CME 108/MATH 114
Introduction to Scientific Computing
Summer 2018

Problem Set 5

Due: Monday, August 6 at 11:59pm

Warm-up Questions

1. *(Conjugate powers.)* Consider an $n \times n$ matrix A such that $A = PDP^{-1}$, for some matrices P, D. Show that
$$A^m = PD^m P^{-1}.$$

2. *(Inverse determinant.)* Using the multiplicativity of the determinant ($\det(AB) = \det(A)\det(B)$ for any $n \times n$ A, B), show that
$$\det(A^{-1}) = (\det(A))^{-1}$$
for any invertible A.

3. *(Polynomial eigenvalues.)* Suppose λ is an eigenvalue of an $n \times n$ matrix A and that v is a λ-eigenvector of A.
   (a) Show that $A^2 v = \lambda^2 v$.
   (b) Show that λ^m is an eigenvalue of A^m.
   (c) Consider the polynomial $p(x) = c_0 + c_1 x + \cdots + c_m x^m$, and define
$$p(A) = c_0 I_n + c_1 A + \cdots + c_m A^m.$$
   Show that $p(\lambda)$ is an eigenvalue of $p(A)$.

4. *(Inverse eigenvalues.)* Suppose that A is an invertible matrix and that v is a λ-eigenvector of A.

   First show that $\lambda \neq 0$. That is, show that an invertible matrix cannot have the eigenvalue 0.

   Now show that v is a $\frac{1}{\lambda}$-eigenvector of A^{-1}.

5. *(Uniqueness of interpolating polynomial, Bradie 5.1.7.)* Consider the following data set:

x	-1	0	1	2
y	5	1	1	11

   (a) Show that the polynomials $f(x) = x^3 + 2x^2 - 3x + 1$ and $g(x) = \frac{1}{8}x^4 + \frac{3}{4}x^3 + \frac{15}{8}x^2 - \frac{11}{4}x + 1$ both interpolate all of the data.

   (b) Why does this not contradict the uniqueness statement of the theorem on existence and uniqueness of the interpolating polynomial?

6. *(Piecewise linear interpolation, Bradie 5.5.1.)* Consider the following data for the density ρ of water as a function of temperature T:

T (°C)	ρ (kg/m^3)
0	1000
10	1000
20	998
30	996
40	992
50	988
60	983
70	978
80	972
90	965
100	958

   Estimate, using piecewise linear interpolation, the density of water when the temperature is 34 °C, 68 °C, and 91 °C.

Assignment Problems

1. (20 points) (*Computing eigenvalues.*) Consider the matrix

$$A = \begin{bmatrix} 0 & -17 & 21 \\ 0 & 13 & -15 \\ 0 & 10 & -12 \end{bmatrix}.$$

   Recall that $\lambda \in \mathbb{R}$ is an *eigenvalue* of A if the equality $Av = \lambda v$ is satisfied by some non-zero vector v.

(a) It turns out that the eigenvalues of any $n \times n$ matrix M are exactly the roots of the polynomial

$$\chi(\lambda) = \det(\lambda I - M).$$

This polynomial is known as the *characteristic polynomial* of M.

Compute the eigenvalues of the given matrix A.

(b) In MATLAB, implement the following algorithm:

INPUT : $n \times n$ matrix A, n-vector q_0, tolerance ϵ, maximum number of iterations N

$q \leftarrow \frac{q_0}{\|q_0\|}$
$\lambda_{\text{curr}} \leftarrow 0$
for $k = 1, \ldots, N$ **do**
  $\lambda_{\text{prev}} \leftarrow \lambda_{\text{curr}}$
  $v \leftarrow Aq$
  $\lambda_{\text{curr}} \leftarrow q^T v$
  $q \leftarrow \frac{v}{\|v\|}$
  if $|\lambda_{prev} - \lambda_{curr}| < \epsilon$ **then**
    break
  end
end
OUTPUT: λ_{curr}

Use the MATLAB command `rand` to generate a vector q_0 with random entries and run your algorithm using as input: the given A, the generated q_0, $\epsilon = 10^{-6}$, and $N = 100$.

With your results from part (a) in mind, what can you say about the output of your algorithm?

(c) Lastly, implement the following algorithm in MATLAB:

INPUT : matrix A, maximum number of iterations N
for $k = 1, \ldots, N$ **do**
  compute LU factorization $A = LU$
  $A \leftarrow UL$
end
OUTPUT: A

For the LU factorization, you may assume no pivoting is needed. You may use code that you have written previously or the MATLAB function `lu`.

Run your algorithm using the given matrix A as input, and set $N = 100$. With your results from (a) in mind, what can you say about the output of your algorithm?

Hint: Pay particular attention to the output diagonal!

2. (10 points) *(Eigen-theory.)* **BONUS PROBLEM!** Let A be an $n \times n$ matrix with real entries.

   (a) Assuming the statement of the previous prompt, show that A can have at most n real eigenvalues. In addition, show that if μ is a complex eigenvalue of A, then $\bar{\mu}$ is also an eigenvalue of A (the over-bar denotes complex conjugation).

   (b) Suppose that P is any invertible $n \times n$ matrix. Show that A and $P^{-1}AP$ have the same eigenvalues.

   (c) If D is a diagonal matrix, what are the eigenvalues of D?

   (d) Use the characteristic polynomial to show that

$$\det(A) = \prod_{i=1}^{n} \lambda_i,$$

   where λ_i denote the eigenvalues of A.

3. (15 points) *(Interpolation points, Bradie 5.1.10.)* The interpolation points influence the interpolation error through the polynomial $\prod_{i=0}^{n}(x - x_i)$. Suppose we are interpolating the function f over the interval $[-1, 1]$ using linear interpolation.

   (a) If $x_0 = -1$ and $x_1 = 1$, determine the maximum value of the expression $|(x - x_0)(x - x_1)|$ for $-1 \leq x \leq 1$.

   (b) If $x_0 = -\sqrt{2}/2$ and $x_1 = \sqrt{2}/2$, determine the maximum value of the expression $|(x - x_0)(x - x_1)|$ for $-1 \leq x \leq 1$. How does this maximum compare to the maximum found in part (a)?

   (c) Select any two numbers from the interval $[-1, 1]$ to serve as the interpolation points x_0 and x_1. Determine the maximum value of the expression $|(x - x_0)(x - x_1)|$ for $-1 \leq x \leq 1$, and compare to the maxima found in (a) and (b).

4. (25 points) *(Cubic spline, Bradie 5.6.10.)* In MATLAB, implement the construction of the cubic spline interpolant, as described in lecture. In particular, follow these steps:

- Set up a tridiagonal linear system for c_j and solve.
- Compute the other coefficients b_j and d_j.
- Combine your coefficients to form $s_j(x)$.

Now consider the following data set:

x	0.0	0.5	1.0	1.5	2.0
y	0.500000	1.425639	2.640859	4.009155	5.305472
y'	1.500000				2.305472

(a) Construct the cubic spline interpolant for this data set using the not-a-knot boundary conditions.

(b) Construct the cubic spline interpolant for this data set using the clamped spline boundary conditions.

(c) The data for this problem is taken from the function $y = (1+x)^2 - 0.5e^x$. Plot the error $\epsilon(x) = y(x) - s(x)$ in each of the splines from (a) and (b) . Which spline produced better results?

Note: You may (should) use the built-in `spline` function to verify your computations.

5. (20 points) *(Lake pollution, from "Fundamentals of Engineering Numerical Analysis".)* The concentration of a certain toxin in a system of lakes downwind of an industrial area has been monitored very accurately at intervals from 1978 to 1992 as shown in the table below. It is believed that the concentration has varied smoothly between these data points.

Year	Concentration (ppm)
1978	12.0
1980	12.7
1982	13.0
1984	15.2
1986	18.2
1988	19.8
1990	24.1
1992	28.1

(a) Interpolate the data using the Lagrange polynomial. Plot the polynomial along with the data points. Use your polynomial to predict the condition of the lakes in 1994.

(b) Use a cubic spline interpolant with not-a-knot boundary conditions to predict the toxin concentration in 1994. You may use your code from Problem 4 or the MATLAB function `spline`.

(c) Comment on the difference between your predictions.

6. (25 points) *(Quadratic spline.)* In this problem you will explore a construction of a quadratic spline. The development is akin to that of the cubic spline, except that the interpolating polynomial is piecewise quadratic (parabolic) instead of cubic.

Given $a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$ and $y_j = f(x_j)$, the quadratic spline $s(x)$ is defined by the conditions:

- (Parabolicity) $s(x) = s_j(x)$ for $x \in [x_j, x_{j+1}]$,
- (Interpolation) $s(x_j) = y_j$ for $j = 0, \ldots, n$,
- (Continuity) $s_j(x_{j+1}) = s_{j+1}(x_{j+1})$ for $j = 0, \ldots, n-2$,
- (Smoothness) $s_j'(x_{j+1}) = s_{j+1}'(x_{j+1})$ for $j = 0, \ldots, n-2$,

with

$$s_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2,$$

for $j = 0, \ldots, n-1$.

(a) Manipulate the defining relations to obtain a *recursive relation* for b_j, $j = 1, \ldots, n-1$.

Note: In lecture we showed how to manipulate the relations defining the cubic spline to obtain a tridiagonal system for the c_j's. The manipulation here is similar in nature, but much simpler.

(b) Note that the defining relations provide only $3n - 1$ conditions on $3n$ variables. A similar situation arises in the case of the cubic spline, and we need to impose additional conditions (e.g., not-a-knot, clamped spline) in order to uniquely specify the interpolant.

How many additional conditions are needed to uniquely specify the quadratic spline? Propose a set of additional conditions.

Note: There are many valid sets of additional conditions.