Math Worksheet, Computational Neuroscience 2024

- This worksheet contains examples mathematics problems for the midterm.
- It is not a practice exam; It does not address *everything*; There will be further questions about the math in courseworks 1, 2.
- The exam format will be 2–5 multiple-choice or short-answer “fact” and “understanding” questions, followed by 3 in-depth multi-part mathematical questions. Marks from the best 2-out-of-3 will be used.
- If you get stuck or something seems weird, it’s probably a typo (email to report).

Q1: Consider the ODE

$$\dot{v} = g \cdot (D - v) + B, \tag{1}$$

where g, D, and B are constants. Show that

$$v(t) = \alpha(t) \cdot v_0 + [1 - \alpha(t)] \cdot v_\infty \tag{2}$$

is a solution to ODE (1) with initial conditions $v(t = 0) = v_0$, where

$$\begin{aligned} \alpha(t) &= e^{-gt} \\ v_\infty &= D + B/g \end{aligned} \tag{3}$$

(Any approach permitted).

Approach 1: Verify

This is meant to be a faster approach for students less-practices in solving ODEs directly, but this is not the only or best method — any other approach is fine. We are given

$$\begin{aligned} v(t) &= \alpha(t) \cdot v_0 + [1 - \alpha(t)] \cdot v_\infty \\ &= v_\infty + e^{-gt}(v_0 - v_\infty); \end{aligned} \tag{4}$$

Treat this as an ansatz and differentiate to verify:

$$\begin{aligned} \dot{v} &= -ge^{-gt}(v_0 - v_\infty) \\ &= -g(v - v_\infty) \\ &= -g[v - (D + B/g)] \\ &= g[D + B/g - v] \\ &= g(D - v) + B \quad \blacksquare \end{aligned} \tag{5}$$

Approach 2: Using integrating factor. Your response can be brifer than this one.

Arrange (1) to apply integrating factor method:

$$\begin{aligned} \dot{v} &= g \cdot (D - v) + B \\ \dot{v} + gv &= \underbrace{gD + B}_{gv_\infty} \end{aligned} \tag{6}$$

The integrating factor is $\exp(\int_{dt} g) = e^{gt}$; multiply through:

$$\begin{aligned} e^{gt}[\dot{v} + gv] &= e^{gt} g v_\infty \\ e^{gt}\dot{v} + g e^{gt} v &= e^{gt} g v_\infty \\ \frac{\mathrm{d}}{\mathrm{d}t}[e^{gt} v] &= e^{gt} g v_\infty \end{aligned} \tag{7}$$

Integrate both sides and collect terms:

$$\begin{aligned} \int_{dt} \frac{\mathrm{d}}{\mathrm{d}t}[e^{gt} v] &= \int_{dt} e^{gt} g v_\infty \\ e^{gt} v(t) &= v_\infty e^{gt} + c \\ v(t) &= v_\infty + c e^{-gt} \end{aligned} \tag{8}$$

Evaluate at $t = 0$ to get constant c:

$$\begin{aligned} v_0 &= v_\infty + c e^{-g \cdot 0} = v_\infty + c \\ c &= v_0 - v_\infty \\ v &= v_\infty + \underbrace{e^{-gt}}_{\alpha(t)}(v_0 - v_\infty) \\ v &= v_\infty + \alpha(t) v_0 - \alpha(t) v_\infty \\ v &= \alpha(t) v_0 + [1 - \alpha(t)] v_\infty \quad \blacksquare \end{aligned} \tag{9}$$

Q2: Provide the solution to $\tau\dot{v} = c - v$ and evaluate $v(t)$ at time $t = 10$ for $c = 0$, $\tau = 10$, and $v(0) = -70$.

The first part ("provide solution") is similar to Q1 and we are not asked to do redundant work. If you can intuit the solution immediately, simply write it. Otherwise, any approach valid for Q1 will also get you here. If you're not 100% confident, you can differentiate to verify.

$$v(t) = c + e^{-t/\tau}[v(0) - c]$$

Evaluate at given values:

$$\begin{aligned} v(10) &= 0 + e^{-10/10}[(-70) - 0] && \text{(plugging in constants' values)} \\ &= -\tfrac{1}{e} \cdot 70 && \text{(evaluating)} \end{aligned} \tag{10}$$

There's no need to use a calculator to go any further (a correct numeric answer with incorrect algebra might raise suspicion). Any variation algebraically equivalent to the above will get full marks.

Note: this reminds us that the time constant τ is the time it takes for the distance between $v(0)$ and c to decay to $1/e$ of its original value.

Q3:

You are asked to write computer code to numerically integrate a linear first-order ODE for a neuron's membrane voltage v, being driven by a known time-varying input $u(t)$

$$C\dot{v} = \frac{1}{R}(E - v) + u(t), \tag{11}$$

where C, R, and E are constants.

Q3a: (facts/trivia) Is forward Euler a first or second-order method?

First

Q3b: (facts/trivia) What is the order of the error in forward Euler in terms of the time step Δt?

O(Δt^2)

(CONTINUED NEXT PAGE)

Q3c: You're told that the signal $u(t)$ varies slowly, and is approximately constant $u(t) \approx u_t$ over a short duration between t and $t + \Delta t$, where $\Delta t = 1$ ms. You implement the following (pseudo)code to integrate from $t = 0$ to $t = 1000$ ms based on this assumption:

$$
\begin{aligned}
&\gamma = ???\\
&v = v_0\\
&\text{for } t \text{ in } 1 \ldots 1000 :\\
&\qquad v_\infty = ???\\
&\qquad \Delta_v = \gamma(v_\infty - v_t)\\
&\qquad v_{t+1} = v_t + \Delta_v
\end{aligned}
\tag{12}
$$

What expressions belong in the ??? for variables γ and v_∞?

Any route to the correct answer is fine; no need to "prove" or integrate from scratch. You may use any notation (or even words) to express yourself, define intermediate variables, etc. Correct answers without work get full marks (but memorization won't work—this question is not on the exam as such, but does test your readiness).

Solve assuming constant $u(t) \approx u_t$:

$$
\begin{aligned}
C\dot{v} &= \tfrac{1}{R}(E - v) + u_t\\
\underbrace{RC}_{\tau}\,\dot{v} &= \underbrace{E + Ru_t}_{v_\infty} - v\\
v(t) &= v_\infty + e^{-t/\tau}(v - v_\infty)
\end{aligned}
\tag{13}
$$

Every time-step solve forward by Δt using a new u_t:

$$
\begin{aligned}
v_{t+1} &= v_\infty + \underbrace{e^{-\Delta_t/\tau}}_{\alpha}(v_t - v_\infty)\\
\Delta_v = v_{t+1} - v_t &= v_\infty + \alpha(v_t - v_\infty) - v_t\\
&= \underbrace{(1-\alpha)}_{\gamma}(v_\infty - v_t)
\end{aligned}
\tag{14}
$$

Full marks for any technically valid response, even if it is not an exact match to the one above.

Q4: Consider a linear-nonlinear perceptron "neuron" $\hat{y} = \phi(\boldsymbol{w}^\top \boldsymbol{x})$, with inputs $\boldsymbol{x}$, weights $\boldsymbol{w}$ (both column vectors), and output $\hat{y}$.

Q4a: (bookwork / concepts) In coursework 2, we covered the delta learning rule in a context where $\phi(\cdot)$ was a hard-threshold, setting $\hat{y} = 1$ for activation $\boldsymbol{w}^\top \boldsymbol{x} > 0$ and -1 otherwise. *State the weight update equation for the delta rule, and explain how we can view the rule as learning based on the prediction error $y^* - \hat{y}$.*

This answer is longer than needed for full marks. The wording and content does not need to be an exact match to the answer below: Any sincere attempt demonstrating understanding gets full marks.

$$\Delta w_i = \eta \underbrace{(y^* - \hat{y})}_{\text{error}} x_i \tag{15}$$

We can interpret this as the "neuron" comparing a prediction $\hat{y}$ to a supervised target y^*, and calculating the error $y^* - \hat{y}$, which is then correlated with the synaptic inputs. For example, a positive $y^* - \hat{y}$ means our output should have been larger. If an input x_i is positive, we should increase its weight w_i to try to increase $\hat{y}$ and reduce this error for similar inputs $\boldsymbol{x}$ and target output y^* in the future.

(CONTINUED NEXT PAGE)

Q4a: Consider a version of linear-nonlinear perceptron that uses $\phi(\cdot) = \exp(\cdot)$ for its nonlinearity. Show that we can recover a learning rule similar to the delta rule by gradient-descent optimization of the following loss function:

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}; \boldsymbol{x}, y^*) &= \hat{y} - y^* \cdot \ln(\hat{y}) \\ \hat{y} &= \phi(r) = e^r \\ r &= \boldsymbol{w}^\top \boldsymbol{x} = \textstyle\sum_i w_i x_i. \end{aligned} \tag{16}$$

$$\begin{aligned} \mathcal{L}(\dots) &= \hat{y} - y^* \cdot \ln(\hat{y}) \\ &= e^r - y^* \cdot r \\ \tfrac{\mathrm{d}}{\mathrm{d}w_k} \mathcal{L}(\dots) &= \tfrac{\mathrm{d}}{\mathrm{d}r} [e^r - y^* \cdot r] \cdot \tfrac{\mathrm{d}r}{\mathrm{d}w_k} \\ &= (e^r - y^*) \cdot x_k \\ &= (\hat{y} - y^*) \cdot x_k \end{aligned} \tag{17}$$

We want to minimize the loss, so a gradient descent update would be the negative of this gradient times some positive learning rate η

$$\Delta w_k = \eta (y^* - \hat{y}) x_k \tag{18}$$

This is the same as the delta rule for the binary perceptron (with the exception that the prediction $\hat{y}$ is calculated differently.)

Full marks for any answer that differentiates $\mathcal{L}$ in terms of the weights and notes similarity to the delta rule. Students may use any notation they wish, e.g. keeping this a vector derivative, etc. A cheeky response might simply be to state $-\frac{\mathrm{d}}{\mathrm{d}w_k}\mathcal{L}(\dots) = (y^* - \hat{y})x_k$. This would not receive full marks unless accompanied by some intermediate steps or notes hinting that the student has verified that it is, at least, true (by inspecting the derivative of $\mathcal{L}$).

Another derivation of the integrating factor

We are interested in approximating the time-evolution of a neuron's membrane voltage as a first-order ODE, driven by inputs $u(t)$.

$$\tau \frac{\mathrm{d}v(t)}{\mathrm{d}t} = u(t) - v(t). \tag{19}$$

Let's re-arrange (20) into the form used for the integrating factor, which is

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} + f(t)x(t) = g(t). \tag{20}$$

Divide (19) by τ and move the voltage decay to the left-hand side to obtain something similar to (20):

$$\frac{\mathrm{d}v(t)}{\mathrm{d}t} + \frac{1}{\tau}v(t) = \frac{1}{\tau}u(t). \tag{21}$$

We find that the integration factor is $e^{\int_{\mathrm{d}t} \frac{1}{\tau}} = e^{t/\tau}$; This is an indefinite integral, and we can omit the constant of integration (or rather, set it to zero), since all choices of constant will give the same result in the algebra that follows. Multiplying (21) through by this gives

$$e^{t/\tau}\frac{\mathrm{d}v(t)}{\mathrm{d}t} + \frac{1}{\tau}v(t)e^{t/\tau} = \frac{1}{\tau}u(t)e^{t/\tau}. \tag{22}$$

Recognizing the chain rule for the exponential, we then write (22) as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left\{v(t)e^{t/\tau}\right\} = \frac{1}{\tau}u(t)e^{t/\tau}. \tag{23}$$

The notation in the coursenotes caused confusion for the next steps, so I will use a slightly different style. The next step is to take the indefinite integral of both sides of (23)

$$\begin{aligned}
\int_{\mathrm{d}t} \tfrac{\mathrm{d}}{\mathrm{d}t}\left\{v(t)e^{t/\tau}\right\} &= \int_{\mathrm{d}t} \tfrac{1}{\tau}u(t)e^{t/\tau} && \\
v(t)e^{t/\tau} + c_1 &= \tfrac{1}{\tau}\int_{\mathrm{d}t} u(t)e^{t/\tau} && \text{integrate} \\
v(t)e^{t/\tau} &= \tfrac{1}{\tau}\int_{\mathrm{d}t} u(t)e^{t/\tau} - c_1 && \text{move constant to RHS} \\
v(t) &= e^{-t/\tau}\left(\tfrac{1}{\tau}\int_{\mathrm{d}t} u(t)e^{t/\tau} - c_1\right) && \text{multiply both sides by } e^{-t/\tau}
\end{aligned} \tag{24}$$

Let's explore the result in (24) for the case where $u(t) = u$ is a constant.

$$\begin{aligned}
v(t) &= e^{-t/\tau}\left(\tfrac{1}{\tau}\int_{\mathrm{d}t} ue^{t/\tau} - c_1\right) \\
&= e^{-t/\tau}\left(u\int_{\mathrm{d}t} \tfrac{1}{\tau}e^{t/\tau} - c_1\right) \\
&= e^{-t/\tau}\left(u\left[e^{t/\tau} + c_2\right] - c_1\right) \\
&= e^{-t/\tau}\left(ue^{t/\tau} + \underbrace{uc_2 - c_1}_{c}\right) \\
&= e^{-t/\tau}\left(ue^{t/\tau} + c\right) \\
&= u + ce^{-t/\tau}
\end{aligned} \tag{25}$$

Above, we have been very explicit about how the constants arising from taking integrals are shuffled about and consolidated. Some sources elide these steps into a single "c" that simply stands in for "$+\mathcal{O}(1)$", rather than something with a specific value. You may also see some sources write "$+$ constant" or "$+$ const." for "$+\mathcal{O}(1)$" as well.

We now have a family of curves for $v(t)$ parameterized by c. Knowing the value of $v(t)$ at some specified time is enough to find c. We typically work with the initial value problem where $v(0) = v_0$ is known.

$$v_0 = v(0) = u + ce^{-0/\tau} \qquad \Rightarrow \qquad c = v_0 - u \tag{26}$$

giving

$$v(t) = u + e^{-t/\tau}(v_0 - u). \tag{27}$$

Why the integration factor is overkill for 1^{st} order linear ODEs with constant forcing

There is something tautological in using the integration factor (which relies on the properties of the derivatives of $\exp(\cdot)$) to solve scalar first-order ordinary differential equations whose solutions are also, essentially, $\exp(\cdot)$.

We can verify that $\frac{d}{dt}\{ce^t\} = ce^t$ by inspecting the Taylor series of e^t (among other methods). This confirms that $x(t) = ce^t$ parameterizes a family of solutions to the ODE $\dot{x} = x$. This fact is already needed to define the integrating factor approach (which combines the derivative of the exponential with the chain rule), so it is no less elementary to start here for first-order ODEs.

If we are solving an initial value problem were $x(0) = x_0$, solving $x_0 = ce^0$ for c gives $c = x_0$, and yields the familiar solution $x(t) = x_0e^t$. Solutions to slightly more generic first-order scalar ODEs can be obtained by scaling (in t) and shifting (in x) this elementary solution:

Scale: If we define a new time scale $t = -\tilde{t}/\tau$ (where $\tilde{t}$ is the original scale), the ODE becomes $\tau\dot{x}(t) = -x(t)$, and the initial-value-problem solution $x(t) = e^{-t/\tau}x_0$. This can be intuited as a change in the units we are using to count time.

Shift: If we further define $v = x+c$ (so that $x = v-c$), then $\frac{d}{dt}v = \frac{d}{dt}x$ and we have $\frac{d}{dt}v = -x/\tau = (c-v)/\tau$ or, in other words $\tau\dot{v} = c - v$. The solution $x(t) = e^{-t/\tau}x_0$ becomes $v(t) - c = e^{-t/\tau}(v_0 - c)$, that is $v(t) = c + e^{-t/\tau}(v_0 - c)$. This is familiar form where the solution decays exponentially from its initial value v_0 toward the asymptotic value c.

Thus, we do not need the integrating factor if the driving term for the first-order ODE is a constant. True, solutions are more involved if the input is time-varying, but here it starts to make more sense to apply the Fourier-Laplace domain approach to linear dynamical systems, which treat linear ODEs as a filter applied to their inputs.

Of course, the integrating factor approach is absolutely fine. A student using it appropriately on an exam will get full marks, and many SEMT students are comfortable with it because it is familiar and well-practiced.