

# 18.S096 PSET 1 Solutions

IAP 2023

January 26, 2023

## Problem 0 (4+4+4+4 points)

The hyperbolic Corgi notebook may be found at [https://mit-c25.netlify.app/notebooks/1\\_hyperbolic\\_corgi](https://mit-c25.netlify.app/notebooks/1_hyperbolic_corgi). Compute the  $2 \times 2$  Jacobian matrix for each of the following image transformations from that notebook:

- (a) `rotate( $\theta$ )`:  $(x, y) \rightarrow (\cos(\theta)x + \sin(\theta)y, -\sin(\theta)x + \cos(\theta)y)$

**Solution:** This is simply a linear function from  $\mathbb{R}^2 \rightarrow \mathbb{R}^2$

$$\underbrace{\begin{pmatrix} x \\ y \end{pmatrix}}_{\vec{x}} \rightarrow \underbrace{\begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix}}_{R(\theta)} \begin{pmatrix} x \\ y \end{pmatrix}$$

By the same reasoning as in problem 1, the derivative (Jacobian) is simply the rotation operator  $R(\theta)$ :  $d(R\vec{x}) = R\vec{dx}$ , and hence the Jacobian is  $\boxed{R(\theta)}$ .

- (b) `hyperbolic_rotate( $\theta$ )`:  $(x, y) \rightarrow (\cosh(\theta)x + \sinh(\theta)y, \sinh(\theta)x + \cosh(\theta)y)$

**Solution:** This is another linear transformation:

$$\underbrace{\begin{pmatrix} x \\ y \end{pmatrix}}_{\vec{x}} \rightarrow \underbrace{\begin{pmatrix} \cosh \theta & \sinh \theta \\ \sinh \theta & \cosh \theta \end{pmatrix}}_{H(\theta)} \begin{pmatrix} x \\ y \end{pmatrix}$$

with Jacobian  $\boxed{H(\theta)}$ .

- (c) `nonlin_shear( $\theta$ )`:  $(x, y) \rightarrow (x, y + \theta x^2)$

**Solution:** The differential is:

$$d \begin{pmatrix} x \\ y + \theta x^2 \end{pmatrix} = \begin{pmatrix} dx \\ dy + 2\theta x dx \end{pmatrix} = \boxed{\begin{pmatrix} 1 & 0 \\ 2\theta x & 1 \end{pmatrix}} \begin{pmatrix} dx \\ dy \end{pmatrix}$$

so the Jacobian is the boxed matrix.

- (d) `warp( $\theta$ )`:  $(x, y) \rightarrow \text{rotate}(\theta\sqrt{x^2 + y^2})(x, y)$

**Solution:** This is the function  $\vec{x} \rightarrow R(\theta\|\vec{x}\|)\vec{x}$  in terms of the rotation matrix  $R(\theta)$  from part (a), so we can use the product rule:

$$d(R(\theta\|\vec{x}\|)\vec{x}) = dR\vec{x} + R\vec{dx}$$

where by the chain rule:

$$dR = R'(\theta\|\vec{x}\|)d(\theta\|\vec{x}\|) = \theta R'(\theta\|\vec{x}\|)d(\|\vec{x}\|)$$

with

$$R'(\phi) = \begin{pmatrix} -\sin \phi & \cos \phi \\ -\cos \phi & -\sin \phi \end{pmatrix}$$

by familiar 18.01 derivatives of each component—which follows from the definition  $dR = R(\phi + d\phi) - R(\phi) = R'(\phi)d\phi$ , since the scalar  $d\phi$  multiplies  $R'$  elementwise. To get  $d(\|\vec{x}\|)$  we can apply the chain rule again:

$$d(\|\vec{x}\|) = d((\vec{x}^T \vec{x})^{1/2}) = \frac{d(\vec{x}^T \vec{x})}{2(\vec{x}^T \vec{x})^{1/2}} = \frac{\vec{x}^T d\vec{x}}{\|\vec{x}\|},$$

noting that familiar 18.01 calculus rules work fine when applying the chain rule to scalar terms.<sup>1</sup> Hence, putting it all together and rearranging scalar terms (which we can move freely), we have:

$$\begin{aligned} d(\text{warp } \vec{x}) &= \frac{\theta}{\|\vec{x}\|} R'(\theta\|\vec{x}\|) \vec{x} \vec{x}^T d\vec{x} + R d\vec{x} \\ &= \left( \boxed{\theta\|\vec{x}\| R'(\theta\|\vec{x}\|) \frac{\vec{x} \vec{x}^T}{\vec{x}^T \vec{x}} + R(\theta\|\vec{x}\|)} \right) d\vec{x} \end{aligned}$$

in terms of  $R$  and  $R'$  defined above, with the boxed term being the Jacobian, and we have re-arranged terms to “beautify” the expression by making it clear that  $\frac{\vec{x} \vec{x}^T}{\vec{x}^T \vec{x}} = \frac{\vec{x} \vec{x}^T}{\|\vec{x}\|^2}$  is an orthogonal projection operator.

## Problem 1 (5+4 points)

- (a) Suppose that  $L[x]$  is a linear operation (for  $x$  in some vector space  $V$ , with outputs  $L[x]$  in some other vector space  $W$ ). If  $f(x) = L[x] + y$  for a constant  $y \in W$ , what is  $f'(x)$  (in terms of  $L$  and/or  $y$ )?

**Solution:** This problem is mainly about knowing the definitions of linear operators and derivatives. If  $f(x) = L[x] + y$ , then

$$df = f(x + dx) - f(x) = (\underbrace{L[x + dx]}_{=L[x] + L[dx]} + y) - (L[x] + y) = L[dx]$$

so we have  $\boxed{f'(x)[dx] = L[dx]}$  or equivalently  $\boxed{f'(x) = L}$ . For affine functions, the derivative is just the linear part.

- (b) Give the derivatives of  $f(A) = A^T$  (transpose) and  $g(A) = 1 + \text{tr } A$  (trace) as special cases of the rule you derived in the previous part.

**Solution:** Again, the key is simply to understand linearity. In both of these examples, we have a linear operator that *you cannot easily write as a matrix  $\times$  vector product* (unless you “vectorize” the inputs and/or outputs).

- (i)  $f(A) = A^T$  is a linear operator because *transposition is linear*:  $(A + B)^T = A^T + B^T$  and  $(\alpha A)^T = \alpha A^T$ . So, in the notation of part (a),  $L[x] = A^T$  and  $y = 0$ , so  $\boxed{f'(A)[dA] = (dA)^T}$ . Equivalently,

$$\boxed{d(A^T) = (dA)^T}.$$

<sup>1</sup>We can alternatively let  $r = \|x\| \implies r^2 = x^T x \implies 2rdr = d(x^T x) = 2x^T dx \implies dr = \frac{2x^T dx}{2r}$ . But this is basically re-deriving a rule from first-year calculus. Once we hit a scalar term we needn’t be shy about applying 18.01 rules.

- (ii) Here, the key is that *trace is linear*:  $\text{tr}(A + B) = \text{tr} A + \text{tr} B$  and  $\text{tr}(\alpha A) = \alpha \text{tr} A$  by inspection of the definition of the trace. So, in the notation of part (a),  $g(x) = \underbrace{1}_y + \underbrace{\text{tr} A}_{L[A]}$  is an affine function with

$$\boxed{g'(A)[dA] = \text{tr}(dA)}, \text{ or equivalently } \boxed{d(1 + \text{tr} A) = \text{tr}(dA)}.$$

## Problem 2 (5+6+5+5 points)

Calculate derivatives of each of the following functions in the requested forms—as a linear operator  $f'(x)[dx]$ , a Jacobian matrix, or a gradient  $\nabla f$ —as specified in each part.

- (a)  $f(x) = x^T(A + \text{diagm}(x))^2x$ , where the inputs  $x \in \mathbb{R}^n$  are vectors, the outputs are scalars,  $A = A^T$  is a constant *symmetric*  $n \times n$  matrix  $\in \mathbb{R}^{n \times n}$ , and  $\text{diagm}(x)$  denotes the  $n \times n$  diagonal matrix  $\begin{pmatrix} x_1 & & \\ & x_2 & \\ & & \ddots \end{pmatrix}$ .

Give the **gradient**  $\nabla f$ , such that  $f'(x)dx = (\nabla f)^T dx$ .

**Solution:** Applying the product rule, we have

$$\begin{aligned} df &= dx^T(A + \text{diagm}(x))^2x + x^T(A + \text{diagm}(x))^2dx \\ &\quad + x^T \underbrace{d(\text{diagm } x)}_{=\text{diagm}(dx)}(A + \text{diagm}(x))x + x^T(A + \text{diagm}(x)) \text{diagm}(dx)x \end{aligned}$$

where  $d(A + \text{diagm}(x)) = d(\text{diagm } x)$  since  $A$  is a constant, and because  $\text{diagm}$  is linear (as in problem 1) we have  $d(\text{diagm } x) = \text{diagm}(dx)$ . Now, in order to get this in the form  $\nabla f \cdot dx$ , we need to move all of our  $dx$  factors to the right. The first trick is one we showed in class for a very similar problem: every scalar equals the transpose of itself, giving

$$dx^T(A + \text{diagm}(x))^2x = [dx^T(A + \text{diagm}(x))^2x]^T = x^T(A + \text{diagm}(x))^2dx$$

using the fact that  $A + \text{diagm}(x)$  is symmetric ( $A = A^T$  was given and  $\text{diagm } x$  is diagonal). Similarly combining the other pair of terms in  $df$ , we get:

$$df = 2x^T(A + \text{diagm}(x))^2dx + 2x^T(A + \text{diagm}(x)) \text{diagm}(dx)x.$$

The second trick is more subtle: if you think carefully about  $\text{diagm}(dx)x$ , you will realize that it is simply an *elementwise product* (denoted by  $.*$  in Julia), so:

$$\text{diagm}(dx)x = dx .* x = x .* dx = \text{diagm}(x)dx$$

Hence

$$df = [2x^T(A + \text{diagm}(x))^2 + 2x^T(A + \text{diagm}(x)) \text{diagm}(x)] dx$$

and  $\nabla f = [\dots]^T$  therefore gives

$$\boxed{\nabla f = 2[(A + \text{diagm}(x))^2 + \text{diagm}(x)(A + \text{diagm}(x))]x = 2(A + 2\text{diagm}(x))(A + \text{diagm}(x))x}.$$

- (b)  $f(x) = (A + yx^T)^{-1}b$ , where the inputs  $x$  and outputs  $f(x)$  are  $n$ -component (column) vectors in  $\mathbb{R}^n$ ,  $y$  and  $b$  are constant vectors  $\in \mathbb{R}^n$ , and  $A$  is a constant  $n \times n$  matrix  $\in \mathbb{R}^{n \times n}$ .

- (i) Give  $f'(x)$  as a **Jacobian** matrix.

**Solution:** The key here is the formula derived in class for the derivative of a matrix inverse:  $d(B^{-1}) = -B^{-1} dB B^{-1}$ . Applying this to  $B = A + yx^T$  and  $dB = y(dx)^T$ , and hence to  $f(x)$  via the product rule, gives:

$$\begin{aligned} df &= -(A + yx^T)^{-1} y(dx)^T \underbrace{(A + yx^T)^{-1} b}_{f(x)} \\ &= -(A + yx^T)^{-1} y f(x)^T dx, \end{aligned}$$

where we have again used  $(dx)^T f(x) = f(x)^T dx$  to move  $dx$  to the right. By inspection, our Jacobian matrix is then the rank-1 matrix:

$$\boxed{f'(x) = -(A + yx^T)^{-1} y f(x)^T}.$$

- (ii) If you are given  $A^{-1}$ , then you can compute  $(A + yx^T)^{-1}$  and hence  $f(x)$  for any  $x$  in  $\sim n^2$  scalar-arithmetic operations (i.e., roughly proportional to  $n^2$ , or in computer-science terms  $\Theta(n^2)$  “complexity”), using the “Sherman–Morrison” formula (Google it). **Explain** how your Jacobian matrix can therefore also be computed in  $\sim n^2$  operations for any  $x$  given  $A^{-1}$  (i.e. give a sequence of computational steps, each of which costs no more than  $\sim n^2$  arithmetic).

**Solution:** Since we have  $(A + yx^T)^{-1}$  in  $\sim n^2$  operations for any  $x$ , we can also use it to compute  $c = (A + yx^T)^{-1} y$  by an additional matrix–vector multiplication ( $\sim n^2$  scalar arithmetic operations). Our Jacobian is then the outer product (column  $\times$  row)

$$f'(x) = -c f(x)^T$$

which requires an additional  $n^2$  multiplications (and  $n$  negations of  $c$ ) to yield an  $n \times n$  matrix. Hence, overall, the whole process requires an operation count that scales proportional to  $n^2$ .

Note that the order in which we do the operations matters! If we computed it in the order

$$f'(x) = -(A + yx^T)^{-1} (y f(x)^T)$$

we would have had a matrix–matrix multiplication costing  $\sim n^3$  operations, even if the matrix inversion had a cost  $\sim n^2$ .

- (c)  $f(x) = \frac{xx^T}{x^T x}$ , with vector inputs  $x \in \mathbb{R}^n$  and matrix outputs  $f \in \mathbb{R}^{n \times n}$ . Give  $f'(x)$  as a linear operator, i.e. a linear formula for  $f'(x)[dx]$ .

**Solution:** We mainly just apply the product rule here, noting that  $d((x^T x)^{-1})$  simplifies to the ordinary

quotient rule because  $x^T x$  is a scalar:

$$\begin{aligned} df &= \frac{d(xx^T)}{x^T x} + xx^T d((x^T x)^{-1}) \\ &= \frac{dx x^T + x dx^T}{x^T x} - \frac{xx^T d(x^T x)}{(x^T x)^2} \\ &= \boxed{\frac{dx x^T + x dx^T}{x^T x} - \frac{xx^T(dx^T x + x^T dx)}{(x^T x)^2} = f'(x)[dx]} \end{aligned}$$

which could be simplified in various ways, but we *cannot* simply ut all of the  $dx$  factors on the right since  $dx x^T \neq x dx^T$  (*very* different from the scalar  $dx^T x = x^T dx$ ).

- (d)  $g(x) = \frac{xx^T}{x^T x}b$ , with vector inputs  $x \in \mathbb{R}^n$  and vector outputs  $f \in \mathbb{R}^n$ , where  $b \in \mathbb{R}^n$  is a constant vector. Give  $g'(x)$  as a **Jacobian** matrix.

**Solution:** We can use the solution from in the previous part since  $g(x) = f(x)b$ , but we can simplify it further because  $dx^T b = b^T dx$ , and  $x^T b$  is a scalar that can be commuted freely, allowing us to move all of the  $dx$  factors to the right:

$$\begin{aligned} dg &= df b = \frac{dx x^T b + x dx^T b}{x^T x} - \frac{xx^T b(2x^T dx)}{(x^T x)^2} \\ &= \underbrace{\frac{1}{x^T x} \left( (x^T b)I + xb^T - 2\frac{xx^T b x^T}{x^T x} \right)}_{g'(x)} dx, \end{aligned}$$

where  $I$  is the  $n \times n$  identity matrix (since  $dx(x^T b) = (x^T b)Id x$ ). This again could be simplified in various ways.

### Problem 3 (5+5+5 points)

- (a) Argue briefly that linear functions that map  $n \times n$  matrices to  $n \times n$  matrices themselves form a vector space  $V$ . What is the dimension of this vector space?

**Solution:** Suppose  $L_1, L_2 \in V$  are two such linear functions. Then this is a vector space if we let  $L = \alpha L_1 + \beta L_2$  be the linear map  $L[X] = \alpha L_1[X] + \beta L_2[X]$  for some scalars  $\alpha, \beta$ —it is clear by inspection that  $L$  satisfies the axioms of linearity if  $L_1, L_2$  do, so this is a vector space (we can add, subtract, and scale).

How many parameters does such a map have? It has  $n^2$  inputs and  $n^2$  outputs, so a linear function has  $\boxed{n^4}$  parameters—we could equivalently write an  $L \in V$  in “vectorized” form as an  $n^2 \times n^2$  matrix multiplying  $\text{vec}(X)$  to produce  $\text{vec}(L[X])$ .

- (b) Argue briefly that linear functions of  $n \times n$  matrices of the form  $X \rightarrow AX$ , where  $A$  is  $n \times n$ , form a vector space. What is the dimension of this vector space?

**Solution:** This is clearly a subspace of  $V$ : if we let  $L_A[X] = AX$ , then by inspection

$$L_{A_1} \pm L_{A_2} = L_{A_1 \pm A_2}$$

and  $\alpha L_A = L_{\alpha A}$  using the definitions above. But it is of dimension  $\boxed{n^2}$ , the number of parameters in the

$n \times n$  matrix  $A$ .

- (c) Argue briefly why it follows that there must be infinitely many linear functions  $\in V$  that are not of the form  $X \rightarrow AX$ .

**Solution:** Since the  $X \rightarrow AX$  functions are an  $n^2$ -dimensional subspace of the  $n^4$ -dimensional  $V$ , it clearly cannot be all of  $V$  unless  $n = 1$ . Indeed, simply counting dimensions we know that there are  $n^4 - n^2 = n^2(n^2 - 1)$  dimensions left.