorem). If $p(x)$ interpolates $f(x)$ at $x_0, \ldots, x_n$, then the "error in polynomial interpolation" theorem states that $f(x) - p(x) = f^{(n+1)}(\xi)/(n+1)!$ for some $\xi$ contained in any interval containing $x_0, \ldots, x_n$ and $x$. You should realize that this second statement follows from the first. You should realize that fixing $x_0$ and taking $x_1, \ldots, x_n$ tending to $x_0$ you essentially get Taylor's theorem.

(4) Let $x_0, x_1, x_2, x_3$ be given by $x_i = i/3$, and $f(x) = \sin(x)$. Give a bound on $|f[x_0, x_1, x_2, x_3]|$ using the generalized mean-value theorem.

(5) Let $x_0, x_1, x_2, x_3$ be given by $x_i = i/3$, and $f(x) = \sin(x)$. Say that you find the polynomial $p(x)$ of degree at most 3 such that $p(x_i) = f(x_i)$ $i = 0, \ldots, 3$. Estimate the error in $p(1/2)$ and $f(1/2)$ using the error in Newton divided differences formula.

(6) Let $n \in \mathbb{N}$, and let $x_0, x_1, x_2, x_3$ be given by $x_i = 10^{-n}i/3$. Say that you find the polynomial $p(x)$ of degree at most 3 such that $p(x_i) = \sin(i)$, and that you compute $p(10^{-n}(3/2))$.
  (a) For $n = 6$, is it more numerically accurate to compute $p(x)$ using monomial interpolation or using Lagrange interpolation? Explain roughly why this holds.
  (b) Does the value (with exact computation) of $p(10^{-n}(3/2))$ depend on $n$? Explain.

(7) Say that $n \in \mathbb{N}$, and that you have computed the monomial form of $p(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$ such that $p(i) = 1/i$ for $i = 1, 2, \ldots, n$. Say that $q(x)$ is the unique polynomial of degree at most $n$ such that $q(i) = 1/i$ for $i = 1, 2, \ldots, n+1$. Imagine that you want to compute $q$ at $\ell$ values $y_1, \ldots, y_\ell$.
  (a) Explain how to do this in a way that takes you $O(\ell n + n^2)$ flops (floating-point operations).[1]
  (b) If you used Lagrange interpolation (from scratch, ignoring the fact that you already know $p$), explain why this would take $O(\ell n + n^2)$ flops as well (with slightly different constants).

(8) Consider the implicit trapezoidal method (see [A&G], bottom page 493) to solve $\mathbf{y} = \mathbf{f}(\mathbf{y})$, subject to $\mathbf{y}(0) = \mathbf{y}_0$: choose a step size, $h > 0$, and define $\mathbf{y}_1, \mathbf{y}_2, \ldots$ via

(1) $$\mathbf{y}_{i+1} = \mathbf{y}_i + h\frac{\mathbf{f}(\mathbf{y}_i) + \mathbf{f}(\mathbf{y}_{i+1})}{2};$$

taking $\mathbf{y}_i$ as an approximation to $\mathbf{y}(ih)$.
  (a) Consider the single variable ODE $y' = ay$, where $a \in \mathbb{R}$ is constant. Show that the above method gives the recurrence
$$y_{i+1} = y_i \frac{1 + ah/2}{1 - ah/2}.$$
  (b) Show that if $a < 0$ and $y_0 = 1$, we have $0 < |y_{i+1}| < |y_i|$ for all $m$. [Hint: show that $(1 + ah)/(1 - ah)$ is both $< 1$ and $> -1$; since $1 - ah$ is positive, you can multiply both sides by $1 - ah$.]
  (c) Show that if $0 \le ah < 2$ and $y_0 = 1$, we have $y_{i+1} > y_i$ for all $i$. [Hint: use the idea for part (b); if $0 \le ah < 2$, is it still the case that $1 - ah$ is positive?]
  (d) What happens when $ah > 2$? Why is the behaviour of $y_1, y_2, \ldots$ qualitatively different than the true solution $y(t) = e^{at}$?

(9) Let $y = y(t)$ be infinitely differentiable (for simplicity), and consider the approximation
$$\frac{y(t+h) - y(t)}{h} \approx \frac{y'(t+h) + y'(t)}{2}$$
for a fixed $t$ and small $h > 0$.

[1] Note: there are ways to do this in $O(\ell n + n) = O(\ell n)$ flops; see the solutions.

(a) By a Taylor expansion, show that both sides equal $y'(t)+(h/2)y''(t)+O(h^2)$. [Hint: You'll want to apply Taylor's theorem to $y'$, namely $y'(t+h)=y'(t)+hy''(t)+O(h^2)$.]

(b) Show that if $y'(t)=f(y(t))$ for some function, $f$, then

$$y(t+h)=y(t)+h\,\frac{f(y(t+h)+f(y(t))}{2}+O(h^3).$$

(c) Conclude that the scheme (1) is a second order scheme, i.e., each iteration holds to within $O(h^3)$.

(10) NO MORE EXERCISES FOR THIS YEAR.

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

*Email address*: jf@cs.ubc.ca

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