Spring 2026

CS 5757: Optimization Methods for Robotics
Homework 0: Pre-Knowledge Test

Not Graded

Learning goals for this problem set:

- **Problem 1:** Review eigenvalues and the definition of positive definite matrices.
- **Problem 2:** Review ordinary differential equations (ODEs) and state-space dynamics.
- **Problem 3:** Review fundamentals of vector calculus and derivatives.
- **Problem 4:** Review least-squares fitting and programming fundamentals.

0.1 Eigenvalues and positive definiteness.

Consider the 2×2 matrix

$$\boldsymbol{A} = \begin{bmatrix} 1 & \sigma \\ \sigma & 1 \end{bmatrix}.$$

(i) Find the eigenvalues of $\boldsymbol{A}$ in terms of σ.

(ii) For what values of σ is $\boldsymbol{A}$ invertible?

(iii) Show that $|\sigma| \leq 1$ implies $\boldsymbol{A}$ positive semi-definite, i.e., $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \geq 0$ for any vector $\boldsymbol{x} \in \mathbb{R}^2$.

Hint: Try completing the square, using

$$u^2 + au + b = \left(u + \tfrac{a}{2}\right)^2 + b - \tfrac{a^2}{4}.$$

Solution:

(i) The eigenvalues of a matrix $\boldsymbol{A}$ are given by the roots of $\det(\boldsymbol{A} - \lambda \boldsymbol{I})$, where $\det(\cdot)$ is the matrix determinant, and $\boldsymbol{I}$ is the identity matrix.
For the given matrix $\boldsymbol{A}$, we can compute the determinant directly using the formula for 2×2 matrices:

$$\begin{aligned} \det(\boldsymbol{A} - \lambda \boldsymbol{I}) &= \begin{vmatrix} 1-\lambda & \sigma \\ \sigma & 1-\lambda \end{vmatrix} \\ &= (1-\lambda)^2 - \sigma^2 \end{aligned}$$

We can rearrange this into a quadratic form in terms of λ,

$$\begin{aligned} \det(\boldsymbol{A} - \lambda \boldsymbol{I}) &= 1 - 2\lambda + \lambda^2 - \sigma^2 \\ &= \lambda^2 - 2\lambda + (1 - \sigma^2). \end{aligned}$$

Finally, we can apply the quadratic formula to find the roots of this polynomial,

$$\lambda = \frac{2 \pm \sqrt{4 - 4(1-\sigma^2)}}{2},$$

which, simplifying, yields

$$\boxed{\lambda = 1 \pm \sigma.}$$

(ii) For $\boldsymbol{A}$ to be invertible, it is sufficient for it to have no eigenvalues equal to zero. Thus, $\boldsymbol{A}$ is invertible when

$$\boxed{\sigma \in \mathbb{R} \backslash \{-1, 1\}.}$$

(iii) *Proof.* We begin by expanding the expression $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$ in terms of the entries of $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$.

$$\begin{aligned} \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 1 & \sigma \\ \sigma & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \\ &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} x_1 + \sigma x_2 \\ \sigma x_1 + x_2 \end{bmatrix}, \\ &= x_1^2 + 2\sigma x_1 x_2 + x_2^2. \end{aligned}$$

As suggested in the hint, we complete the square in terms of x_1 (equivalently, x_2) to simplify the expression,

$$\begin{aligned} \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} &= x_1^2 + 2\sigma x_1 x_2 + x_2^2 \pm \sigma^2 x_2^2, \\ &= (x_1^2 + 2\sigma x_1 x_2 + \sigma^2 x_2) + (1-\sigma^2) x_2^2, \\ &= (x_1 + \sigma x_2)^2 + (1-\sigma^2) x_2^2. \end{aligned}$$

By inspection, $(x_1 + \sigma x_2)^2 \geq 0$ and $x_2^2 \geq 0$ for all x_1, x_2. Further, since we assume $|\sigma| \leq 1$, we have $(1-\sigma^2) \geq 0$.

Thus, for any $\boldsymbol{x} \in \mathbb{R}^2$, we have $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \geq 0$, as needed.

□

0.2 Ordinary Differential Equations.

Consider the mass-spring-damper system given in (1). The block has mass $m > 0$ and is rigidly attached to a spring and a damper. Assume the position coordinate p is centered at the equilibrium position. The spring has a spring constant $k > 0$ and provides a force negatively proportional to the positional displacement p. The damper has a damping coefficient $b > 0$ and provides a force negatively proportional to the velocity $\dot{p}$. The equation of motion is

$$m\ddot{p} + b\dot{p} + kp = 0. \tag{1}$$

(i) For appropriate constants $A, B, \omega > 0, \lambda > 0$, what is the general form of the solution to this second-order ordinary differential equation?

(a) $p(t) = A\sin(\omega t) + B\cos(\omega t)$

(b) $p(t) = Ae^{\lambda t}\sin(\omega t) + Be^{\lambda t}\cos(\omega t)$

(c) $p(t) = Ae^{-\lambda t} + Be^{\lambda t}\cos(\omega t)$

(d) $p(t) = Ae^{-\lambda t}\sin(\omega t) + Be^{-\lambda t}\cos(\omega t)$

(e) $p(t) = Ae^{\lambda t}\sin(\omega t) + Be^{-\lambda t}\cos(\omega t)$

(ii) Consider the system state $\boldsymbol{x} = \begin{bmatrix} p \\ \dot{p} \end{bmatrix}$, and suppose $m = 1$. Which state-space system describes the same system dynamics as the equations of motion given in (1)?

(a) $\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 1 & 1 \\ -k & -b \end{bmatrix} \boldsymbol{x}(t)$

(b) $\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & 1 \\ k & b \end{bmatrix} \boldsymbol{x}(t)$

(c) $\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 1 & 1 \\ k & b \end{bmatrix} \boldsymbol{x}(t)$

(d) $\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & 0 \\ -k & -b \end{bmatrix} \boldsymbol{x}(t)$

(e) $\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & 1 \\ -k & -b \end{bmatrix} \boldsymbol{x}(t)$

Solution:

(i) The solution $p(t)$ will take the form given in **(d)**. Intuitively: answers **(b)**, **(c)**, **(e)** have exponential terms $e^{\lambda t}$ which will grow unboundedly in time (since $\lambda > 0$). Answer **(a)** has no exponential terms, meaning it describes a periodic trajectory of fixed amplitude (inconsistent with a non-zero damping coefficient $b > 0$.
Only **(d)** describes the damped oscillations that would describe the trajectory of the system.

(ii) For this definition of $\boldsymbol{x}$, the state-space description is given by **(e)**. To see this, we can construct a system of equations,

$$\dot{\boldsymbol{x}}(t) = \frac{d}{dt}\begin{bmatrix} p \\ \dot{p} \end{bmatrix} = \begin{bmatrix} \dot{p} \\ \ddot{p} \end{bmatrix}.$$

The first entry's dynamics can be described using a linear combination of $\boldsymbol{x}$, trivially $\dot{p} = \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{x}(t)$.
The second entry can be determined by solving (1) for $\ddot{p}$,

$$\begin{aligned} \ddot{p} &= -\frac{k}{m}p - \frac{b}{m}\dot{p}, \\ &= \begin{bmatrix} -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \boldsymbol{x}(t). \end{aligned}$$

Putting these solutions together, with $m = 1$, we get

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & 1 \\ -k & -b \end{bmatrix} \boldsymbol{x}(t),$$

which is the system given in answer **(e)**.

0.3 Vector Calculus.

In this course, we will frequently use the following identities for vectors $\boldsymbol{x}, \boldsymbol{a} \in \mathbb{R}^n$ and matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$:

1. $\nabla_{\boldsymbol{x}}(\boldsymbol{a}^T \boldsymbol{x}) = \boldsymbol{a}$
2. $\nabla_{\boldsymbol{x}}(\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}) = (\boldsymbol{A} + \boldsymbol{A}^T)\boldsymbol{x}$
3. $\nabla_{\boldsymbol{x}}(f(\boldsymbol{g}(\boldsymbol{x}))) = \left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right)^T \nabla_{\boldsymbol{g}} f$

Use these rules to find the symbolic gradient $\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x})$ for the following functions:

(i) $\mathcal{L}(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_r)^T \boldsymbol{Q} (\boldsymbol{x} - \boldsymbol{x}_r)$, where $\boldsymbol{Q} = \boldsymbol{Q}^T$ is symmetric.

(ii) $\mathcal{L}(\boldsymbol{x}) = \|\boldsymbol{R}\boldsymbol{x}\|_2^2$, where $\boldsymbol{R} \in \mathbb{R}^{m \times n}$ is a matrix of weights.

(iii) $\mathcal{L}(\boldsymbol{x}) = \boldsymbol{a}^T \phi(\boldsymbol{x})$, where $\phi : \mathbb{R}^n \to \mathbb{R}^m$ is a nonlinear function. Express the gradient in terms of the Jacobian $\boldsymbol{J}_\phi(\boldsymbol{x}) = \frac{\partial \phi}{\partial \boldsymbol{x}}$.

Solution:

(i) Let $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{x}_r$. Then $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} = \boldsymbol{I}$. Using the chain rule:

$$\nabla_{\boldsymbol{x}} \mathcal{L} = \boldsymbol{I}^T \left[\frac{1}{2} (\boldsymbol{Q} + \boldsymbol{Q}^T)(\boldsymbol{x} - \boldsymbol{x}_r) \right].$$

Since $\boldsymbol{Q}$ is symmetric, $\boldsymbol{Q} + \boldsymbol{Q}^T = 2\boldsymbol{Q}$, yielding:

$$\boxed{\nabla_{\boldsymbol{x}} \mathcal{L} = \boldsymbol{Q}(\boldsymbol{x} - \boldsymbol{x}_r).}$$

(ii) $\mathcal{L}(\boldsymbol{x}) = (\boldsymbol{R}\boldsymbol{x})^T (\boldsymbol{R}\boldsymbol{x}) = \boldsymbol{x}^T (\boldsymbol{R}^T \boldsymbol{R}) \boldsymbol{x}$. Let $\boldsymbol{A} = \boldsymbol{R}^T \boldsymbol{R}$. Since $\boldsymbol{A}$ is symmetric by construction:

$$\boxed{\nabla_{\boldsymbol{x}} \mathcal{L} = 2\boldsymbol{R}^T \boldsymbol{R} \boldsymbol{x}.}$$

(iii) Using the first rule and the chain rule:

$$\boxed{\nabla_{\boldsymbol{x}} \mathcal{L} = \boldsymbol{J}_\phi(\boldsymbol{x})^T \boldsymbol{a}.}$$

0.4 Least squares via JAX.

Suppose you are performing "system identification" and aim to find a best-fit model of a motor's torque output as a function of its input current. To do this, you collect a dataset $\{(I_k, \tau_k)\}_{k=1}^N$ of paired input currents I_k and output torques τ_k.

You hypothesize that the output torque has a quadratic relationship to the input current:

$$\tau \approx f(I; \boldsymbol{x}) = x_0 + x_1 I + x_2 I^2,$$

where $\boldsymbol{x} = \begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix}^T \in \mathbb{R}^3$ is the vector of parameters defining the motor model.

This question will use the code distributed with this assignment to implement the least-squares methods we discuss below.

We have included a `README.md` in the distributed code to help with setting up and running the environment. For help with `JAX` or `numpy` basics, we suggest you work through some of the tutorials we have linked on the course webpage.

If you would like to make sure you can submit your code and run the autograder successfully, feel free to submit to the HW0 assignment on Gradescope. Please note this is just for you to try a "dry run" of homework submission - the assignment will not count (either for or against) your final grade.

(i) Formulate the problem of finding the best-fit model parameters, i.e., those that minimize the mean-squared error $\ell(x_0, x_1, x_2) = \frac{1}{N}\sum_{k=1}^{N} \left(\tau_k - f(I_k; x_0, x_1, x_2)\right)^2$, as a least squares problem

$$\min_{\mathbf{x}} ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2.$$

Specifically, describe how to form $\mathbf{A}, \mathbf{y}$ from the problem data $\{(\tau_k, I_k)\}_{k=1}^{N}$.

(ii) Implement the function `get_problem_data` in `solutions.py`. The function should take the current data `I`, with shape `(N,)` and the torque data `tau`, shape `(N,)`, and return the least-squares problem data `(A, y)` as a tuple.
Your function should:
- Return `A, y` with the proper shape and ordering.
- Be composable with `jax.jit`.

(iii) Implement the function `least_squares_loss` which maps a current guess `x`, shape `(3,)`, along with the problem data `A, y`, of the proper shape, and returns the mean-squared error ℓ.
Your function should:
- Return a scalar.
- Be composable with `jax.jit` and `jax.grad`.
- Be vectorized (i.e., use only array operations (broadcasting, matmul/einsum).)

(iv) Finally, implement the function `lstsq_params_closed_form` which maps problem data `I, tau`, both shape `(N,)`, to the closed-form optimal solution $\boldsymbol{x}^*$ to the least-squares problem

(v) In the notebook `hw0.ipynb`, we provide a simple gradient descent loop which starts from an initial guess $\boldsymbol{x}_0 = \mathbf{0}$ and takes gradient steps on `least_squares_loss` to find an approximate solution to the least squares problem.
We provide plotting code for you to visualize the closed-form and numerical solutions obtained by gradient descent, as well as the loss curve obtained over the course of gradient descent.
In your writeup, include a plot of the gradient descent solution versus your analytic solution.

(vi) Try modifying the learning rate in the gradient descent code. In your own words, in your writeup, explain how the learning rate affects convergence of your method.