COURSE NOTES

Control Co-Design

Integrated Optimization of Physical and Control Systems

Compiled on Monday 13th July, 2026
Last repo commit: Git metadata unavailable

Note: Interactive elements may be simplified or omitted in this PDF export.

About These Notes

Control Co-Design: Integrated Optimization of Physical and Control Systems

Google Chrome Recommended

For the best experience, please view these notes using **Google Chrome** (or another Chromium-based browser, like Arc). You may experience slower scrolling and performance issues if using Safari.

Overview

Control co-design integrates physical-system, controller, sensing, actuation, information, and architecture decisions. The course develops the theory and computational methods needed to design dynamic engineering systems as coordinated wholes rather than as a sequence of disciplinary handoffs.

The complete course is organized into five parts and twenty planned chapters:

1. **Foundations:** motivation, feedback, optimization, and a unified CCD formulation.
2. **Formulations and architectures:** coordination methods, open- and closed-loop CCD, component architecture, and lifecycle objectives.
3. **Computational methods:** numerical optimal control, sparse optimization, surrogate modeling, uncertainty, and diagnostics.
4. **Application studies:** active suspension, wind energy, marine and hybrid energy, and cross-domain systems.
5. **Deployment and frontiers:** controller realization, hardware validation, reproducibility, and future research.

Each chapter consists of several pages, called "sections", so Chapter 1.2 (for example) is a section.

Prerequisites

Readers should be comfortable with differential equations, linear algebra, introductory control systems, numerical methods, and basic optimization. Examples are provided in both Python and MATLAB.

How This Course Is Organized

Each chapter begins with learning objectives and a motivating engineering problem, develops the governing theory and mathematical formulation, and closes with implementation guidance, common mistakes, exercises, and an open research question.

Examples appear at three levels: compact teaching systems such as a mass-spring-damper, a recurring active-suspension benchmark, and capstone applications in wind energy, marine energy, aerospace, robotics, and electrified transportation. The emphasis throughout is not only on obtaining an optimum, but on understanding why the design changed and whether the improvement survives uncertainty and implementation.

Sources and Reproducibility

The course draws on the control co-design, multidisciplinary design optimization, optimal-control, robust-design, and application literature. Each chapter identifies its principal sources and distinguishes published results from original teaching examples. Figures are redrawn as original vector graphics, while numerical plots are intended to be regenerated from version-controlled Python and MATLAB code.

Accessibility

The course aims to meet WCAG 2.1 AA practices. Figures include descriptive alternative text, color is not the only carrier of meaning, and equations and code remain available as text.

Advice to the Student

Control co-design is learned by moving repeatedly between physical reasoning, mathematical formulation, computation, and design interpretation. Do not treat the optimization solver as a black box.

For each chapter:

1. identify the physical and control decisions before reading the formulation;
2. predict the important coupling mechanisms;
3. reproduce the worked example in Python or MATLAB;
4. check units, constraints, scaling, and numerical convergence;
5. explain why the optimized design changed; and
6. test the result under conditions that were not optimized.

The activities and exercises are part of the course narrative. A converged optimization is only the beginning of the engineering argument.

Why Control Co-Design?

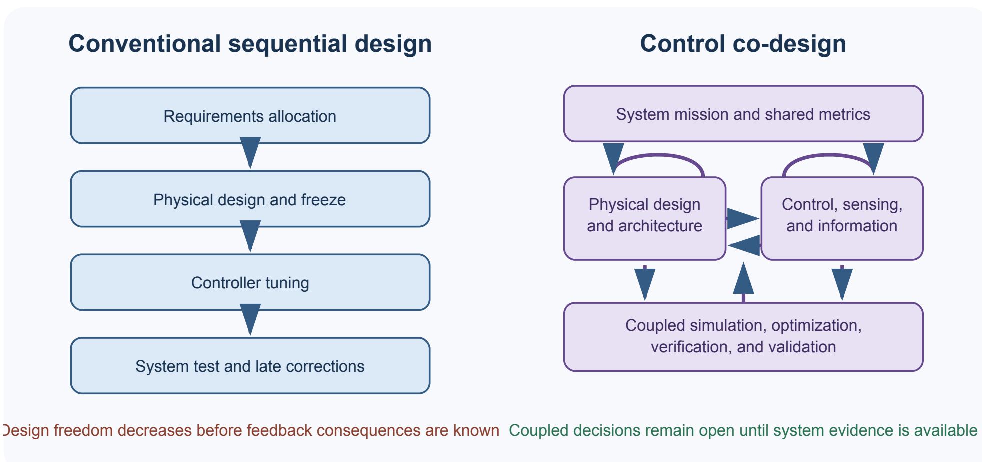
1.1 Conventional Sequential Engineering Design

The familiar handoff

Many organizations divide a dynamic-system project into a chain of discipline-specific tasks:

1. requirements are allocated;
2. mechanical, electrical, hydraulic, or thermal hardware is designed;
3. actuators and sensors are selected;
4. the physical design is frozen;
5. a model is handed to the controls team; and
6. the controller is tuned to recover as much system performance as possible.

This workflow is understandable. It aligns with departmental boundaries, supplier contracts, design reviews, and established simulation tools. It also reduces coordination: each team sees a smaller problem with fewer variables.



The simplification, however, comes from an assumption that is rarely tested: **physical design can be completed before the consequences for feedback are known**.

The quarter-car handoff

For the suspension, a sequential plant team might choose k_s and c_s by minimizing a passive metric

$$J_p(k_s, c_s) = w_a\, a_{\mathrm{rms}}^2 + w_t\, \delta_{t,\mathrm{rms}}^2 + w_m\, m,$$

subject to packaging and stress limits. After those values are frozen, the controls team tunes gains K to minimize

$$J_c(K \mid k_s^*, c_s^*)$$

subject to actuator force, suspension travel, stability, and bandwidth constraints. The vertical bar matters: the controller is optimized **conditional on a plant selected for another problem**.

The plant team may have made the spring soft to improve passive comfort. The controller then needs more actuator stroke or low-frequency force to regulate body motion. Alternatively, a stiff plant may protect suspension travel while moving the controlled system toward uncomfortable accelerations. Neither team has authority to revisit both sides of the trade.

Sequential design is not automatically poor engineering

Sequential design can be effective when interfaces are genuinely weak, a mature platform must be reused, certification freezes the plant, or a standard controller already offers ample margin. Its weakness is not the ordering alone. Its weakness is **prematurely removing decisions that strongly affect the system-level optimum**.

When Sequential Design Is Sufficient

A sequential process is defensible when changing controller decisions barely changes the preferred plant, changing plant decisions barely changes achievable closed-loop performance, and the final design retains comfortable feasibility margins under uncertainty.

Hidden assumptions in a handoff

A sequential process often embeds assumptions without stating them:

- the selected actuator can deliver the required force, rate, and bandwidth;
- the chosen sensors reveal the states needed for feedback;
- the plant model remains valid over the controller's operating range;
- control effort has negligible effects on mass, heat, energy, and cost;
- saturation and delay will not reshape the preferred physical design; and
- failure or degraded-control modes remain acceptable.

CCD begins by turning these assumptions into variables, constraints, and testable hypotheses.

Activity 1.1: audit a sequential workflow

Activity 1.1: audit a sequential workflow

Choose a controlled product you know. Draw its likely discipline handoffs. Circle every decision frozen before closed-loop performance is evaluated. For each circled decision, name one control consequence and one physical consequence of changing it.

1.2 The Physical Plant and Controller as a Single System

The closed-loop model is the design object

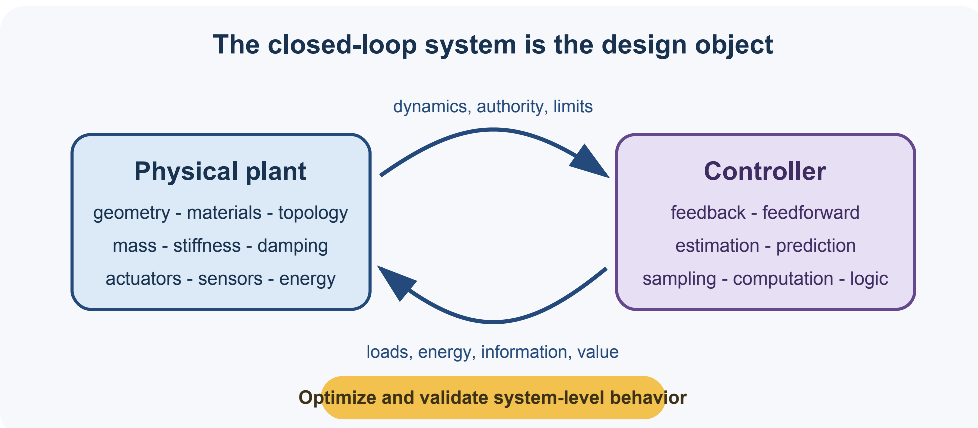
Consider a parameterized linear plant

$$\dot{x}(t) = A(p)x(t) + B(p)u(t) + E(p)d(t), \qquad y(t) = C(p)x(t),$$

where p contains physical decisions such as geometry, stiffness, mass distribution, motor constants, or actuator placement. With state feedback $u(t) = -K(c)x(t)$, where c denotes controller parameters, the closed-loop dynamics are

$$\dot{x}(t) = [A(p) - B(p)K(c)]\, x(t) + E(p)d(t).$$

The system poles, transient response, disturbance amplification, and control effort are properties of $A(p) - B(p)K(c)$—not of A, B, or K in isolation. Physical variables may alter both natural dynamics $A(p)$ and control authority $B(p)$. Controller variables then reshape how those physical properties appear in operation.



Physical meaning of the coupling

The integrated view reveals recurring mechanisms:

Physical decision	Control consequence	System consequence
Lower structural mass	Faster motion, often more flexible modes	Efficiency may improve while robustness margins shrink
Larger actuator	More force and bandwidth	Better regulation but higher mass, heat, and cost
Higher gear ratio	More output torque	Reduced speed range and reflected inertia
Softer suspension spring	Better passive isolation	More travel and low-frequency control demand
Sensor relocation	Different signal quality and modal visibility	Estimation quality and achievable damping change

The controller also changes what constitutes a good plant. Strong feedback can reduce the need for passive damping, but only if sensing, authority, bandwidth, energy, and reliability are adequate. Preview control can favor a different suspension than purely reactive feedback. A robust controller may prefer physical designs with less nominal performance but smaller uncertainty amplification.

System boundaries matter

Calling the plant and controller "one system" does not mean every variable must be optimized at once. It means the model boundary must include the consequences needed for a valid decision. For an electric actuator, that boundary may include power electronics, thermal dynamics, a battery, communication delay, and a supervisory controller. Omitting them can make an apparently optimal design physically impossible.

CCD Insight

The correct object of comparison is usually not an uncontrolled plant metric plus a controller metric. It is a system-level metric evaluated on the coupled closed-loop or optimally controlled dynamics.

Three kinds of decisions

It is useful to separate:

- **plant decisions** p: geometry, materials, component ratings, topology;
- **control decisions** c and $u(\cdot)$: gains, estimator settings, horizons, or trajectories; and
- **information decisions** s: sensors, sampling, communication links, preview, and model availability.

A compact system objective is therefore

$$J = J(p, c, s, x(\cdot), u(\cdot), d(\cdot)),$$

with dynamics and constraints determining which combinations are feasible.

Activity 1.2: move the boundary

Activity 1.2: move the boundary

For the quarter-car suspension, first draw a boundary containing only masses, spring, damper, and actuator. Then enlarge it to include sensors, computation, electrical power, and thermal limits. List two design conclusions that could change when the boundary expands.

1.3 Plant-Control Design Coupling

What coupling means

Plant-control coupling exists when a decision on one side changes the value, feasibility, or preferred setting of decisions on the other side. Coupling may enter through dynamics, constraints, objectives, information, or architecture.

For the quarter car, let $p = [k_s, c_s, F_{\max}]^T$ and let c contain feedback gains. A combined objective might be

$$J(p, c) = w_a \int_0^T \ddot{z}_s(t)^2\, dt + w_u \int_0^T u(t)^2\, dt + w_m m_a(F_{\max}),$$

subject to

$$|z_s - z_u| \le z_{\max}, \qquad |u| \le F_{\max},$$

and the closed-loop equations. The actuator rating is simultaneously a plant variable, a control bound, a mass/-cost contributor, and a feasibility mechanism. That is strong coupling.

Five coupling channels

1. **Dynamic coupling:** plant variables change poles, zeros, modes, and nonlinear behavior.
2. **Authority coupling:** actuator size and placement change reachable forces or moments.
3. **Information coupling:** sensor choices change observability, noise, and delay.
4. **Constraint coupling:** control action activates physical limits such as stress, temperature, travel, or power.
5. **Economic coupling:** control hardware and operating effort affect capital and lifecycle cost.

Weak and strong coupling

Coupling is not binary. A practical diagnostic asks whether the controller-optimal design map

$$c^*(p) = \arg\min_c J(p, c)$$

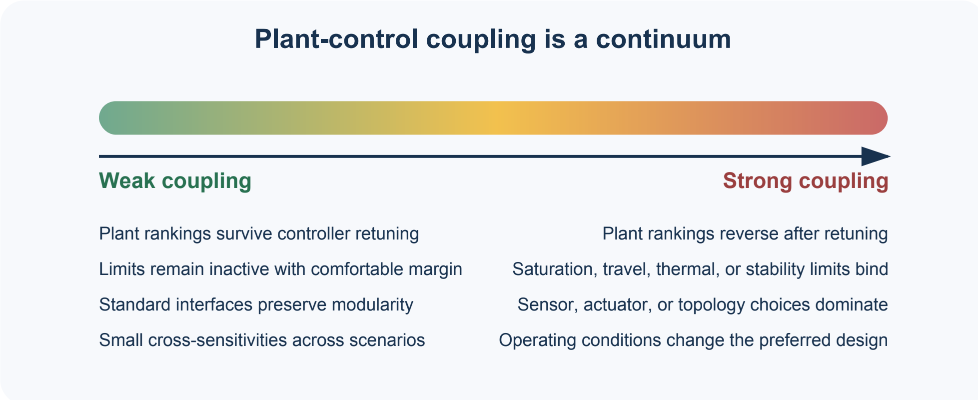
changes substantially over the physical design space, and whether the reduced objective

$$\widehat{J}(p) = J\left(p, c^*(p)\right)$$

has a different minimizer than a plant-only objective. Large cross-sensitivities

$$\frac{\partial^2 J}{\partial p\, \partial c}$$

are one clue, but active constraints and discrete architecture changes can create important coupling even when local derivatives look small.



Physical Interpretation

Strong coupling often appears near a bottleneck: actuator saturation, a flexible mode, a travel limit, a thermal ceiling, a sensor blind spot, or a stability boundary. If feedback never approaches a plant-dependent limit, integrated optimization may offer little improvement.

Coupling is operating-condition dependent

A plant and controller may be weakly coupled in routine operation but strongly coupled during gusts, maneuvers, faults, or extreme sea states. A suspension sized on smooth roads may appear insensitive to actuator authority; the same design may saturate over a sharp road event. CCD studies must therefore include the operating conditions that drive design decisions.

A first coupling test

Before launching a large optimization:

1. choose several plausible plant designs;
2. retune the controller for each design;
3. record system performance and active constraints;
4. compare rankings before and after retuning; and
5. perturb actuator and information assumptions.

If plant rankings reverse, active constraints migrate, or the controller changes sharply, the problem is a strong CCD candidate.

Activity 1.3: identify the coupling channel

Activity 1.3: identify the coupling channel

Classify each interaction as dynamic, authority, information, constraint, or economic coupling: (a) a longer robot link reduces its first natural frequency; (b) a camera adds 40 ms latency; (c) regenerative braking heats the battery; (d) a larger motor increases vehicle mass. Some interactions belong to more than one class—explain why.

1.4 Concurrent Engineering and the Origins of CCD

From handoffs to concurrent decisions

Concurrent engineering keeps interacting disciplines engaged early enough to influence one another. CCD specializes that idea for dynamic systems: subsystem dynamics, control objectives, sensing, actuation, and implementation constraints are considered from the beginning rather than appended after the physical design is mature.

Integrated structure-control design in aerospace, combined passive-active suspension design, mechatronic optimization, optimal actuator placement, and multidisciplinary dynamic-system optimization all contributed to modern CCD. The terminology varies across fields, but the recurring goal is the same: preserve system-level design freedom where dynamics and feedback interact.

CCD is broader than one optimization problem

Garcia-Sanz organizes CCD into three complementary perspectives.

1. Control-inspired engineering design. Control knowledge guides physical invention and architecture before a formal optimizer is built. Engineers may reshape a structure to improve controllability, introduce passive load paths that cooperate with feedback, relocate sensors, separate time scales, or create a mechanism whose natural dynamics reduce control effort.

This perspective preserves physical intuition and can produce architecture changes that continuous parameter optimization would never discover.

2. Formal mathematical co-optimization. Plant variables, controller parameters or trajectories, states, and constraints are placed in an integrated mathematical program. The formulation may be nested, simultaneous, continuous, mixed-integer, deterministic, or uncertain. This perspective makes tradeoffs explicit and provides repeatable numerical evidence.

3. Model- and data-based co-simulation. Multiphysics, multiscale, or mixed-fidelity simulations evaluate candidate system designs. Experimental data, hardware tests, or learned models may augment the physics. Co-simulation is especially valuable when a compact analytic model cannot preserve the interactions that determine the design.

Perspective	Primary strength	Typical limitation
Control-inspired design	Creativity and physical interpretation	May not quantify global tradeoffs
Co-optimization	Formal, repeatable system-level tradeoffs	Sensitive to formulation and numerical quality
Co-simulation	Captures complex interactions and data	Can be computationally expensive and opaque

CCD Insight

The strongest projects combine all three perspectives: control reasoning proposes better architectures, optimization coordinates variables, and simulation or experiments challenge the resulting design.

A concise historical arc

- **Classical control era:** controllers are designed for largely fixed plants.
- **Integrated structure-control work:** structural parameters, actuator placement, and feedback begin to be coordinated.
- **Mechatronic and multidisciplinary optimization:** physical and control variables enter shared system studies.
- **Dynamic optimization and direct transcription:** time histories become optimization variables in sparse nonlinear programs.
- **Modern CCD:** architecture, uncertainty, information availability, high-fidelity simulation, and data are integrated.

This history cautions against defining CCD as merely "put plant parameters and gains in the same vector." CCD is a concurrent engineering discipline supported—but not exhausted—by optimization.

Sources and further reading

- M. Garcia-Sanz, "Control Co-Design: An Engineering Game Changer," *Advanced Control for Applications*, 2019.
- J. T. Allison and D. R. Herber, "Multidisciplinary Design Optimization of Dynamic Engineering Systems," *AIAA Journal*, 2014.

1.5 What Performance Opportunities Sequential Design Misses

Conditional optima are not system optima

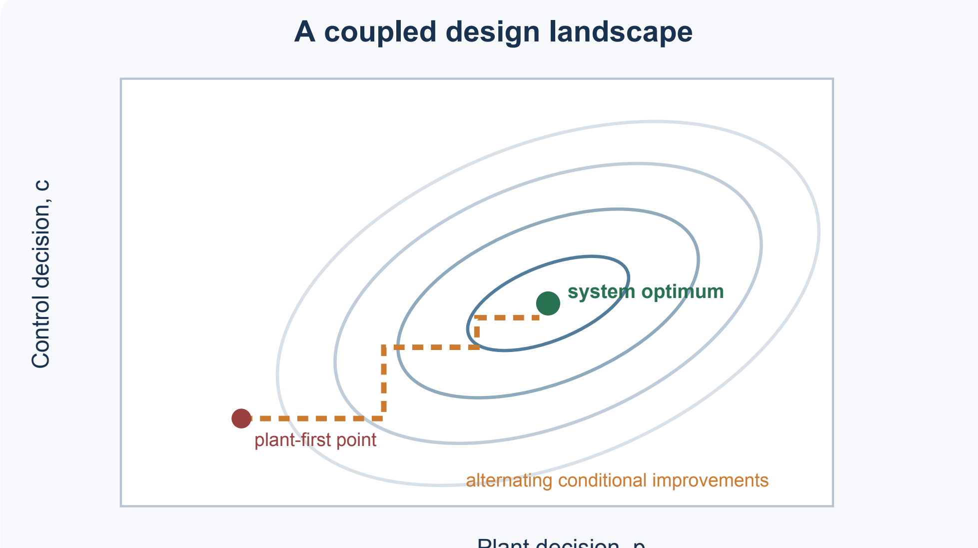
Suppose p is a plant variable and c is a controller variable. A sequential process computes

$$p_{\text{seq}} = \arg\min_{p} J_p(p), \qquad c_{\text{seq}} = \arg\min_{c} J(p_{\text{seq}}, c).$$

CCD instead computes

$$(p_{\text{ccd}}, c_{\text{ccd}}) = \arg\min_{p,c} J(p, c)$$

subject to the same system dynamics and constraints. Even if both sequential subproblems are solved perfectly, their pair need not minimize the combined objective.



Contours are conceptual; actual CCD landscapes may be constrained, nonsmooth, and multimodal.

Four kinds of missed opportunity

Performance synergy. A physical change may look harmful without feedback but become valuable when paired with control. Reduced passive damping, for example, can lower energy dissipation or mass while active feedback supplies damping selectively.

Feasibility. Freezing the plant can leave no controller capable of satisfying travel, force, thermal, or stability constraints. An integrated redesign may recover feasibility by changing geometry, natural frequencies, actuator authority, and feedback together.

Architecture. Sequential tuning cannot discover a missing sensor, a different actuator location, or a passive-active topology if those choices were frozen before control analysis.

Lifecycle value. A design with a higher capital cost may reduce operating energy, fatigue, maintenance, or downtime enough to improve lifecycle value. Separating hardware and control budgets can hide this system benefit.

A small numerical landscape

The following teaching model represents a plant choice p and controller choice c:

$$J(p,c) = (p-1.8)^2 + 1.4\,[c-(1.2-0.65p)]^2 + 0.15pc.$$

It is not a suspension model; it is a visualization of coupled preferences. The controller preferred at one p is not preferred at another.

```
import numpy as np

def system_objective(p, c):
    return (p - 1.8)**2 + 1.4*(c - (1.2 - 0.65*p))**2 + 0.15*p*c

p_grid = np.linspace(0.0, 3.0, 301)
c_grid = np.linspace( -1.0, 2.0, 301)
P, C = np.meshgrid(p_grid, c_grid)
J = system_objective(P, C)
i = np.unravel_index(np.argmin(J), J.shape)
print(f"Grid -search CCD optimum: p={P[i]:.3f}, c={C[i]:.3f}, J={J[i]:.3f}")
```

```
p = linspace(0,3,301);
c = linspace( -1,2,301);
[P,C] = meshgrid(p,c);
J = (P -1.8).^2 + 1.4.*(C -(1.2 -0.65.*P)).^2 + 0.15.*P.*C;
[Jmin,idx] = min(J(:));
fprintf('CCD optimum: p = %.3f, c = %.3f, J = %.3f\n', ...
        P(idx), C(idx), Jmin);
```

Numerical Diagnostic

Never claim a CCD benefit by comparing an optimized integrated design with a poorly tuned sequential baseline. Retune the controller for every plant baseline, use equivalent models and constraints, and report solver tolerances and initialization.

Activity 1.5: construct a fair baseline

Activity 1.5: construct a fair baseline

Write the minimum set of steps needed to compare sequential and CCD suspension designs fairly. Include model fidelity, disturbance, controller retuning, constraints, initialization, and validation conditions.

1.6 When CCD Is Valuable

Indicators of high value

CCD is most valuable when important system outcomes depend on interactions that no single discipline controls. Strong candidates usually exhibit several of the following characteristics.

Tight closed-loop requirements. Requirements on tracking, disturbance rejection, vibration, maneuver time, energy, or precision leave little margin. The controller operates near stability, bandwidth, force, rate, travel, power, or thermal limits.

Physical dynamics are designable. Geometry, stiffness, inertia, damping, actuator placement, sensor placement, or topology can still change. CCD cannot recover value from plant variables that contractual or certification constraints have already frozen.

Control hardware is not negligible. Actuators, sensors, processors, batteries, cooling systems, and communication networks contribute meaningful mass, cost, power, volume, or reliability effects.

Multiple operating conditions compete. A wind turbine must produce energy while limiting fatigue loads; a vehicle must balance comfort, road holding, travel, and energy; a robot must move quickly without exciting flexible modes. Feedback changes how these conflicts should be resolved physically.

Information is a design resource. Preview, estimation quality, sampling, communication, and computation materially affect performance. Different information assumptions may favor different plants.

Architecture remains open. The project can still decide which components exist and how they connect. Early CCD can prevent an architecture from becoming expensive to control—or impossible to control.

A qualitative value screen

Score each question from 0 (no) to 2 (strongly yes):

Question	0	1	2
Do plant choices substantially alter relevant dynamics?			
Does control approach a plant-dependent limit?			
Can sensing, actuation, or information architecture change?			
Are system-level objectives shared across disciplines?			
Is redesign still affordable at this project stage?			
Would a performance improvement have meaningful value?			

A high score does not prove CCD will help; it justifies a coupling study. A low score suggests using a simpler workflow unless safety or feasibility concerns dominate.

Quarter-car diagnosis

The active suspension is a strong CCD benchmark because spring and damper choices alter body and wheel modes; actuator force and travel limits become active; comfort, road holding, energy, and mass conflict; preview and estimation change achievable performance; and passive, semi-active, and active architectures are all plausible.

Design Interpretation

CCD value often comes from moving a constraint intelligently rather than lowering the unconstrained objective. An integrated design may spend a small amount of mass on actuator authority to unlock a much larger improvement in comfort or energy.

Timing matters

The earlier the architecture is open, the larger the potential benefit—but the less certain the models. Later studies have better data but less design freedom. A staged CCD process should therefore begin with low-fidelity coupling screens and progressively add fidelity as decisions become consequential.

1.7 When CCD May Not Be Worth the Additional Effort

Integration has a cost

CCD requires cross-disciplinary models, shared objectives, optimization expertise, data exchange, and organizational coordination. The integrated problem may be larger, more nonlinear, less transparent, and harder to verify than separate discipline models. Using CCD everywhere is neither efficient nor credible.

Conditions favoring a simpler process

Weak coupling. Controller retuning preserves the ranking of candidate plants, cross-sensitivities are small, and active constraints do not migrate. A modular sequential workflow may then deliver nearly the same system result.

Frozen or standardized hardware. If regulation, certification, supply chains, or platform reuse fix the plant, optimizing unavailable variables adds no value. The appropriate task is robust controller design for the fixed plant.

Mature interfaces with ample margin. Well-characterized subsystems may interact through interfaces designed with comfortable authority and bandwidth margins. Standard actuators and controllers can be cheaper and less risky than a custom integrated design.

Model uncertainty dominates the predicted gain. An optimizer may exploit details that the model cannot predict. If the expected CCD improvement is smaller than model-form error or manufacturing variability, the apparent optimum may be numerical fiction.

Implementation or certification cannot accept the result. An ideal trajectory may require unavailable preview, computation, sensing, or actuator rates. A plant optimized around that trajectory may perform poorly with the realizable controller.

The decision is low consequence. If the component is inexpensive, easily replaced, and weakly related to safety or lifecycle cost, the engineering effort may exceed the value of integration.

Implementation Warning

Do not confuse a large optimization problem with a comprehensive engineering study. An integrated formulation that omits failure modes, uncertainty, implementation delay, or hardware cost can be less trustworthy than a carefully validated sequential design.

A minimum evidence rule

Before committing to full CCD, require evidence from at least one of these tests:

- plant rankings change after controller retuning;
- a sequential baseline is infeasible but joint changes recover feasibility;
- an active constraint depends on both plant and control decisions;
- system-level sensitivity shows a meaningful interaction;
- architecture or information choices dominate performance; or
- preliminary co-design gains exceed uncertainty and implementation margins.

Staged alternatives

Declining full CCD does not mean ignoring control. Lower-cost alternatives include:

1. control-aware plant requirements;
2. parameter sweeps with controller retuning;
3. alternating design reviews;
4. response surfaces of optimized closed-loop performance;
5. robust control over a family of plant designs; and
6. integrated optimization of only the few strongly coupled variables.

Activity 1.7: argue against CCD

Activity 1.7: argue against CCD

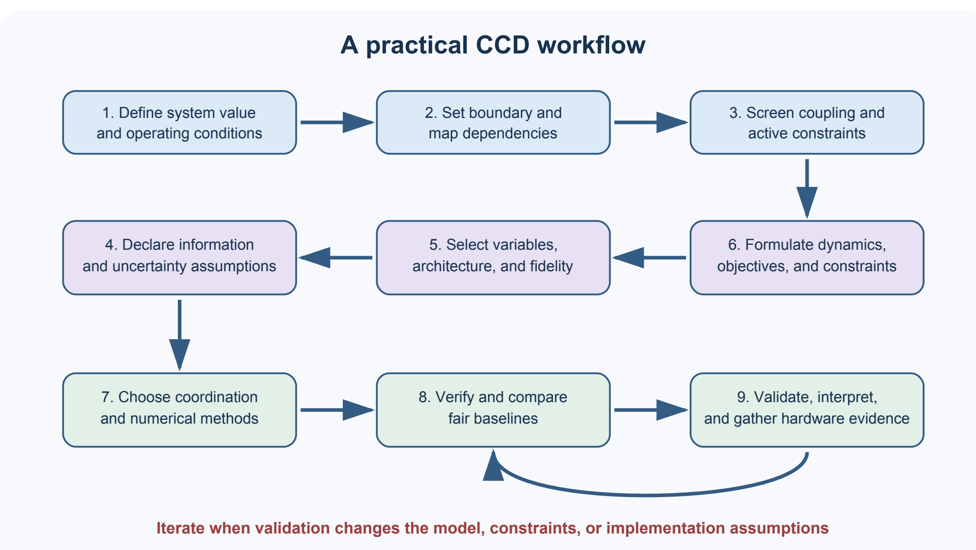
Choose one candidate application and write a one-page "no-CCD" case. State what evidence would make you reverse your recommendation. A credible CCD engineer must know when not to use CCD.

1.8 Overview of the Complete CCD Process

An end-to-end workflow

A trustworthy CCD study is a sequence of engineering arguments, not merely a solver call.

1. **Define system value.** State the mission, operating conditions, stakeholders, and system-level metrics.
2. **Choose the boundary.** Include physical, control, sensing, actuation, information, energy, and thermal effects that can change the decision.
3. **Map dependencies.** Connect every candidate variable to dynamics, objectives, constraints, and other disciplines.
4. **Screen coupling.** Retune controls across candidate plants and inspect active constraints and sensitivities.
5. **Declare information assumptions.** Separate full preview, limited prediction, state feedback, and noisy estimated feedback.
6. **Select variables and architecture.** Distinguish continuous parameters, discrete choices, controller parameters, and trajectories.
7. **Formulate dynamics and constraints.** Include saturation, rates, path constraints, uncertainty, and failure requirements.
8. **Choose coordination and numerical methods.** Sequential, nested, simultaneous, or hybrid; simulation-based or direct transcription.
9. **Verify the computation.** Check derivatives, scaling, convergence, mesh or time-step refinement, and multiple initial guesses.
10. **Compare fair baselines.** Retune every controller and apply identical requirements.
11. **Validate beyond the optimization model.** Use unoptimized conditions, higher-fidelity simulation, uncertainty, and ultimately hardware.
12. **Interpret and communicate.** Explain why the design changed and whether the gain survives implementation.



First quarter-car formulation

With body displacement z_s, wheel displacement z_u, and road input r, a simple active quarter car is

$$m_s \ddot{z}_s = -k_s(z_s - z_u) - c_s(\dot{z}_s - \dot{z}_u) + u,$$

$$m_u \ddot{z}_u = k_s(z_s - z_u) + c_s(\dot{z}_s - \dot{z}_u) - k_t(z_u - r) - u.$$

A first CCD problem can choose $p = [k_s, c_s, F_{\max}]$ and feedback gains K to minimize

$$J = \int_0^T \left(w_a \ddot{z}_s^2 + w_t (z_u - r)^2 + w_u u^2 \right) dt + w_F F_{\max},$$

subject to closed-loop stability, $|u(t)| \leq F_{\max}$, suspension travel, tire deflection, and bounds on the physical variables.

Implementation pseudocode

```
define plant variables, controller variables, and operating scenarios
for each candidate design requested by the optimizer:
    assemble the plant model from physical variables
    construct or tune the controller
    simulate every required scenario
    evaluate system objective and constraints
verify derivatives and numerical convergence
solve from multiple initial designs
retune and evaluate sequential baselines
validate selected designs using higher -fidelity and uncertain models
```

Common mistakes

Common Mistakes

- optimizing an open-loop trajectory and calling it an implementable controller;
- omitting actuator rate, thermal, energy, or bandwidth limits;
- comparing methods with different meshes or tolerances;
- allowing the CCD case more design freedom than the baseline;
- trusting one local solution without sensitivity or initialization studies; and
- reporting objective improvement without explaining the physical mechanism.

Chapter summary

Control co-design treats physical and control decisions as parts of one dynamic-system design problem. Its value comes from preserving coupled design freedom, discovering performance or feasibility synergies, and making implementation assumptions explicit. CCD includes control-inspired invention, formal co-optimization, and model/data-based co-simulation. It is most useful near coupled limits and least useful when interfaces are weak, decisions are frozen, or modeling uncertainty overwhelms the predicted benefit.

Exercises

1. Derive a state-space model for the quarter-car equations using $x = [z_s, z_u, \dot{z}_s, \dot{z}_u]^T$. Identify every matrix entry affected by k_s and c_s.
2. Give one example in which increasing actuator authority changes the optimal physical design and one in which it does not.
3. Construct a two-variable objective for a DC motor and gear train in which gear ratio and feedback gain are coupled. State realistic constraints.
4. Compare control-inspired design, co-optimization, and co-simulation for a flexible robotic arm. What would each perspective contribute?
5. Design a fair numerical experiment that tests whether a sequential suspension design is within 2% of the integrated optimum.
6. Explain why an OLOC-optimal plant may not be optimal under noisy, delayed feedback.
7. Apply the qualitative value screen from Section 1.6 to a wind turbine, microgrid, or assistive robot.
8. Identify three pieces of evidence required before an organization should adopt the CCD result in hardware.

Open research question

Can a computationally inexpensive, model-agnostic diagnostic reliably predict the value of CCD before a full co-design optimization is solved?

Sources and further reading

- M. Garcia-Sanz, "Control Co-Design: An Engineering Game Changer," *Advanced Control for Applications*, vol. 1, e18, 2019.
- J. T. Allison and D. R. Herber, "Multidisciplinary Design Optimization of Dynamic Engineering Systems," *AIAA Journal*, vol. 52, no. 3, pp. 691–710, 2014.
- D. R. Herber and J. T. Allison, "Nested and Simultaneous Solution Strategies for General Combined Plant and Control Design Problems," *Journal of Mechanical Design*, 2019.

Dynamic Systems and Feedback for Co-Design

2.1 States, Inputs, Outputs, and Disturbances

Assign each quantity one role

A dynamic model is useful for CCD only when every quantity has a clear engineering role. Consider

$$\dot{x} = f(t, x, u, d, p), \qquad y = h(t, x, u, d, p),$$

where $x(t)$ is the state, $u(t)$ is the manipulated input, $d(t)$ is an environmental or exogenous disturbance, $y(t)$ is an output, and p is a vector of time-independent physical design variables.

The **state** is the minimum information needed, together with future inputs and disturbances, to predict future behavior. States usually represent stored energy or memory: position and momentum, capacitor charge, temperature, fluid inventory, or estimator memory. An **input** is a command the controller can choose. A **disturbance** influences the system but is not chosen by the controller. An **output** is a quantity calculated or measured for feedback, constraints, or performance evaluation.

Mass-spring-damper example

For a mass m, spring k, damper c, actuator force u, and base displacement r, let q be mass displacement. One model is

$$m\ddot{q} + c(\dot{q} - \dot{r}) + k(q - r) = u.$$

With $x = [q, \dot{q}]^T$ and disturbance $d = [r, \dot{r}]^T$,

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} x + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u + \begin{bmatrix} 0 & 0 \\ k/m & c/m \end{bmatrix} d.$$

If the design variables are $p = [m, k, c, F_{\max}]^T$, then the matrices and the constraint $|u| \le F_{\max}$ both depend on physical design.

Measured outputs are not performance outputs

A displacement sensor may provide $y_m = q - r$, while the objective penalizes acceleration $y_p = \ddot{q}$. The first is available to the controller; the second may be computed only for evaluation. Confusing them silently grants the controller information it does not possess.

Similarly, a road profile may be a disturbance. If a preview sensor measures it ahead of the vehicle, some portion becomes an available feedforward signal. Information availability changes the controller class and can change the optimal plant.

Implementation Warning

Do not place a disturbance, unmeasured state, or future trajectory inside the feedback law unless a sensor, estimator, preview model, or communication channel provides it.

State choice and design dependence

Different state coordinates describe the same dynamics, but some expose design dependence more clearly. Modal coordinates are useful for flexible structures, energy variables for multiphysics systems, and relative coordinates for suspensions. A model developed around fixed numerical parameters may be excellent for control tuning yet unusable for CCD if it cannot be reassembled when geometry or material variables change.

Activity 2.1: classify the variables

Activity 2.1: classify the variables

For an electric vehicle battery thermal system, classify cell temperature, coolant flow command, ambient temperature, vehicle speed, pump size, measured outlet temperature, and desired temperature as states, inputs, disturbances, outputs, or design variables. Explain any quantity that could occupy more than one role.

2.2 Continuous- and Discrete-Time Dynamic Models

Two time descriptions

Physical laws are often expressed continuously,

$$\dot{x}(t) = A(p)x(t) + B(p)u(t) + E(p)d(t),$$

while digital controllers update at sample times $t_k = kT_s$,

$$x_{k+1} = A_d(p, T_s)x_k + B_d(p, T_s)u_k + E_d(p, T_s)d_k.$$

Under a zero-order hold,

$$A_d = e^{AT_s}, \qquad B_d = \int_0^{T_s} e^{A\tau} B \, d\tau.$$

Discretization is therefore not a clerical conversion. Sampling time interacts with plant eigenvalues, actuator bandwidth, delay, estimator quality, computational cost, and the number of decision variables in an optimal-control transcription.

A CCD interpretation

A lighter or more flexible plant may introduce a faster mode. The old sample time can then become inadequate. A faster controller may require a more expensive processor or sensor, increasing power and cost. If sample time is fixed while the plant changes, the optimizer may select a design that is acceptable in the continuous model but unstable or poorly damped when implemented.

For a rough rule, the sample frequency should exceed the important closed-loop bandwidth by a comfortable factor. The correct factor depends on delay, filtering, phase margin, and implementation method; it must be verified rather than assumed.

Exact discretization in Python

```
import numpy as np
from scipy.signal import cont2discrete

m, c, k = 2.0, 8.0, 120.0
A = np.array([[0.0, 1.0], [ -k/m, -c/m]])
B = np.array([[0.0], [1.0/m]])
C = np.eye(2)
D = np.zeros((2, 1))

for Ts in (0.001, 0.01, 0.05):
    Ad, Bd, _, _, _ = cont2discrete((A, B, C, D), Ts, method="zoh")
    print(f"Ts={Ts:0.3f} s, discrete poles={np.linalg.eigvals(Ad)}")

m = 2; c = 8; k = 120;
A = [0 1; -k/m -c/m];
B = [0; 1/m];
sysc = ss(A,B,eye(2),zeros(2,1));
for Ts = [0.001 0.01 0.05]
```

```
    sysd = c2d(sysc,Ts,'zoh');
    fprintf('Ts = %.3f s, max pole magnitude = %.5f\n', ...
            Ts,max(abs(eig(sysd.A))));
end
```

Time-step convergence

Simulation and optimization time steps serve different purposes. A simulation step must resolve dynamic behavior accurately. A controller sample time defines information and command updates. A transcription mesh also controls optimization accuracy. These can differ, but each needs a convergence study.

Numerical Diagnostic

Repeat the CCD optimization with smaller simulation steps or a refined transcription mesh. If the plant or controller changes materially, the original discretization was part of the answer rather than a faithful numerical approximation.

Activity 2.2: sampling as a design constraint

Activity 2.2: sampling as a design constraint

Suppose a beam redesign raises its first flexible-mode frequency from 8 Hz to 28 Hz, while the controller samples at 100 Hz with 4 ms total delay. Explain why reusing the original digital controller model may be unsafe.

2.3 Linear and Nonlinear Systems

Linear models expose structure

A linear time-invariant model has the form

$$\dot{x} = A(p)x + B(p)u + E(p)d, \qquad y = C(p)x + D(p)u.$$

Linear models support poles, zeros, transfer functions, controllability, observability, frequency response, and convex controller formulations. They are often the best first model for coupling diagnosis and local controller design.

But most engineered systems are nonlinear:

$$\dot{x} = f(x, u, d, p).$$

Geometry, aerodynamic forces, friction, contact, actuator saturation, fluid flow, and power electronics introduce nonlinearities. Linearization about (x_0, u_0, p_0) gives

$$\delta\dot{x} = A\delta x + B\delta u, \quad A = \left.\frac{\partial f}{\partial x}\right|_0, \quad B = \left.\frac{\partial f}{\partial u}\right|_0.$$

Adjustable inverted pendulum

For pendulum angle θ from upright, length l, mass m, damping b, and pivot torque u,

$$ml^2\ddot{\theta} + b\dot{\theta} - mgl\sin\theta = u.$$

Near upright, $\sin\theta \approx \theta$, so

$$\ddot{\theta} \approx \frac{g}{l}\theta - \frac{b}{ml^2}\dot{\theta} + \frac{1}{ml^2}u.$$

Increasing l reduces the open-loop divergence rate $\sqrt{g/l}$ but also reduces torque authority through $1/(ml^2)$. Geometry changes instability and actuation simultaneously. A linear CCD study can reveal that trade locally, but it cannot certify recovery from large initial angles or account for torque saturation during swing-up.

When a linear model is adequate

A linear model is often sufficient when operating deviations are small, one equilibrium dominates, nonlinear constraints remain inactive, and the final design is validated over the required envelope. A nonlinear model is needed when the design exploits large motion, saturation, switching, contact, state-dependent aerodynamics, or multiple equilibria.

CCD Insight

The optimizer may push the system toward regions where the approximation is weakest. Model validity must be treated as a design-dependent constraint, not checked only at the initial plant.

Multi-point and parameter-varying models

Between a single linearization and a full nonlinear model lie useful compromises: linearizations at several operating points, gain-scheduled models, linear parameter-varying models, and reduced nonlinear surrogates. A wind turbine, for example, may require different local models across below-rated and above-rated operation. The CCD problem must prevent improvement at one point from degrading another unacceptably.

Activity 2.3: locate the nonlinearity

Activity 2.3: locate the nonlinearity

For a robotic arm, list nonlinearities caused by geometry, friction, actuator limits, and contact. Identify which could change the preferred link length or motor size.

2.4 Feedback and Feedforward Control

Feedback reacts to measured consequences

Feedback computes a command from measured or estimated behavior,

$$u(t) = \kappa(r_c(t), \hat{x}(t), y(t), c)\,,$$

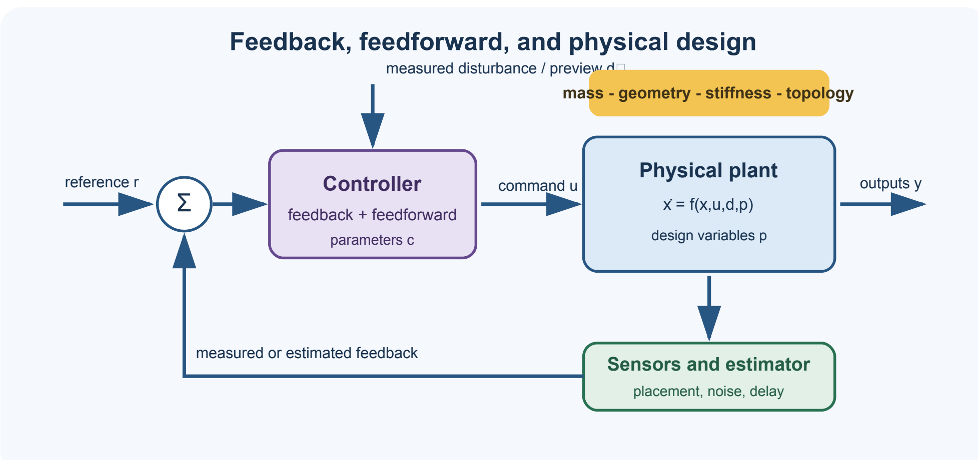
where r_c is a reference and c contains controller parameters. Feedback can reject unmeasured disturbances, stabilize unstable plants, reduce sensitivity to parameter error, and enforce desired dynamics. It acts only after information about the disturbance or resulting error becomes available.

Feedforward anticipates known causes

Feedforward uses a reference or measured disturbance directly,

$$u(t) = u_{\mathrm{ff}}(r_c, d_m, p) + u_{\mathrm{fb}}(\hat{x}, r_c, c)\,.$$

It can cancel predictable effects without waiting for error, but its quality depends on model accuracy and information. Road preview, wind preview, commanded robot trajectories, and known payload changes can support feedforward action.



Why the distinction matters in CCD

An open-loop optimal trajectory often assumes the entire disturbance is known. A realizable feedback controller may know only current measurements. A preview controller lies between these cases. Because information changes achievable performance, each assumption can favor a different plant.

For the suspension, road preview allows the actuator to prepare for an upcoming bump. Without preview, the controller reacts after body or wheel motion begins. More preview may justify softer passive elements or a smaller actuator; limited sensing may favor a more forgiving passive design.

Two-degree-of-freedom control

Separating feedforward and feedback clarifies design responsibilities. Feedforward shapes nominal reference tracking; feedback supplies robustness and disturbance rejection. CCD should evaluate both nominal performance and sensitivity to modeling error. An architecture with excellent feedforward tracking but weak feedback robustness may be undesirable.

From OLOC to Real-Time Control

Treat open-loop optimal control as a performance discovery tool unless the assumed future information truly exists. Then approximate its useful structure with causal feedforward, preview control, feedback, or model predictive control and re-optimize the plant if necessary.

Controller structure is a design choice

PID, state feedback, output feedback, gain scheduling, robust control, and MPC expose different parameters and information requirements. A fair CCD comparison must hold physical design freedom and requirements constant while acknowledging that controller classes have different implementation costs.

Activity 2.4: information audit

Activity 2.4: information audit

For an autonomous vehicle following a planned path in unknown crosswind, classify the path, measured wind, future wind, lateral error, and estimated tire force as feedback, feedforward, preview, or unavailable information.

2.5 Stability and Transient Response

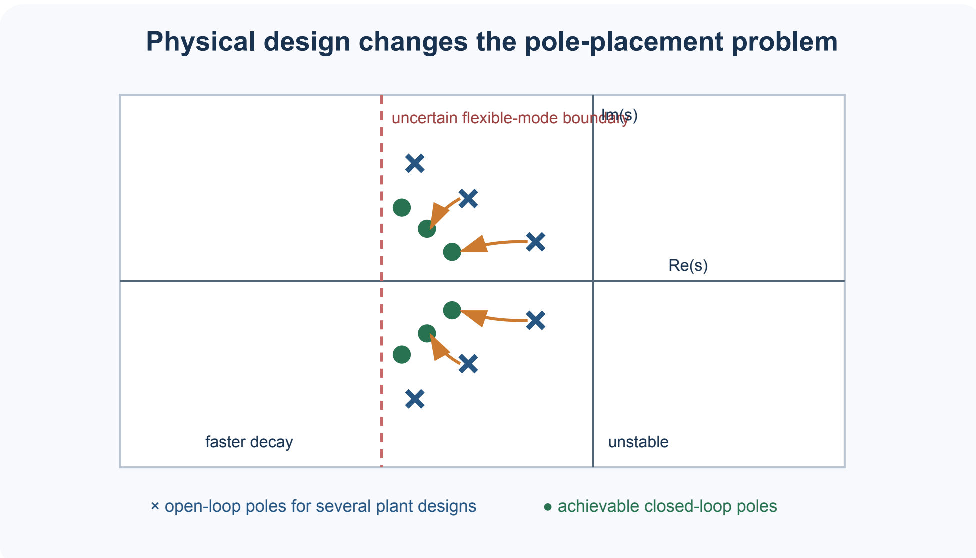
Stability is a feasibility condition

For the linear closed-loop system

$$\dot{x} = A_{\mathrm{cl}}(p, c)x,$$

asymptotic stability requires every continuous-time eigenvalue of A_{cl} to have negative real part. In discrete time, every eigenvalue must lie inside the unit circle. Stability should normally be enforced as a constraint, not traded against cost as if mild instability were acceptable.

Physical design changes open-loop poles. Stiffness and inertia change natural frequencies; damping changes decay; aerodynamics can add negative damping; flexible modes add lightly damped poles. The controller moves closed-loop poles only within limits imposed by authority, sensing, delay, uncertainty, and unmodeled dynamics.



Second-order interpretation

For

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0,$$

ω_n sets a response time scale and ζ controls oscillation and decay. For the passive mass-spring-damper,

$$\omega_n = \sqrt{k/m}, \qquad \zeta = \frac{c}{2\sqrt{km}}.$$

Changing mass or stiffness therefore changes the bandwidth a controller must address. Raising stiffness may speed the rigid-body mode while introducing or exposing structural modes that limit gain.

Transient metrics

Rise time, settling time, overshoot, peak acceleration, vibration decay, and disturbance amplification describe different aspects of response. CCD should use metrics tied to the mission. A pole-placement target alone does not enforce actuator effort, comfort, stress, or saturation.

Pole calculation

```
import numpy as np

def passive_poles(m, c, k):
    A = np.array([[0.0, 1.0], [ -k/m, -c/m]])
    return np.linalg.eigvals(A)

for k in (50.0, 150.0, 450.0):
    poles = passive_poles(m=2.0, c=8.0, k=k)
    print(f"k={k:5.1f} N/m -> poles {poles}")
```

Robust stability

Nominal poles are not enough. Plant uncertainty, delay, neglected modes, and controller implementation can erode margins. A physical design that permits high nominal feedback gain may amplify an uncertain flexible mode. Gain margin, phase margin, structured uncertainty, and worst-case analysis become design criteria when robustness drives feasibility.

Implementation Warning

Do not reward arbitrarily fast nominal poles without actuator, sensor, delay, noise, and uncertainty models. Unlimited bandwidth is a modeling error, not a control achievement.

Activity 2.5: interpret a redesign

Activity 2.5: interpret a redesign

A structural redesign doubles a dominant natural frequency but reduces modal damping by half. List the possible benefits and risks for feedback design, sensing, actuator bandwidth, and uncertainty.

2.6 Controllability and Control Authority

Can the input move every relevant state?

For $\dot{x} = Ax + Bu$, the controllability matrix is

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix}.$$

The system is controllable if $\mathrm{rank}(\mathcal{C}) = n$. Rank is a yes-or-no structural test. CCD also needs a quantitative question: how much effort is required to move each state or mode?

The finite-horizon controllability Gramian

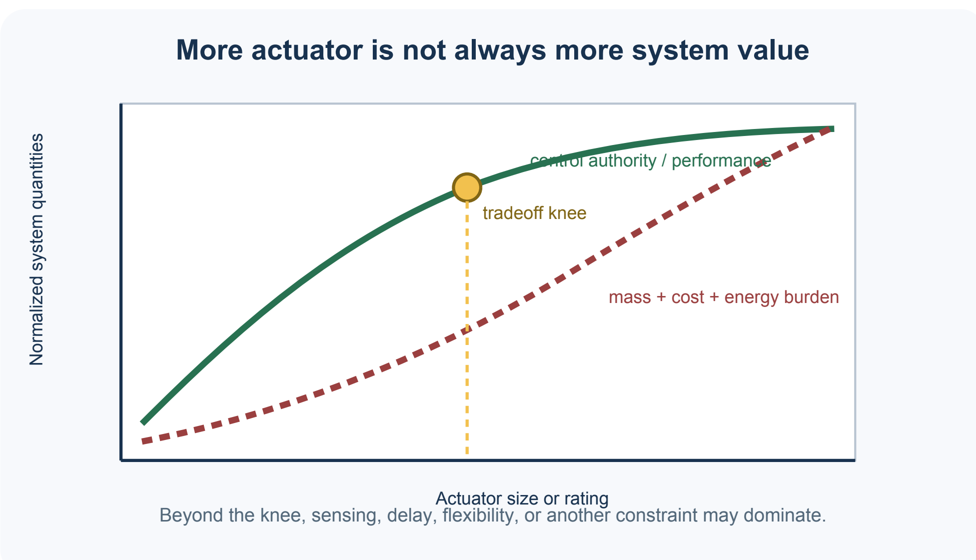
$$W_c(T) = \int_0^T e^{At} B B^T e^{A^T t}\, dt$$

describes reachable directions. Small eigenvalues indicate directions that require large energy. A design can be mathematically controllable but practically uncontrollable because force, stroke, rate, power, or bandwidth is insufficient.

Actuator placement on a flexible beam

A point actuator couples strongly to modes whose shape is large at the actuator location and weakly to modes near a node. Moving the actuator changes the modal input matrix $B(p)$. Placement should therefore consider the modes that constrain performance, not only geometric convenience.

Size, authority, and diminishing returns



A larger actuator may improve the smallest Gramian eigenvalue and reduce saturation, but it adds mass, volume, heat, cost, and energy demand. Beyond a point, performance becomes limited by sensing, delay, structural flexibility, or another constraint. CCD seeks the knee of this system-level trade, not maximum authority in isolation.

A placement calculation

```
import numpy as np
from scipy.linalg import solve_continuous_lyapunov
```

```
def gramian_metrics(A, B):
    # For stable A, A W + W A^T + B B^T = 0.
    W = solve_continuous_lyapunov(A, -(B @ B.T))
    eigs = np.linalg.eigvalsh(W)
    return eigs.min(), np.trace(W)

A = np.array([[0, 1, 0, 0], [ -4, -0.3, 1, 0],
              [0, 0, 0, 1], [1, 0, -25, -0.8]], dtype=float)
for label, B in {
    "near mode -1 antinode": np.array([[0], [1.0], [0], [0.15]]),
    "balanced location":   np.array([[0], [0.65], [0], [0.65]]),
}.items():
    print(label, gramian_metrics(A, B))
```

Physical Interpretation

Controllability informs whether an actuator location can influence a mode. Control authority determines whether it can do so within real force, energy, and time limits.

Activity 2.6: choose an actuator location

Activity 2.6: choose an actuator location

Two beam locations have equal static deflection per unit force. One lies near a node of the second mode; the other couples moderately to the first three modes. Which is preferable for vibration suppression, and what additional information is required?

2.7 Observability and State Estimation

Can measurements reveal the state?

For

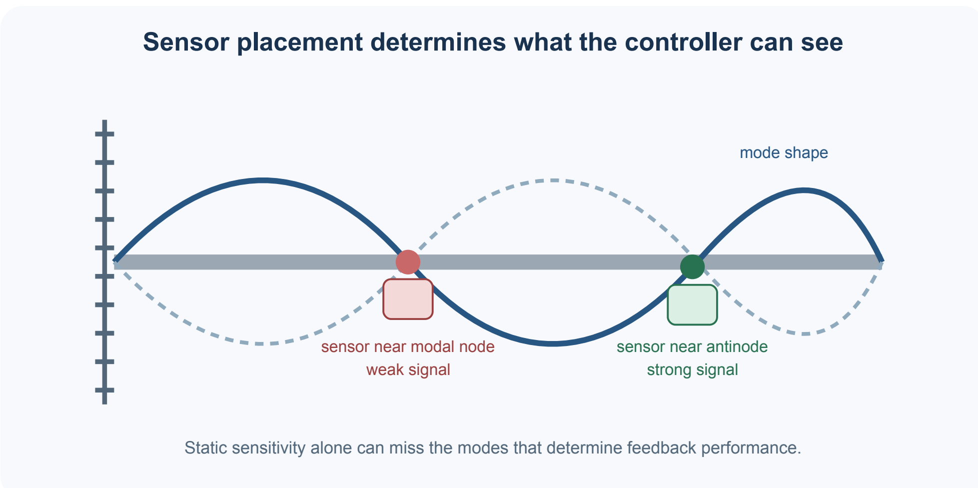
$$\dot{x} = Ax + Bu, \qquad y = Cx,$$

the observability matrix is

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}.$$

The system is observable if $\operatorname{rank}(\mathcal{O}) = n$. As with controllability, binary rank is only the beginning. Poorly observable directions amplify noise and model error in an estimator.

Sensor placement is a physical design decision



A strain gauge near a modal node provides little information about that mode. A body-mounted accelerometer may reveal vibration but provide weak low-frequency position information. Camera placement affects occlusion and geometry; thermocouple placement affects delay; pressure sensors alter plumbing and packaging.

CCD may optimize sensor type, location, precision, sampling, and redundancy along with the plant and controller. The measurement matrix $C(p, s)$ and noise covariance then depend on design choices.

State estimation

An observer combines the model, command, and measurement:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}).$$

The estimator error follows

$$\dot{e} = (A - LC)e$$

in the nominal deterministic case. A Kalman filter selects L from process- and measurement-noise assumptions. Estimator bandwidth must balance tracking against noise amplification.

Separation is useful but not absolute in CCD

For a fixed linear plant under standard assumptions, controller and observer poles may be designed separately. In CCD, plant changes alter A, B, C, noise transmission, and model uncertainty. A sensor layout that works for one structure may become poor after geometry changes. Estimator and controller design therefore remain coupled through the physical design even when the separation principle applies locally.

Observability check

```
import numpy as np

def observability_matrix(A, C):
    return np.vstack([C @ np.linalg.matrix_power(A, i) for i in range(A.shape[0])])

A = np.array([[0, 1, 0, 0], [ -4, -0.3, 1, 0],
              [0, 0, 0, 1], [1, 0, -25, -0.8]], dtype=float)
for label, C in {
    "sensor near modal node": np.array([[1.0, 0, 0.02, 0]]),
    "two separated sensors": np.array([[1.0, 0, 0, 0], [0, 0, 1.0, 0]]),
}.items():
    O = observability_matrix(A, C)
    print(label, "rank=", np.linalg.matrix_rank(O), "condition=", np.linalg.cond(O))
```

Numerical Diagnostic

Full observability rank does not guarantee a useful estimator. Inspect conditioning, observability Gramians, expected noise, uncertainty, and estimator performance over all candidate plant designs.

Activity 2.7: measurement versus performance

Activity 2.7: measurement versus performance

For an active suspension, distinguish sensors needed for body acceleration, suspension travel, wheel motion, and road preview. Which outputs are measured directly, estimated, or used only for performance evaluation?

2.8 Actuator Bandwidth, Saturation, and Rate Limits

The command is not the delivered input

An ideal model applies $u_c(t)$ directly to the plant. A real actuator has internal dynamics and limits. A first-order model is

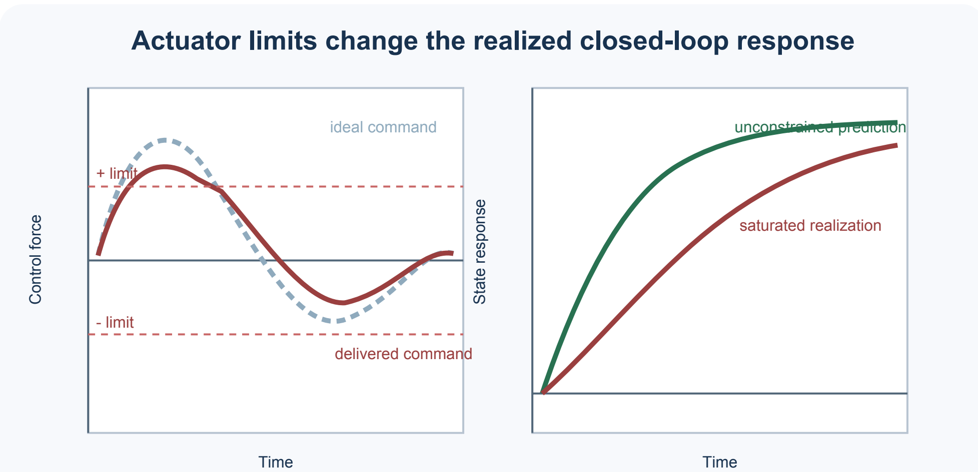
$$\tau_a \dot{u} + u = u_c,$$

with constraints

$$u_{\min} \le u \le u_{\max}, \qquad \dot{u}_{\min} \le \dot{u} \le \dot{u}_{\max}.$$

Electrical current, voltage, hydraulic flow, motor speed, temperature, duty cycle, and stored energy may impose additional constraints. The actuator rating is often a plant design variable; the achievable control trajectory depends on it.

Saturation changes dynamics and design



Ignoring limits can favor a plant that depends on physically unavailable control effort.

When the controller saturates, the closed loop is nonlinear and the assumed pole locations no longer describe the response. Integral action may wind up. An optimizer that ignores saturation can reduce passive capability and rely on control effort that hardware cannot deliver.

Including only a large penalty on u^2 is not equivalent to enforcing a hard force limit. A penalty discourages effort when convenient; a constraint prohibits an impossible command.

Bandwidth and structural flexibility

Actuator bandwidth must cover useful control frequencies, but extending bandwidth into uncertain flexible modes can reduce robustness. Collocated actuator-sensor pairs often have favorable passivity properties, while noncollocated arrangements can introduce difficult zeros and phase loss.

For a beam, a high-force slow actuator may control rigid motion but not vibration. A smaller piezoelectric actuator may address high-frequency modes but provide little stroke. Architecture and placement may require multiple actuator technologies.

Anti-windup and constraint-aware control

Anti-windup logic, reference governors, MPC, and command shaping can manage limits, but they cannot create unavailable authority. CCD should include the intended limit-handling strategy during evaluation because different strategies can change the preferred passive design and actuator rating.

Implementation Warning

Report peak force, RMS force, rate, power, energy, and thermal duty separately. Two actuators with the same peak force can have very different feasible trajectories.

Activity 2.8: redesign under saturation

Activity 2.8: redesign under saturation

An active suspension meets comfort targets with unlimited force but saturates over severe bumps. Propose three physical changes and three control changes. Explain which changes trade passive performance against actuator demand.

2.9 Sensor Noise, Delay, and Sampling

Measurement quality is part of the architecture

A sampled measurement can be written

$$y_k = Cx(t_k - \tau_d) + v_k,$$

where v_k is measurement noise and τ_d includes sensing, communication, filtering, computation, and actuation delay. Quantization, dropouts, bias, drift, asynchronous clocks, and calibration error may also matter.

Noise limits useful bandwidth

High-gain feedback can reject disturbances but also amplify measurement noise into actuator activity, structural vibration, heat, and wear. Filtering reduces noise but adds phase lag and delay. A physical design that requires aggressive feedback near the noise floor may be inferior to one with slightly better passive behavior.

Sensor precision therefore trades against actuator demand and plant design. A more accurate sensor may enable a lighter structure or smaller actuator. Conversely, improving physical damping may permit cheaper sensing.

Delay consumes phase margin

A pure delay contributes phase $-\omega\tau_d$. As bandwidth increases, the same delay becomes more damaging. Delays can make an attractive centralized architecture unusable and favor local sensing and control, faster communication, lower bandwidth, or a different plant.

For a flexible structure, moving a mode to higher frequency may appear beneficial, but if the controller bandwidth follows while delay is fixed, phase margin can shrink. CCD must evaluate the complete sampled-data loop.

Aliasing and sampling

Sampling below twice a signal frequency causes aliasing, but merely satisfying the Nyquist inequality is not enough for control. Anti-alias filters, transient response, delay, and closed-loop bandwidth require faster sampling. Flexible modes above the intended bandwidth can still alias into measurements and destabilize feedback if filtering is inadequate.

Architecture consequences

Sensor choice affects more than C:

- location changes modal visibility;
- technology changes bandwidth and noise;
- filtering changes delay;
- communication changes latency and dropout risk;
- redundancy changes reliability, mass, and cost; and
- sampling changes computation and energy.

CCD Insight

Information quality has physical value. Treat sensor accuracy, delay, and preview as design resources whose cost and system benefit can be traded explicitly.

Activity 2.9: delay budget

Activity 2.9: delay budget

Construct a delay budget for a vision-based inverted pendulum controller. Include exposure, image transfer, processing, state estimation, control computation, communication, and actuator response. Which terms can physical or architectural redesign reduce?

2.10 Hybrid and Switched Systems

Dynamics plus logic

A hybrid system combines continuous state evolution with discrete modes. In mode q,

$$\dot{x} = f_q(x, u, d, p),$$

and guards trigger transitions $q^+ = \Delta(q, x, u, d, p)$. State resets may occur at impact, connection, or switching events.

Examples include contact and flight in legged robots, gear shifts in vehicles, below-rated and above-rated wind-turbine operation, valve logic in fluid systems, charging and discharging modes in batteries, and passive/semi-active/active suspension modes.

Why hybrid behavior matters for CCD

Physical design changes when transitions occur. Pendulum length affects impact speed; battery size affects when power limits activate; gearing affects shift timing; platform motion affects turbine shutdown events. If an optimizer simulates only one smooth mode, it may exploit a model that never represents the limiting event.

Switching also creates nonsmooth objectives and constraints. Small design changes can add or remove an event, making finite-difference derivatives unreliable. Event detection, consistent reset maps, smoothing, mixed-integer formulations, complementarity methods, or derivative-free searches may be required.

Avoid hidden simulation logic

Common hidden logic includes clipping, lookup-table discontinuities, conditional controller gains, solver event resets, and safety shutdowns. These operations must be documented because they shape the design landscape.

Numerical Diagnostic

Record mode sequences and event times for candidate designs. If two nearby designs follow different sequences, investigate derivative validity and whether the optimization formulation represents switching explicitly.

Chapter summary

Dynamic-system design begins with clear roles for states, inputs, outputs, disturbances, information, and physical variables. Continuous models must be reconciled with sampled implementation. Linear models reveal local structure, while nonlinear and hybrid models protect against invalid extrapolation. Feedback, feedforward, stability, controllability, and observability connect plant choices to achievable closed-loop behavior. Actuator and sensor limitations are not implementation details; they can change the system-optimal physical design.

Exercises

1. Derive continuous- and discrete-time state-space models for a base-excited mass-spring-damper. Retain m, k, and c symbolically.
2. Linearize the adjustable inverted pendulum about upright. Explain how m and l affect instability and torque authority.
3. For a two-mode beam model, compare actuator locations using controllability rank and a Gramian metric.
4. Propose two sensor layouts for the beam and compare observability, noise, and implementation cost.
5. Show how a 10 ms delay changes phase at 5, 20, and 50 Hz. Discuss the resulting architecture implications.

6. Simulate a saturated PI-controlled mass-spring-damper with and without anti-windup.
7. Explain why a controller designed on a continuous model may fail after sampling and zero-order hold.
8. Formulate a hybrid model for a semi-active suspension whose damper switches between two settings.
9. Create a table that separates measured outputs, estimated states, performance outputs, and constrained outputs for the quarter car.
10. Identify one plant change that improves controllability but worsens observability, cost, or robustness.

Open research question

How can actuator placement, sensor placement, control architecture, and physical geometry be optimized together while preserving robust closed-loop guarantees for nonlinear and hybrid systems?

Sources and further reading

- G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*.
- K. Ogata, *Modern Control Engineering*.
- J. T. Allison and D. R. Herber, "Multidisciplinary Design Optimization of Dynamic Engineering Systems," *AIAA Journal*, 2014.
- M. Garcia-Sanz, "Control Co-Design: An Engineering Game Changer," *Advanced Control for Applications*, 2019.

Mathematical Foundations

Appendix 1: Summation Notation and the Mean

This course will rely on your understanding of some ideas from calculus and high school algebra. The appendix of the course notes serves to review these ideas.

Sums and averages play an important role in machine learning. Here, we'll review the most relevant properties of summation notation, and use the arithmetic mean as a case study of sorts.

Introduction

Definition: Summation Notation

The $\sum$ symbol, read "sigma", is used to indicate a sum of a sequence. In general, if a and b are integers, and $x_1, x_2, \ldots$ is a collection of numbers, then:

$$\sum_{i=a}^{b} x_i = x_a + x_{a+1} + x_{a+2} + \quad + x_{b-1} + x_b$$

Above, i is the *index of summation*.

For example, if we take $x_i = i^2$, then $\sum_{i=1}^{6} i^2$ represents the sum of the squares of all integers from 1 to 6:

$$\sum_{i=1}^{6} i^2 = 1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2$$

Notice that both the starting and ending indices (1 and 6, respectively) are included in the sum. Summation notation allows us to express sums conveniently – the left-hand side above is more compact than the right-hand side.

Often, we'll take the sum of the first n terms of a sequence. For example, the sum of the squares of the first n positive integers is:

$$\sum_{i=1}^{n} i^2$$

Note that the index of summation can be any variable name (i is just a typical choice). That is, $\sum_{j=1}^{n} j^2$, $\sum_{i=1}^{n} i^2$, and $\sum_{\text{zebra}=1}^{n} \text{zebra}^2$ all represent the same sum.

Summation notation can be thought of in terms of a `for`-loop. In Python, to compute the sum $\sum_{i=a}^{b} i^2$, we could write:

```
total = 0
for i in range(a, b + 1):
total = total + i ** 2
```

As we mentioned above, the ending index is *inclusive* in summation notation. This is in contrast to Python, where the ending index is *exclusive*, which is why we provided `b + 1` as the second argument to the `range` function instead of `b`.

Activity 1

Activity 1

Identify what makes each of the following expressions invalid or ambiguous, and why.

- $\sum_{k=1}^{13} k^2 + k$
- $i \sum_{i=1}^{5} i$
- $\sum_{i=1}^{n} x_i + x_i$
- $\sum_{i=5}^{3} i^2$
- $\sum_{i=2}^{\pi} x_i$

Properties and Examples

To illustrate various properties of summation notation, we'll use the following fact:

$$\sum_{i=0}^{n} 2^i = 2^0 + 2^1 + 2^2 + \ldots + 2^n = 2^{n+1} - 1$$

The expression $2^{n+1} - 1$ is called a **closed form** for the summation. When closed forms exist, they make it easy to compute the value of a summation. You'll also notice that in addition to writing the sum using summation notation, I also showed the first few and last terms of the sum, with a ... to indicate that the pattern continues in between. As convenient as summation notation is, explicitly writing the first few and last terms of a sum can sometimes make it easier to understand what exactly is being summed.

For example, $\sum_{i=0}^{3} 2^i = 2^0 + 2^1 + 2^2 + 2^3 = 1 + 2 + 4 + 8 = 15$, which is indeed $2^{3+1} - 1 = 15$. The sequence $2^0, 2^1, 2^2, \ldots, 2^n$ is called a geometric sequence, and the resulting sum is called a geometric series.

For convenience, we'll define $S(n) = \sum_{i=0}^{n} 2^i$; again, $S(n) = 2^{n+1} - 1$.

The best way to learn through these examples is to try to solve them yourself before looking at the solution.

Example: Partial Sums. Determine the value of $\sum_{i=8}^{19} 2^i$. (Find an answer that doesn't involve summation notation or a sum over many terms.)

Solution

We can't use the formula for $S(n)$ directly, because the starting index is 8, not 0. But, if we look at the expansion of $S(19)$, we'll see that it contains the sum we're looking for:

$$\begin{aligned} S(19) &= \sum_{i=0}^{19} 2^i \\ &= 2^0 + 2^1 + 2^2 + \cdots + 2^{19} \\ &= (2^0 + 2^1 + 2^2 + \cdots + 2^7) + (2^8 + 2^9 + 2^{10} + \cdots + 2^{19}) \\ &= \underbrace{\sum_{i=0}^{7} 2^i}_{S(7)} \quad + \quad \underbrace{\sum_{i=8}^{19} 2^i}_{\text{what we're looking for}} \end{aligned}$$

So, the piece we're looking for is $S(19) - S(7)$, or:

$$\sum_{i=8}^{19} 2^i = S(19) - S(7) = (2^{20} - 1) - (2^8 - 1) = 2^{20} - 2^8$$

The specific value of the sum, $2^{20} - 2^8$, is $1048576 - 256 = 1048320$, if you're curious.

Example: Constant Multiples. Determine the value of $\sum_{i=0}^{7} 5 \cdot 2^i$.

Solution

$$\begin{aligned} \sum_{i=0}^{7} 5 \cdot 2^i &= 5 \cdot 2^0 + 5 \cdot 2^1 + 5 \cdot 2^2 + 5 \cdot 2^3 + 5 \cdot 2^4 + 5 \cdot 2^5 + 5 \cdot 2^6 + 5 \cdot 2^7 \\ &= 5(2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 + 2^6 + 2^7) \\ &= 5 \sum_{i=0}^{7} 2^i \\ &= 5(2^8 - 1) \end{aligned}$$

This last line is what's most important; its specific value is $5 \cdot (2^8 - 1) = 5 \cdot 255 = 1275$.

Example: Sum of a Constant. Determine the value of $\sum_{i=7}^{20} 5$.

Solution

$$
\begin{aligned}
\sum_{i=7}^{20} 5 &= \underbrace{5+5+5+\cdots+5}_{14\text{ times}} \\
&= 5 \cdot 14 \\
&= 70
\end{aligned}
$$

Where did 14 come from? The starting index is 7, and the ending index is 20. So, the number of terms is $20 - 7 + 1 = 14$.

Example: Shifting Indices. Determine the value of $\sum_{i=10}^{20} 2^{i-5}$.

Solution

As we did in the very first example, let's try and write the sum as a difference of two calls to $S(n)$. $S(n)$ involves the sum of terms of the form 2^i, and the sum we're looking for involves the terms 2^{i-5}, so we'll need to shift the indices.

When $i = 10$, $i - 5 = 5$, and when $i = 20$, $i - 5 = 15$. So, we can rewrite the sum in question as:

$$\sum_{i=10}^{20} 2^{i-5} = \sum_{i=5}^{15} 2^i$$

This looks like $S(15) - S(4)$, or $2^{16} - 1 - (2^5 - 1) = 2^{16} - 2^5 = 65536 - 32 = 65504$.

Example: Separating Sums. Given that $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$, determine the value of $\sum_{i=0}^{12}(2^i + 5i)$.

Solution

$$
\begin{aligned}
\sum_{i=0}^{12}(2^i + 5i) &= (2^0 + 5\cdot 0) + (2^1 + 5\cdot 1) + (2^2 + 5\cdot 2) + \ldots + (2^{12} + 5\cdot 12) \\
&= (2^0 + 2^1 + 2^2 + \ldots + 2^{12}) + (5\cdot 0 + 5\cdot 1 + 5\cdot 2 + \ldots + 5\cdot 12) \\
&= \sum_{i=0}^{12} 2^i + \sum_{i=0}^{12} 5i \\
&= 2^{13} - 1 + 5 \cdot \frac{12\cdot 13}{2} \\
&= 8581
\end{aligned}
$$

The key here is that we can split the sum into two sums.

Key Takeaways. In the order they were introduced in the examples, here are some useful properties of summations:

1. Partial Sums:

$$\sum_{i=1}^{n} x_i = \sum_{i=1}^{k} x_i + \sum_{i=k+1}^{n} x_i$$

2. Constant Multiples:

$$\sum_{i=1}^{n} cx_i = c\sum_{i=1}^{n} x_i$$

3. Sum of a Constant:

$$\sum_{i=1}^{n} c = c \cdot n$$

4. Shifting Indices:

$$\sum_{i=k}^{n} x_i = \sum_{i=0}^{n-k} x_{i+k}$$

5. Separating Sums:

$$\sum_{i=1}^{n}(x_i + y_i) = \sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i$$

Warning

Summation notation **does not distribute over multiplication or division**. That is,

$$\sum_{i=1}^{n}(x_i y_i) \neq \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)$$

$$\sum_{i=1}^{n} \frac{x_i}{y_i} \neq \frac{\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} y_i}$$

For more practice, work through Activity 2 (which has 5 sub-activities).

Activity 2

Activity 2

Activity 2.1

Given that $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$, determine the value of $\sum_{j=3}^{10}(j^2 - 7j)$.

Activity 2.2

Show that $\sum_{n=1}^{100} \frac{1}{n(n+1)} = 1 - \frac{1}{101}$.
Hint: Try writing $\frac{1}{n(n+1)}$ as a difference of two fractions.
Activity 2.3
Recall, $n! = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 1$.
The Taylor Series expansion for the function $f(x) = e^x$, at the point $x = 0$, is given by:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

Find a closed form expression for the infinite series:

$$\frac{1}{2!} - \frac{1}{3!} + \frac{1}{4!} - \frac{1}{5!} + \frac{1}{6!} - \frac{1}{7!} + \cdots$$

Activity 2.4
$\binom{n}{k}$, pronounced "n choose k", is the number of ways to choose k items from a set of n items.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

For example, $\binom{5}{2} = 10$, because there are 10 ways to choose 2 items from a set of 5 items, and $\frac{5!}{2!(5-2)!} = \frac{120}{2 \cdot 6} = 10$.
Using the fact introduced in Activity 2.1, find a closed form expression for the following sum:

$$\sum_{k=2}^{n} \binom{k}{2}$$

Activity 2.5
Argue why the following equality holds (it'll be hard to prove this algebraically):

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

Mean and Standard Deviation

As I mentioned at the start of this section, we'll work with sums of data points quite frequently in this class. We'll often set up a problem by saying we have a sequence of n scalar[1] values, represented by $x_1, x_2, \ldots, x_n$. For instance, perhaps there are n students in this course, and x_i represents the height of student i.

Mean. The **mean**, or **average**, of all n values is given the symbol $\bar{x}$ (pronounced "x-bar") and is defined as follows:

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

[1] "Scalar" just means "individual number", as opposed to a vector or matrix which can contain multiple numbers, as we'll see later in the course.

You've likely seen this definition before. But, an often-forgotten property of the mean is that the **sum of the deviations from the mean is zero**. By that, I mean (no pun intended) that if you:

1. compute the mean of a sequence of numbers,
2. compute the *signed* difference between each number and the mean, and then
3. sum all of those differences, the result will be zero.

Let's first see this in action, then show why it is true in general. Suppose there are only 4 students in the class, with heights 72, 63, 68, and 65 inches. The mean of these heights is:

$$\bar{x} = \frac{72 + 63 + 68 + 65}{4} = 67$$

The deviations from the mean are:

$$\begin{aligned} 72 - 67 &= 5 \\ 63 - 67 &= -4 \\ 68 - 67 &= 1 \\ 65 - 67 &= -2 \end{aligned}$$

The sum of the four deviations, then, is:

$$5 + (-4) + 1 + (-2) = 0$$

So, the mean deviation from the mean is zero in this example.

This is also true in general. Precisely, I'm claiming that if $x_1, x_2, ..., x_{n-1}, x_n$ are any n numbers, and $\bar{x}$ is their mean, then $\displaystyle\sum_{i=1}^{n}(x_i - \bar{x}) = 0$.

Let's prove it:

$$\begin{aligned} \sum_{i=1}^{n}(x_i - \bar{x}) &= \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \bar{x} \\ &= \sum_{i=1}^{n} x_i - n\bar{x} \\ &= \sum_{i=1}^{n} x_i - n\left(\frac{x_1 + x_2 + \ldots + x_n}{n}\right) \\ &= \sum_{i=1}^{n} x_i - (x_1 + x_2 + \ldots + x_n) \\ &= 0 \end{aligned}$$

So, we've shown that the sum of the deviations from the mean is 0 in general. A consequence of this is that the positive deviations and negative deviations are equal in magnitude, since they need to cancel each other out. In the 72, 63, 68, 65 example, the positive deviations are 5 and 1 and the negative deviations are -4 and -2, and both have magnitude 6. As a result, the mean is sometimes thought of as the "balance point" of the dataset – the point at which the negative deviations are balanced by the positive deviations. More on this in Chapter 1.3.

Standard Deviation. Since the sum (and average) of deviations from the mean is 0, no matter the dataset, we can't use the average deviation from the mean to measure how far values tend to be from the mean. The average deviation will be 0, whether the dataset is tightly clustered or spread out.

To measure how far values in a dataset tend to deviate from their mean, then, we'll need to address the fact that some deviations are positive and some are negative. A common approach is to:

1. Compute the mean of the dataset
2. Compute the deviation of each value from the mean
3. Square each deviation
4. Take the average of the squared deviations

The result of this process is called the **variance**, denoted s^2 or σ^2; its square root is called the **standard deviation**.

Following the above steps, the variance is given by:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

(If you've read Chapter 1.2, you'll notice some similarities between this formula and the formula for mean squared error.)

Activity 3

Activity 3

What is the variance of the dataset 72, 63, 68, 65?

As a final note, the variance has a convenient, equivalent form:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n}\sum_{i=1}^{n}x_i^2 - \bar{x}^2$$

In English, we might say the variance is "the mean of the squares of x minus the square of the mean of x".

Your turn: prove this equivalent form of the variance.

Solution

$$
\begin{aligned}
\sigma^2 &= \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}(x_i^2 - 2x_i\bar{x} + \bar{x}^2) \\
&= \frac{1}{n}\sum_{i=1}^{n}x_i^2 - \frac{2}{n}\sum_{i=1}^{n}x_i\bar{x} + \frac{1}{n}\sum_{i=1}^{n}\bar{x}^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}x_i^2 - 2\bar{x}\frac{1}{n}\sum_{i=1}^{n}x_i + \bar{x}^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}x_i^2 - 2\bar{x}^2 + \bar{x}^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}x_i^2 - \bar{x}^2
\end{aligned}
$$

Activity 4

Activity 4

Suppose the dataset $x_1, x_2, \ldots, x_n$ has mean $\bar{x}$ and variance σ^2.
Find the mean and variance of the dataset $-4x_1+3, -4x_2+3, \ldots, -4x_n+3$, and justify your answer rigorously using the definitions of mean and variance.

Appendix 2: Derivatives

Calculus is the study of **rates of change**, and without it, modern machine learning would not be possible. In many ways, machine learning is about **optimizing** quantities – making the *best* possible predictions, or making prediction errors as *small* as possible – and calculus is the tool that enables us to perform this optimization.

We'll address these lofty goals throughout the semester. Here, we will review key ideas from a first course in calculus (e.g. Math 115).

Tangent Lines

Suppose f is a function that takes in a single real number and outputs a single real number, i.e. $f : \mathbb{R} \to \mathbb{R}$.

If f is a function, then the **derivative** of f is another function, sometimes denoted f', such that $f'(x)$ is the "instantaneous" rate of change of f at the input x.

To understand what I mean by "instantaneous" rate of change, let's consider an example. Suppose $f(x) = \frac{1}{4}x^2 - 3$. The graph of f is shown below, in blue, along with a slider for input values of x. Drag the slider.

At any point x, the tangent line to the graph of f is the **best linear approximation** of the graph of f near x.

For instance, consider when $x = -4$. At $x = -4$, $f(-4) = \frac{1}{4}(-4)^2 - 3 = 1$.

The tangent line at $x = -4$ is the line that passes through the point $(-4, 1)$ that best approximates f near $x = -4$, among all other lines that pass through $(-4, 1)$. In the plot above, use your mouse to zoom in on the point $(-4, 1)$, and you'll see that when zoomed in, the tangent line and original function are very difficult to distinguish.

The Derivative is the Slope of the Tangent Line

Here's the million dollar connection. The slope of the tangent line at x **is, by definition**, the derivative of f at x.
In the example above, when $x = -4$, the slope of the tangent line is -2, which is the derivative of f at $x = -4$, i.e. $f'(-4) = -2$.

This intuitive definition of the derivative – as being the slope of the tangent line – is the most important way to think about the derivative. The formal definition is also important, and we'll get there next, but you should have enough context for this first activity.

Activity 1

Activity 1

Let $f(x) = \frac{1}{4}x^4 + \frac{1}{3}x^3 - x^2 + 2$.
Given that $f'(x) = x^3 + x^2 - 2x$, find the equation of the tangent line to $f(x)$ at the following points:

- $x = -3$
- $x = -1$
- $x = 1$

As a bonus exercise, try to verify that the provided formula for $f'(x)$ is correct, using your knowledge of derivatives from Calculus 1.

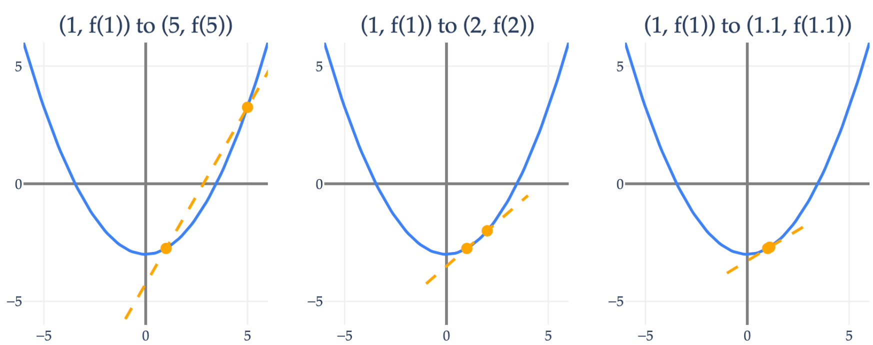
Formal Definition

Secant Lines. Let's review the more formal definition of the derivative. First, remember the general formula for the slope of the line between two points (x_1, y_1) and (x_2, y_2):

$$\frac{y_2 - y_1}{x_2 - x_1}$$

Let's say we're trying to find the slope of the tangent line at $x = a$, which is the line that passes through the point $(a, f(a))$ whose slope is the instantaneous rate of change of f at $x = a$. To find that instantaneous rate of change, we can find the slope of the line between $(a, f(a))$ and some other point $(b, f(b))$, where $b - a$ is as close to 0 as possible.

In the example below, as b approaches $a = 1$, the slope of the line between $(1, f(1))$ and $(b, f(b))$ approaches the slope of the tangent line at $x = 1$. Note that the formal name for the line between any two points on a function is a **secant line**.



Limits. Let's be more precise, using the idea of a **limit** from Calculus 1. Recall, $\lim_{x \to a} g(x) = L$ is pronounced "the limit of $g(x)$ as x approaches a is L". If $\lim_{x \to a} g(x) = L$, then as x gets closer and closer to a, $g(x)$ gets closer and closer to L. (Intuitively, you might think this must always mean that $g(a) = L$, but that's not always the case.)

The slope of the tangent line at $x = a$ is the limit of the slope of the line between $(a, f(a))$ and $(b, f(b))$ as b approaches a:

$$\text{slope of tangent line at } a = \lim_{b \to a} \frac{f(b) - f(a)}{b - a}$$

This definition alone can help us compute some common derivatives. For instance, if $f(x) = x^2$, then the slope of the tangent line at $x = a$ is:

$$\lim_{b \to a} \frac{f(b) - f(a)}{b - a} = \lim_{b \to a} \frac{b^2 - a^2}{b - a} = \lim_{b \to a} \frac{(b - a)(b + a)}{b - a} = \lim_{b \to a} (b + a) = 2a$$

Here, I used the difference of squares formula, $b^2 - a^2 = (b - a)(b + a)$. To find the slope of the tangent line using the limit definition, you'll need to use some sort of algebraic manipulation to simplify the expression, because the limit of the denominator is 0, and we can't divide by 0.

Instead of thinking of the slope of the tangent line as the limit of the slope of the line between the points $(a, f(a))$ and $(b, f(b))$ as $b \to a$, a more general (and equivalent!) definition of the derivative is the limit of the slope of the line between the points $(x, f(x))$ and $(x + h, f(x + h))$ as $h \to 0$.

Derivatives.

Definition: Derivative

Suppose $f : \mathbb{R} \to \mathbb{R}$ is a function. Then the **derivative** of f at x is defined as:

$$\frac{\mathrm{d}f}{\mathrm{d}x}(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

This is pronounced "the derivative of f with respect to x."

I will not use this formal definition much in this class, but it's good to understand where it comes from and why it works.

There are two equivalent notations for the derivative: $\frac{\mathrm{d}f}{\mathrm{d}x}(x)$ and $f'(x)$. I used the notation $f'(x)$ in the previous section since it's easier to write and more commonly used in calculus courses. However, I'll use the notation $\frac{\mathrm{d}f}{\mathrm{d}x}(x)$ from now on, as it'll make the transition to multivariable calculus more natural when we get there.

Often, for brevity, I will drop the (x) and just write $\frac{\mathrm{d}f}{\mathrm{d}x}$. As an example, suppose $g(x) = \sin^2(x) + 3\log(x)$, where $\log(\cdot)$ is the natural logarithm (with base e). Then, the derivative of g is:

$$\frac{\mathrm{d}g}{\mathrm{d}x} = 2\sin(x)\cos(x) + \frac{3}{x}$$

$\frac{\mathrm{d}g}{\mathrm{d}x}$ (equivalently, $\frac{\mathrm{d}g}{\mathrm{d}x}(x)$) is a function, not a number. To get a number as an output, we need to plug in a value for x. For example, $\frac{\mathrm{d}g}{\mathrm{d}x}(\pi)$ is the number ($\frac{3}{\pi}$) corresponding to the slope of the tangent line to g at $x = \pi$.

To actually find $\frac{\mathrm{d}g}{\mathrm{d}x}$, we used several derivative rules, which we'll now review.

Rules and Examples

Five Key Derivative Rules

1. **Constant Rule**: If $f(x) = c$, where c is some constant, then:

$$\frac{\mathrm{d}f}{\mathrm{d}x} = 0$$

2. **Addition Rule**: If $h(x) = f(x) + g(x)$, then:

$$\frac{\mathrm{d}h}{\mathrm{d}x} = \frac{\mathrm{d}f}{\mathrm{d}x} + \frac{\mathrm{d}g}{\mathrm{d}x}$$

   Equivalently, $h'(x) = f'(x) + g'(x)$.

3. **Power Rule**:

$$\frac{\mathrm{d}}{\mathrm{d}x}x^n = nx^{n-1}$$

4. **Product Rule**: If $h(x) = f(x)g(x)$, then:

$$\frac{\mathrm{d}h}{\mathrm{d}x} = f \cdot \frac{\mathrm{d}g}{\mathrm{d}x} + \frac{\mathrm{d}f}{\mathrm{d}x} \cdot g$$

Equivalently, $h'(x) = f(x)g'(x) + f'(x)g(x)$.

5. **Chain Rule**: If $h(x) = f(g(x))$, then:

$$\frac{\mathrm{d}h}{\mathrm{d}x} = \frac{\mathrm{d}f}{\mathrm{d}g} \cdot \frac{\mathrm{d}g}{\mathrm{d}x}$$

Equivalently, $h'(x) = f'(g(x))g'(x)$.

These rules can all be proved using the formal definition of the derivative, but those proofs won't be emphasized in this class.

Attempt each of the following examples before peeking at their solutions.

Example: Polynomials. Differentiate $f(x) = 4x^5 + 3x^2 + 2x + 1$.

Solution

None of the five key rules directly specify how to take the derivative of an expression like $4x^5$, and even though you *may* look at this expression and see that its derivative is $20x^4$, it's well worth our time to review the rules carefully.

Let's split $4x^5$ into two separate functions, 4 and x^5, and then use the product rule.

$$\frac{\mathrm{d}}{\mathrm{d}x}4x^5 = 4\left(\frac{\mathrm{d}}{\mathrm{d}x}x^5\right) + \left(\frac{\mathrm{d}}{\mathrm{d}x}4\right)x^5 = 4 \cdot 5x^4 + 0 = 20x^4$$

While applying the product rule, I also used the power rule to differentiate x^5, and the constant rule to differentiate 4.

With this in mind, we can return to the goal of differentiating $f(x)$.

$$\begin{aligned}\frac{\mathrm{d}f}{\mathrm{d}x} &= \frac{\mathrm{d}}{\mathrm{d}x}(4x^5 + 3x^2 + 2x + 1)\\ &= \frac{\mathrm{d}}{\mathrm{d}x}4x^5 + \frac{\mathrm{d}}{\mathrm{d}x}3x^2 + \frac{\mathrm{d}}{\mathrm{d}x}2x + \frac{\mathrm{d}}{\mathrm{d}x}1\\ &= 20x^4 + 6x + 2\end{aligned}$$

Example: Reciprocals. Differentiate $f(x) = \frac{1}{x^2-1}$.

Solution

Note that $f(x)$ can also be written as $f(x) = (x^2 - 1)^{-1}$, which looks like something we can use the power rule on. To fully apply the power rule, we'll need to use the chain rule, too.

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}x}(x^2-1)^{-1} &= \underbrace{-1 \cdot (x^2-1)^{-2}}_{\text{power rule}} \cdot \underbrace{\frac{\mathrm{d}}{\mathrm{d}x}(x^2-1)}_{\text{required by the chain rule}} \\
&= -1 \cdot (x^2-1)^{-2} \cdot 2x \\
&= -\frac{2x}{(x^2-1)^2}
\end{aligned}$$

Example: Quotients. Differentiate $f(x) = \frac{x^2+1}{x^2-1}$.

Solution

It looks like $f(x)$ is a quotient of two functions, $x^2 + 1$ and $x^2 - 1$, but I intentionally did not introduce a quotient rule – it's unnecessary, and can be replicated using the product rule and chain rule, since $f(x)$ can be written as a product:

$$f(x) = \frac{x^2+1}{x^2-1} = (x^2+1) \cdot \frac{1}{x^2-1}$$

You'll notice that we've already found the derivative of $\frac{1}{x^2-1}$ in Example 2, so we can use that result here while applying the product rule.

$$\begin{aligned}
\frac{\mathrm{d}f}{\mathrm{d}x} &= \frac{\mathrm{d}}{\mathrm{d}x}\left((x^2+1) \cdot \frac{1}{x^2-1}\right) \\
&= \underbrace{\left(\frac{\mathrm{d}}{\mathrm{d}x}(x^2+1)\right) \cdot \frac{1}{x^2-1} + (x^2+1) \cdot \left(\frac{\mathrm{d}}{\mathrm{d}x}\frac{1}{x^2-1}\right)}_{\text{product rule}} \\
&= 2x \cdot \frac{1}{x^2-1} + (x^2+1) \cdot \underbrace{\left(-\frac{2x}{(x^2-1)^2}\right)}_{\text{from Example 2}} \\
&= \frac{2x}{x^2-1} - \frac{2x(x^2+1)}{(x^2-1)^2} \\
&= \underbrace{\frac{2x(x^2-1) - 2x(x^2+1)}{(x^2-1)^2}}_{\text{common denominator}} \\
&= \frac{2x(x^2-1-x^2-1)}{(x^2-1)^2} \\
&= \frac{-4x}{(x^2-1)^2}
\end{aligned}$$

Example: Trigonometric and Logarithmic Functions. Differentiate $f(x) = \log(\cos(e^x))$.

To do so, you'll need to remember a few other common derivatives that the five key rules don't cover.

Trigonometric Derivatives:

$$\frac{\mathrm{d}}{\mathrm{d}x}\sin(x) = \cos(x)$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\cos(x) = -\sin(x)$$

Logarithmic Derivatives: If $\log(\cdot)$ is the natural logarithm (with base e), then:

$$\frac{\mathrm{d}}{\mathrm{d}x}\log(x) = \frac{1}{x}$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\log_b(x) = \frac{1}{x\log(b)}$$

Exponential Derivatives:

$$\frac{\mathrm{d}}{\mathrm{d}x}e^x = e^x$$

$$\frac{\mathrm{d}}{\mathrm{d}x}a^x = a^x\log(a)$$

Solution

Differentiating $f(x) = \log(\cos(e^x))$ requires a few careful applications of the chain rule.

$$\begin{aligned}
\frac{\mathrm{d}f}{\mathrm{d}x} &= \frac{\mathrm{d}}{\mathrm{d}x}\log(\cos(e^x)) \\
&= \frac{1}{\cos(e^x)} \cdot \frac{\mathrm{d}}{\mathrm{d}x}\cos(e^x) \\
&= \frac{1}{\cos(e^x)} \cdot (-\sin(e^x)) \cdot \frac{\mathrm{d}}{\mathrm{d}x}e^x \\
&= \frac{1}{\cos(e^x)} \cdot (-\sin(e^x)) \cdot e^x \\
&= -\frac{e^x\sin(e^x)}{\cos(e^x)} \\
&= -e^x\tan(e^x)
\end{aligned}$$

The chain rule is **extremely** pervasive in machine learning, which is why I've included examples like the one above. This site contains dozens more examples of the chain rule in practice.

Activity 2

Activity 2

Note that the activities in this section are quite challenging, so make sure you've attempted and fully understood the examples above first.

Activity 2.1

An important function in machine learning is the **sigmoid function**, which is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$\sigma(x)$ has a nice S-shape, and is used for predicting probabilities.
Find the derivative of $\sigma(x)$, and show that it satisfies the following property:

$$\frac{\mathrm{d}}{\mathrm{d}x}\sigma(x) = \sigma(x)(1 - \sigma(x))$$

Activity 2.2

Find the derivative of each of the following functions:

- $f(x) = \sqrt{\sin(4\pi x)}$
- $g(x) = (2x + 1)^{3x}$ (Hint: Start by taking the natural logarithm of both sides, then take the derivative of both sides.)

Activity 2.3

Suppose x and y satisfy the following relationship:

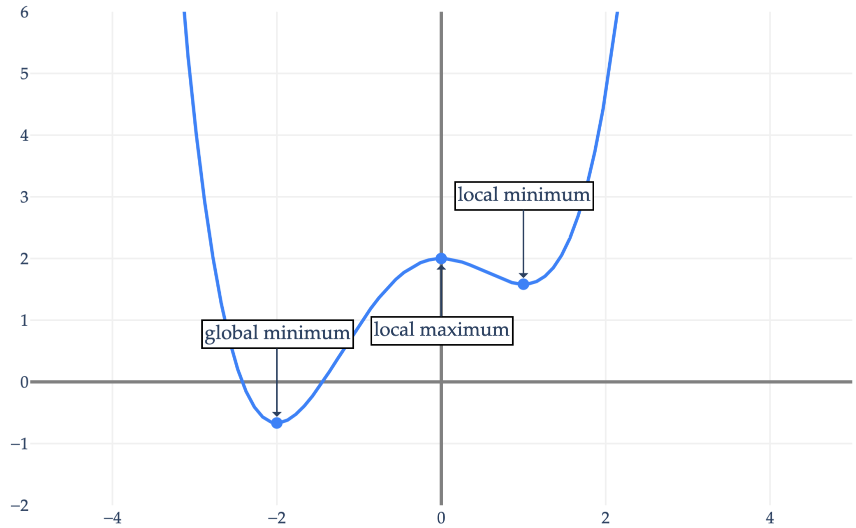
$$x^2 = y^3 - 11$$

1. Find an expression for $\frac{\mathrm{d}y}{\mathrm{d}x}$ that involves **both** x and y. To do this, don't solve for y in terms of x – instead, take the derivative of both sides of the equation with respect to x, use the power and chain rules, and re-arrange to isolate $\frac{\mathrm{d}y}{\mathrm{d}x}$.
2. Find the slope of the tangent line to the curve at the point $(x, y) = (4, 3)$.

Optimization

Maxima and Minima. As stated at the start of this section, calculus is a tool for optimization – that is, finding the inputs that maximize or minimize a function. Let's be more precise about what we mean by "maximize" and "minimize".

Consider $f(x) = \frac{1}{4}x^4 + \frac{1}{3}x^3 - x^2 + 2$, shown below.



Where is $f(x)$ maximized and minimized?

- At $x = -2$, $f(x)$ is less than it is at all other inputs. This is means $(-2, f(-2))$ is a **global minimum**.
- At $x = 0$, $f(x)$ *looks like* it is greater than all other inputs, but only if you restrict your attention to points near $x = 0$. This means $(0, f(0))$ is a **local maximum**. $(0, f(0))$ is not a **global maximum** because there are plenty of points where $f(x) > f(0)$ – they are just not immediately adjacent to $x = 0$.
- Similarly, $(1, f(1))$ is a **local minimum**.

$f(x)$ does not have a global maximum, since $f(x)$ approaches infinity as x increases beyond $x = 1$ or decreases beyond $x = -2$. If we were to restrict the domain (or set of possible inputs) of $f(x)$ to, say, $[-3, 3]$, then there would be a global maximum at $x = 3$.

Note that we usually care more about the inputs that maximize or minimize a function, rather than the actual values of the function at those inputs. In the above example, the fact that $x = -2$ is a global minimum is important; the fact that $f(-2) = -\frac{2}{3}$ is not *as* important.

To recap:

Definition: Maxima and Minima

Suppose $f : \mathbb{R} \to \mathbb{R}$ is a function.
On maxima:

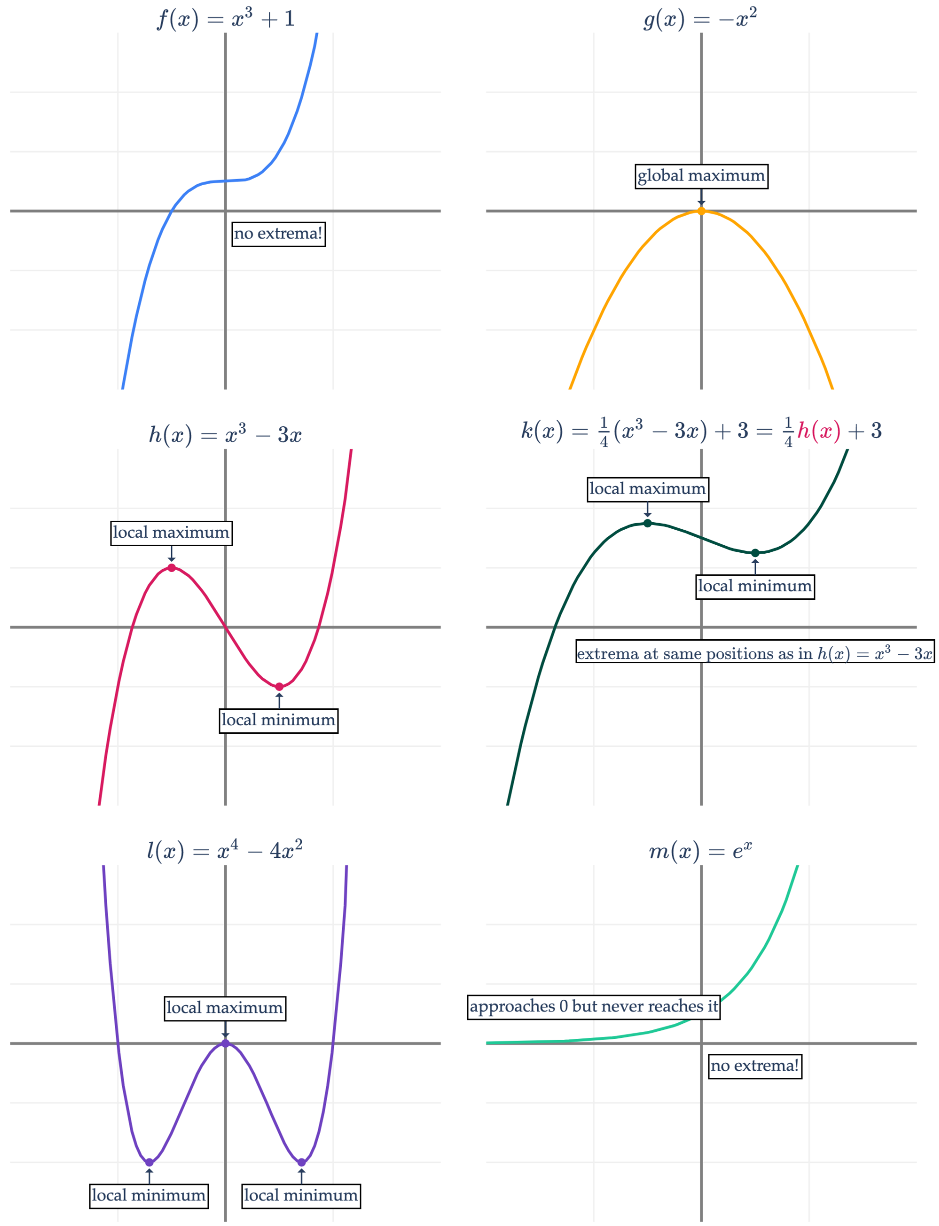
- A function $f : \mathbb{R} \to \mathbb{R}$ has a **local maximum** at $x = x^*$ if $f(x^*) \geq f(x)$ for all x in some interval around x^*.
- A function $f : \mathbb{R} \to \mathbb{R}$ has a **global maximum** at $x = x^*$ if $f(x^*) \geq f(x)$ for all x in the domain of f.

Intuitively, a local maximum is a point where the function is higher than all nearby points, and a global maximum is a point where the function is higher than all points in the domain.
On minima:

- A function $f : \mathbb{R} \to \mathbb{R}$ has a **local minimum** at $x = x^*$ if $f(x^*) \leq f(x)$ for all x in some interval around x^*.
- A function $f : \mathbb{R} \to \mathbb{R}$ has a **global minimum** at $x = x^*$ if $f(x^*) \leq f(x)$ for all x in the domain of f.

Note that maxima is the plural of maximum, minima is the plural of minimum, and extrema refers to both maxima and minima.

Below, you'll find several other examples of functions with varying amounts and types of extrema. Play close attention to the relationship between the two functions in the second row, $h(x)$ and $k(x)$. $k(x)$ results from stretching $h(x)$ vertically and shifting it up vertically, and has the same extrema.



Activity 3

Activity 3

Let $q(x) = 8x^4 - 4x$. $q(x)$ has a global minimum at $\left(\frac{1}{2}, -\frac{3}{2}\right)$.
For each of the following functions, find all extrema, and specify whether each extremum is a local maximum, global maximum, local minimum, or global minimum. Make sure to specify both the x-values and the y-values of each extremum.

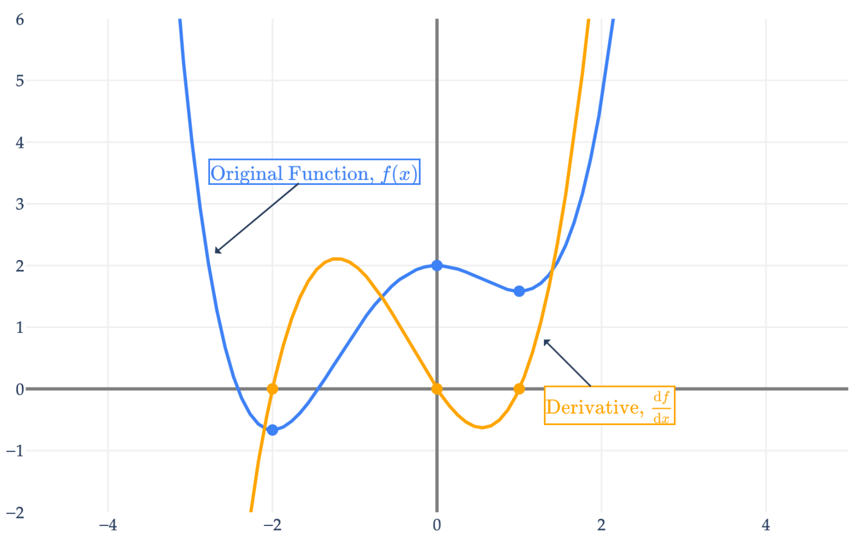
1. $f(x) = 2q(x) + 10$
2. $g(x) = -10q(x)$
3. $h(x) = |q(x)|$
4. Finding the extrema of $l(x) = q(x)^2$ is a bit more complicated than in the examples above. Why?

Critical Points. How do we actually find the critical points of a function, especially when we can't graph the function? The derivative plays a crucial role.

Definition: Critical Point

A **critical point** of a function $f : \mathbb{R} \to \mathbb{R}$ is a point $(a, f(a))$ where $\frac{\mathrm{d}f}{\mathrm{d}x}(a) = 0$ or $\frac{\mathrm{d}f}{\mathrm{d}x}(a)$ is undefined.

Let's revisit the function $f(x) = \frac{1}{4}x^4 + \frac{1}{3}x^3 - x^2 + 2$, shown in blue along with its derivative, $\frac{\mathrm{d}f}{\mathrm{d}x} = x^3 + x^2 - 2x$, shown in orange.



You should notice that the derivative is 0 at all three extrema we identified earlier – the global minimum at $x = -2$, the local maximum at $x = 0$, and the local minimum at $x = 1$. Intuitively, the derivative is 0 at a maximum or minimum because the tangent lines at these points are horizontal (with slope 0), as the function is neither increasing nor decreasing at these points.

In the region between $x = -2$ and $x = 0$, the derivative is positive, meaning the function is increasing.

Solving for the inputs that make the derivative 0 – i.e., finding the critical points – is a necessary, but not sufficient, step. If all we know is that the derivative is 0 at a point, we don't know whether the point is a maximum or minimum. It may not be either, such as in the case of $f(x) = x^3$, which has a critical point at $x = 0$ that is neither a maximum nor a minimum.

Second Derivatives. To be able to determine whether a critical point of $f(x)$ is a maximum or minimum, we need to look at the **second derivative** of $f(x)$. If the (first) derivative of $f(x)$ is a function that describes the rate at which $f(x)$ is changing, the second derivative – denoted $\frac{\mathrm{d}^2 f}{\mathrm{d}x^2}$ – is a function that describes the rate at which the derivative is changing.

Physics provides us with an analogy that helps us understand the role of the second derivative. Suppose you're driving down a straight road, and $s(t)$ is your position on the road at time t, relative to your starting point (so a negative value of $s(t)$ means you've moved backwards).

Then, $v(t) = \frac{\mathrm{d}s}{\mathrm{d}t}$ is your velocity (the rate at which your position is changing) and $a(t) = \frac{\mathrm{d}^2 s}{\mathrm{d}t^2}$ is your acceleration (the rate at which your velocity is changing).

- If $\frac{\mathrm{d}s}{\mathrm{d}t} > 0$ and $\frac{\mathrm{d}^2 s}{\mathrm{d}t^2} = 0$, you are moving forward at a constant speed (say, on cruise control).
- If $\frac{\mathrm{d}s}{\mathrm{d}t} > 0$ and $\frac{\mathrm{d}^2 s}{\mathrm{d}t^2} > 0$, you are moving forward and your speed is increasing (you are accelerating).
- If $\frac{\mathrm{d}s}{\mathrm{d}t} > 0$ and $\frac{\mathrm{d}^2 s}{\mathrm{d}t^2} < 0$, you are moving forward, but your speed is decreasing, and **eventually, your car will come to a halt**.
- Cases where $\frac{\mathrm{d}s}{\mathrm{d}t} < 0$ correspond to driving backwards!

Activity 4

Activity 4

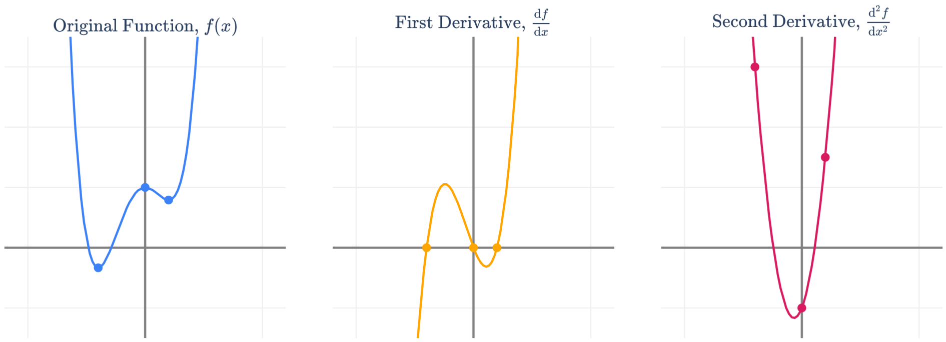
Give real-world examples (similar to those provided above) for each of the following scenarios in the context of the physics analogy:

- $\frac{\mathrm{d}s}{\mathrm{d}t} < 0$ and $\frac{\mathrm{d}^2 s}{\mathrm{d}t^2} > 0$
- $\frac{\mathrm{d}s}{\mathrm{d}t} < 0$ and $\frac{\mathrm{d}^2 s}{\mathrm{d}t^2} < 0$
- $\frac{\mathrm{d}s}{\mathrm{d}t} = 0$ and $\frac{\mathrm{d}^2 s}{\mathrm{d}t^2} < 0$

Let's put this in the context of our running example, $f(x) = \frac{1}{4}x^4 + \frac{1}{3}x^3 - x^2 + 2$. The second derivative of $f(x)$ is:

$$
\begin{aligned}
\frac{\mathrm{d}^2 f}{\mathrm{d}x^2} &= \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{4}x^4 + \frac{1}{3}x^3 - x^2 + 2\right)\right) \\
&= \frac{\mathrm{d}}{\mathrm{d}x}\left(\underbrace{x^3 + x^2 - 2x}_{\text{first derivative of } f(x)}\right) \\
&= 3x^2 + 2x - 2
\end{aligned}
$$

The second derivative, $\frac{\mathrm{d}^2 f}{\mathrm{d}x^2}$, is a function, not a number. What does the second derivative look like, relative to the original function and first derivative?



$f(x)$ is a polynomial of degree 4, $\frac{\mathrm{d}f}{\mathrm{d}x}$ is a polynomial of degree 3, and $\frac{\mathrm{d}^2f}{\mathrm{d}x^2}$ is a polynomial of degree 2 – the degree drops by one each time, as a consequence of the power rule.

Recall that $f(x)$ has critical points at $x = -2$, $x = 0$, and $x = 1$, which we've highlighted in all three plots above. Our goal is to determine an algebraic approach for determining whether these points are maxima, minima, or neither; we shouldn't rely on the graph of $f(x)$ alone, since we won't always be able to see its graph.

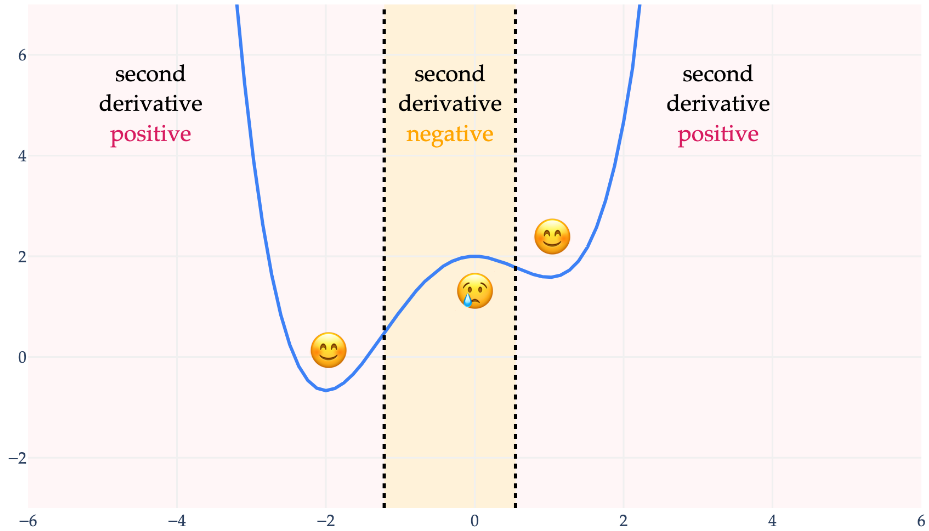
At all three of these points, the first derivative is 0, meaning that the function is neither increasing nor decreasing at these points. But the second derivative $\frac{\mathrm{d}^2f}{\mathrm{d}x^2} = 3x^2 + 2x - 2$ gives us additional information:

- At $x = -2$, $\frac{\mathrm{d}^2f}{\mathrm{d}x^2}(-2) = 6$, which is **positive**. So, at $x = -2$, $f(x)$ is neither increasing nor decreasing, but is also "*speeding up*", since the second derivative is positive. So, as we move to the right of $x = -2$, the slope of the tangent line will increase, causing the function to increase. If the function increases to the right of $x = -2$, then $x = -2$ must correspond to a **local minimum** of $f(x)$.

- At $x = 0$, $\frac{\mathrm{d}^2f}{\mathrm{d}x^2}(0) = -2$, which is **negative**. So, at $x = 0$, $f(x)$ is neither increasing nor decreasing, but is also "*slowing down*". So, as we move to the right of $x = 0$, the slope of the tangent line will decrease, causing the function to decrease. If the function decreases to the right of $x = 0$, then $x = 0$ must correspond to a **local maximum** of $f(x)$.

- At $x = 1$, $\frac{\mathrm{d}^2f}{\mathrm{d}x^2}(1) = 3$ is **positive**, which, using the logic from the $x = -2$ case, means that $x = 1$ also corresponds to a **local minimum** of $f(x)$.

Convexity. The sign of the second derivative is useful for more than just determining whether a critical point is a local maximum or minimum. Below, we've plotted $f(x)$, along with annotations for the regions where the second derivative is positive and negative.



When the second derivative is positive, the function is **concave opening up**, also known as **convex**. You should think of convex functions as "bowl-shaped" or "smiling". When the second derivative is negative, the function is **concave opening down**, or simply **concave**; the equivalent analogy is that concave down regions are "upside-down bowls" or "sad faces".

From the perspective of finding local mimina, if a function is concave up at a critical point, then we must be at the bottom of a bowl – a local minimum – and if a function is concave down at a critical point, we must be at the top of a hill, corresponding to a local maximum.

If a function is concave up across its entire domain – **unlike** in the example above, but like in $f(x) = x^2$ – then any local minimum must be a **global minimum**. Convexity is a hugely important concept in optimization and machine learning, and we'll see it again in more detail throughout the course.

The points at which the second derivative is 0 are called **inflection points**. $f(x)$ has two inflection points, marked by vertical dotted lines above, roughly at $x = -1.22$ and $x = 0.55$. These are the roots of the quadratic equation $\frac{\mathrm{d}^2 f}{\mathrm{d}x^2} = 3x^2 + 2x - 2 = 0$.

We've implictly used a **second derivative test** for determining whether a critical point is a local maximum or minimum:

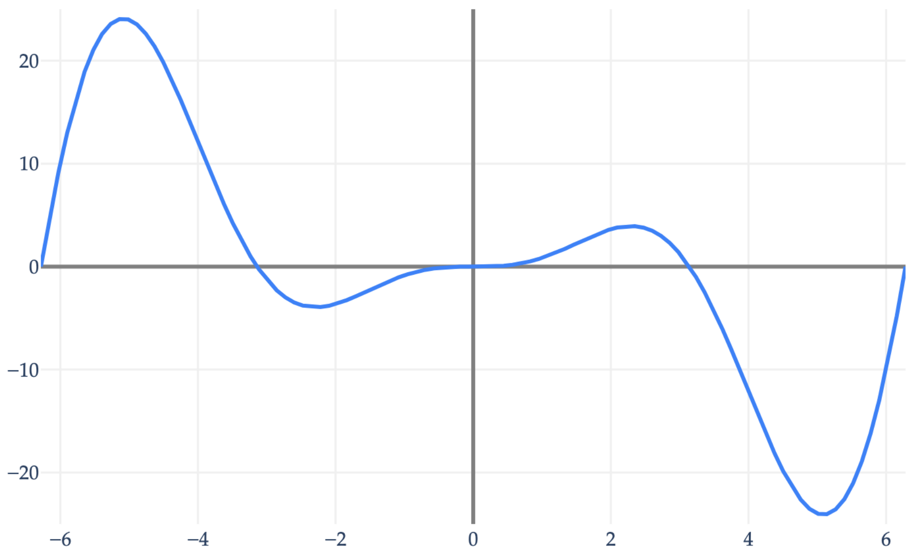
Definition: Second Derivative Test

Suppose a is a critical point of $f(x)$, i.e. $\frac{\mathrm{d}}{\mathrm{d}x} f(a) = 0$ or $\frac{\mathrm{d}}{\mathrm{d}x} f(a)$ is undefined. Then, there are 3 possibilities:

- If $\frac{\mathrm{d}^2 f}{\mathrm{d}x^2}(a) > 0$, then $f(x)$ is concave opening up (i.e. convex) at $x = a$, and $x = a$ is a local **minimum**.
- If $\frac{\mathrm{d}^2 f}{\mathrm{d}x^2}(a) < 0$, then $f(x)$ is concave opening down at $x = a$, and $x = a$ is a local **maximum**.
- If $\frac{\mathrm{d}^2 f}{\mathrm{d}x^2}(a) = 0$, then $x = a$ is an inflection point, and the second derivative test is inconclusive.

Again, the second derivative test only tries to tell us whether critical points are local maxima or minima; it **does not** tell us whether they are global maxima or minima.

Let's look at another example, particularly one where the second derivative test is inconclusive. Consider $f(x) = x^2 \sin(x)$, shown below.



$f(x) = x^2 \sin(x)$, like $\sin(x)$, is oscillatory, and has no global extrema; see **here** for a larger graph of it. Above, we've plotted $f(x)$ within the domain $[-2\pi, 2\pi]$, and we see several local maxima and minima.

The first and second derivatives of $f(x) = x^2 \sin(x)$ are given by:

$$\begin{aligned}\frac{\mathrm{d}f}{\mathrm{d}x} &= x^2 \left(\frac{\mathrm{d}}{\mathrm{d}x}\sin(x)\right) + \left(\frac{\mathrm{d}}{\mathrm{d}x}x^2\right)\sin(x) \\ &= x^2\cos(x) + 2x\sin(x)\end{aligned}$$

$$\begin{aligned}\frac{\mathrm{d}^2f}{\mathrm{d}x^2} &= \frac{\mathrm{d}}{\mathrm{d}x}\left(x^2\cos(x) + 2x\sin(x)\right) \\ &= x^2\left(\frac{\mathrm{d}}{\mathrm{d}x}\cos(x)\right) + \left(\frac{\mathrm{d}}{\mathrm{d}x}x^2\right)\cos(x) + 2x\left(\frac{\mathrm{d}}{\mathrm{d}x}\sin(x)\right) + \left(\frac{\mathrm{d}}{\mathrm{d}x}2x\right)\sin(x) \\ &= -x^2\sin(x) + 2x\cos(x) + 2x\cos(x) + 2\sin(x) \\ &= 2x\cos(x) - (x^2-2)\sin(x)\end{aligned}$$

Solving for the critical points of $f(x)$ by setting $\frac{\mathrm{d}f}{\mathrm{d}x} = x^2\cos(x) + 2x\sin(x) = 0$ is no easy task, as there are infinitely many solutions, most of which cannot be solved for by hand. Numerical optimization methods such as gradient descent can approximate solutions to $\frac{\mathrm{d}f}{\mathrm{d}x} = 0$. There are also infinitely many inflection points, since $\frac{\mathrm{d}^2f}{\mathrm{d}x^2} = 0$ has infinitely many solutions, meaning that there are many regions where $f(x)$ is concave up and many others where it is concave down.

However, one critical point is easy to spot: $x = 0$. At $x = 0$, the derivative is 0:

$$\frac{\mathrm{d}f}{\mathrm{d}x}(0) = 0^2\cos(0) + 2\cdot 0\cdot\sin(0) = 0$$

$x = 0$ is also an inflection point, since the second derivative is also 0:

$$\frac{d^2 f}{dx^2}(0) = 2 \cdot 0 \cdot \cos(0) - (0^2 - 2)\sin(0) = 0$$

To be clear, not every inflection point is a critical point, and not every critical point is an inflection point; $x = 0$ just happens to be both.

If we look at the graph of $f(x)$ near $x = 0$, we'll see that $x = 0$ corresponds to neither a local maximum nor a local minimum, but rather, a region where $f(x)$ is very flat. If we weren't able to graph $f(x)$, we could try and determine its behavior around $(0, f(0))$ by looking at points immediately to the left and right of $x = 0$ – say, $(0.001, f(0.001))$ and $(-0.001, f(-0.001))$. If $f(0.001) > f(0)$ and $f(-0.001) > f(0)$, then $x = 0$ would be a local minimum (but that's not the case here).

Activity 5

Activity 5

Activity 5.1
Let $f(x) = x\log(x^2)$, where $\log(\cdot)$ is the natural logarithm.

1. Find the critical points of $f(x)$, and determine whether they are local maxima, minima, or neither.
2. Find the inflection points of $f(x)$, and use them to sketch a possible graph of $f(x)$.

Activity 5.2
Let $g(3) = 10$, $\frac{dg}{dx}(3) = -2$, and $\frac{d^2 g}{dx^2}(3) = 1$.

1. Describe the behavior of $g(x)$ near $x = 3$.
2. The Taylor series of a function allows us to approximate the value of a function near a point $x = a$, given the value of the function and its derivatives at $x = a$. The Taylor series of an arbitrary function $f(x)$ around $x = a$ is given by:

$$f(x) \approx f(a) + \left(\frac{df}{dx}(a)\right)(x-a) + \frac{1}{2}\left(\frac{d^2 f}{dx^2}(a)\right)(x-a)^2 + \frac{1}{6}\left(\frac{d^3 f}{dx^3}(a)\right)(x-a)^3 + \frac{1}{24}\left(\frac{d^4 f}{dx^4}(a)\right)(x-a)^4 + \cdots$$

Note that this is an infinite series; the more terms we use, the more accurate the approximation.

Use the Taylor series to approximate the value of $g(3.1)$, using only the information provided. You'll only be able to use the first 3 terms of the Taylor series.

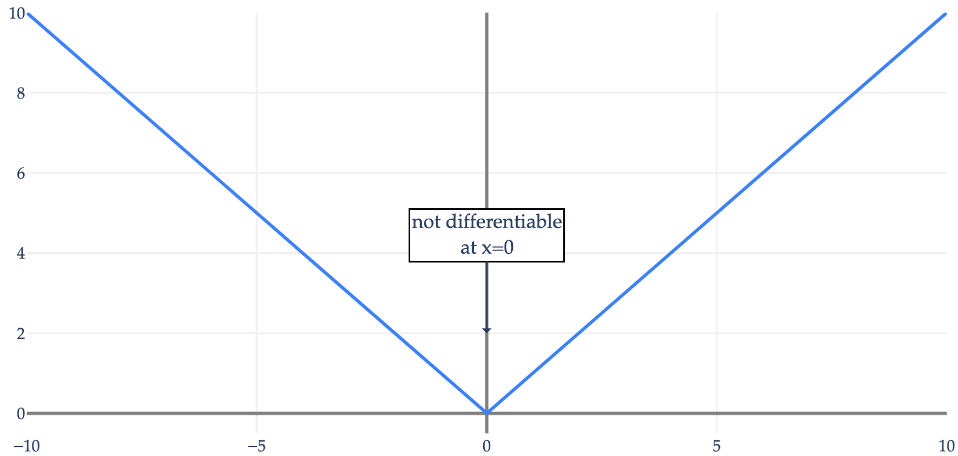
Activity 5.3
Given that $\frac{d^2 h}{dx^2} = 2x(x-3)(x+1)$, sketch a possible graph of $h(x)$.

Continuity and Differentiability

Finally, I'll remark that we've presented derivatives, extrema, and optimization all in the most ideal setting: where the functions we're working with are continuous and differentiable. A function is **continuous** if its graph can be drawn without lifting a pen; any point where the graph has a "jump" or "break" is a discontinuity. (Of course, there is a more formal definition of continuity, but this is a good enough illustration for now.) A function is **differentiable** if its derivative exists everywhere; otherwise, there exist some points at which the derivative does not exist.

Most relevant functions in machine learning are continuous, but non-differentiable functions do appear, so it's worth understanding what they are and how to deal with them. Let's look at a few examples.

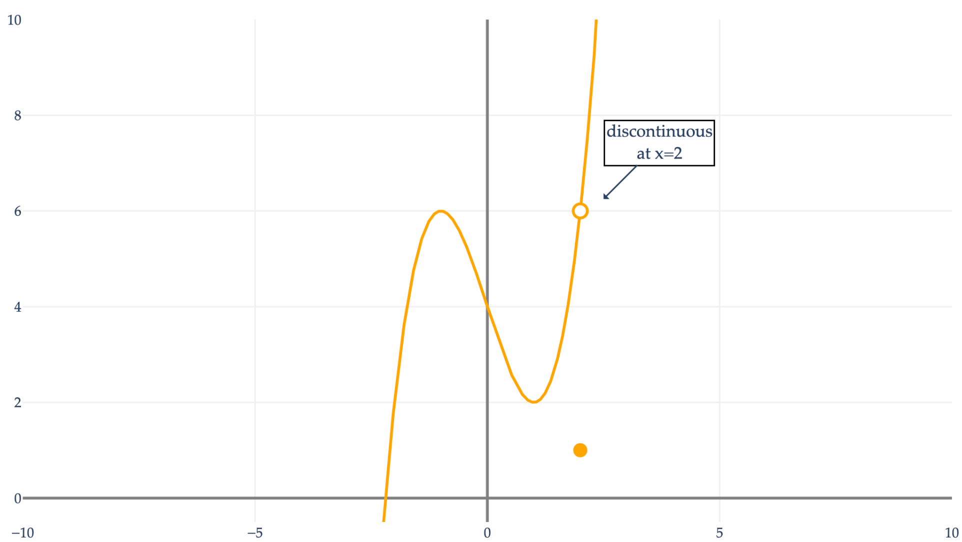
Example 1: $f(x) = |x|$



$f(x) = |x|$ is continuous everywhere, as intuitively, we can draw its graph without lifting our pen. It is differentiable everywhere **except** at $x = 0$; the reason it is not differentiable at $x = 0$ is that the slopes approaching it from the left (-1) and right (1) are different, and in order for a derivative at $x = a$ to exist, the limit of the slopes approaching a from the left and right must be the same.

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \begin{cases} -1 & x < 0 \\ 1 & x > 0 \\ \text{undefined} & x = 0 \end{cases}$$

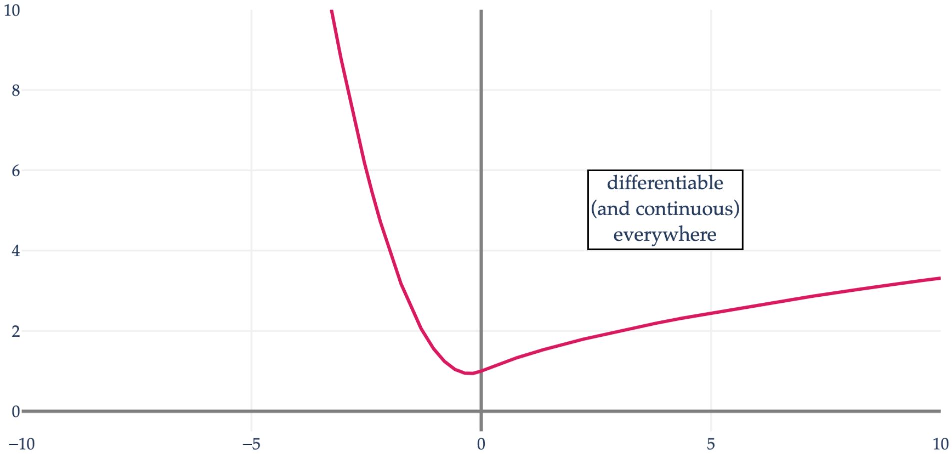
Example 2: $g(x) = \begin{cases} x^3 - 3x + 4 & x \neq 2 \\ 1 & x = 2 \end{cases}$



$g(x) = \begin{cases} x^3 - 3x + 4 & x \neq 2 \\ 1 & x = 2 \end{cases}$ is continuous and differentiable everywhere, except at $x = 2$, where it is neither.

$$\frac{\mathrm{d}g}{\mathrm{d}x} = \begin{cases} 3x^2 - 3 & x \neq 2 \\ \text{undefined} & x = 2 \end{cases}$$

Example 3: $h(x) = \begin{cases} x^2 + \frac{1}{2}x + 1 & x < 0 \\ \sqrt{x+1} & x \geq 0 \end{cases}$

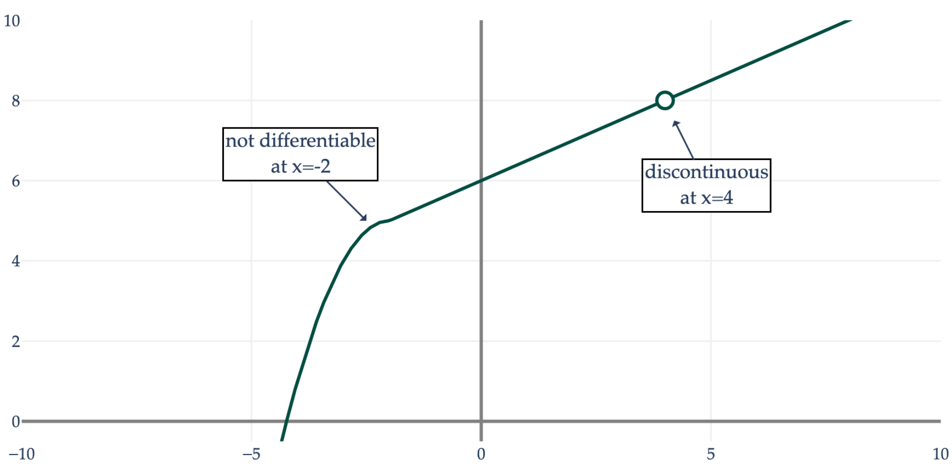


$h(x) = \begin{cases} x^2 + \frac{1}{2}x + 1 & x < 0 \\ \sqrt{x+1} & x \geq 0 \end{cases}$ is continuous and differentiable everywhere, despite being a piecewise function. Its individual pieces are continuous, and the entire function is continuous because the "left" and "right" functions at the connection point of $x = 0$ have the same value.

$$\frac{\mathrm{d}h}{\mathrm{d}x} = \begin{cases} 2x + \frac{1}{2} & x < 0 \\ \frac{1}{2\sqrt{x+1}} & x \geq 0 \end{cases}$$

Since the two piecewise derivatives agree at $x = 0$, $h(x)$ is differentiable at $x = 0$ (and across its entire domain).

Example 4: $k(x) = \begin{cases} -(x+2)^2 + 5 & x < -2 \\ \frac{1}{2}x + 6 & x \geq -2,\ x \neq 4 \\ \text{undefined} & x = 4 \end{cases}$



$k(x) = \begin{cases} -(x+2)^2 + 5 & x < -2 \\ \frac{1}{2}x + 6 & x \geq -2,\ x \neq 4 \\ \text{undefined} & x = 4 \end{cases}$ is continuous everywhere, except at $x = 4$, where it has a "jump" and is neither continuous nor differentiable. But in addition, $k(x)$ is not differentiable at $x = -2$ because the slopes approaching it from the left and right are different.

$$\frac{\mathrm{d}k}{\mathrm{d}x} = \begin{cases} -2(x+2) & x < -2 \\ \text{undefined} & x = -2 \\ \frac{1}{2} & x > -2,\ x \neq 4 \\ \text{undefined} & x = 4 \end{cases}$$

An important point is that any function that is differentiable everywhere is also continuous everywhere; differentiability is a stronger condition than continuity. Plenty of functions are continuous but not differentiable, like $f(x) = |x|$ in Example 1.