Homework 2

January 27, 2022

Problem 1
Let \( f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n \) be an invertible map from inputs \( x \in \mathbb{R}^n \) (\( n \)-component column vectors) to outputs in \( \mathbb{R}^n \). If \( g(x) \) is the inverse function, so that \( g(f(x)) = x = f(g(x)) \), use the chain rule to show how the Jacobians \( f'(x) \) and \( g'(x) \) are related.

Problem 2
Consider \( f(A) = \sqrt{A} \) where \( A \) is an \( n \times n \) matrix. (Recall that the matrix square root is a matrix such that \( (\sqrt{A})^2 = A \). In 18.06, we might compute this from a diagonalization of \( A \) by taking the square roots of all the eigenvalues.)

a) Give a formula for the Jacobian of \( f \), acting on \( \text{vec}(A) \), in terms of Kronecker products and matrix powers only (no eigenvalues required!). (Hint: see problem 1.)

b) Check your formula numerically against a finite-difference approximation. (To ensure that no complex numbers show up in the matrix square root, pick a random positive-definite \( A = B^T B \) for some random square \( B \).)

Problem 3
Suppose that \( A(p) = A_0 + \text{diag}(p) \) constructs an \( n \times n \) matrix \( A \) from \( p \in \mathbb{R}^n \), where \( \text{diag}(p) = \begin{pmatrix} p_1 & & \\ & p_2 & \\ & & \ddots \end{pmatrix} \) is the diagonal matrix with the entries of \( p \) along its diagonal, and \( A_0 \) is some constant matrix. We now perform the following sequence of steps:

1. solve \( A(p)x = a \) for \( x \in \mathbb{R}^n \), where \( a \) is some vector.
2. form \( B(x) = B_0 + \text{diag}(x \times x) \) where \( \times \) is the elementwise product and \( B_0 \) is some \( n \times n \) matrix.
3. solve \( By = b \) for \( y \in \mathbb{R}^n \), where \( b \) is some vector.
4. compute \( f = y^T F y \) where \( F = F^T \) is some symmetric \( n \times n \) matrix.

Now, suppose that we want to optimize \( f(p) \) as a function of the parameters \( p \) that went into the first step. We need to compute the gradient \( \nabla f \) for any kind of large-scale problem. If \( n \) is huge, though, we must be careful to do this efficiently, using “reverse-mode” or “adjoint” calculations in which we apply the chain rule from left to right.

a) Explain the sequence of steps to compute \( \nabla f \). Your answer should show that \( \nabla f \) can also be computed by solving only two \( n \times n \) linear systems, similar to \( f \) itself.

b) Check your answer numerically against finite differences (for some randomly chosen \( A_0, B_0, a, b, F \)).

c) Check your answer against the result of a reverse-mode AD software (e.g. Zygote in Julia).

Problem 4
a) Write down some \( 4 \times 4 \) matrix that is not the Kronecker product of two \( 2 \times 2 \) matrices. Convince us this is true.

b) Prove that if \( A \) (\( m \times m \)) and \( B \) (\( n \times n \)) are orthogonal (i.e., \( A^T A = I_m \) and \( B^T B = I_n \)) then \( A \otimes B \) (\( mn \times mn \)) is orthogonal.

c) If \( f(A) = e^A = \sum_{k=0}^{\infty} A^k / k! \), write down a power series involving Kronecker products for the Jacobian \( f'(A) \). Check your answer with a numerical example, e.g. against finite differences.