SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING
(Deemed to be University)
Department of Mathematics and Computer Science

LECTURE NOTES — Prerequisites, Insights

On Continuity and Differentiation

UMAT-503: Optimization Techniques	Instructor: Dr. D Bhanu Prakash	Date: 20-07-2026

Prerequisite Mathematical Concepts

1. Single-Variable and Multivariable Calculus

Derivatives and critical points. For a differentiable function $f(x)$, a necessary condition for a local extremum at $x = x_0$ is

$$f'(x_0) = 0.$$

Such points are called **critical points** (or stationary points). Not every critical point is an extremum — it could be a point of inflection.

Second derivative test. To classify a critical point x_0:

- $f''(x_0) > 0 \Rightarrow$ local minimum
- $f''(x_0) < 0 \Rightarrow$ local maximum
- $f''(x_0) = 0 \Rightarrow$ test is inconclusive (need higher-order terms)

Partial derivatives and the gradient. For $f: R^n \to R$, the gradient vector is

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right)^T.$$

The gradient points in the direction of **steepest ascent**, and $-\nabla f(x)$ points in the direction of steepest descent — this single fact is the entire justification for gradient descent.

The Hessian matrix. The matrix of second partial derivatives,

$$H(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix},$$

captures curvature in every direction simultaneously and generalizes the second-derivative test to many variables.

> **Mathematical detail — A zero gradient is not enough; classification needs the Hessian, and even the Hessian can lie about the whole picture.** In one variable, $f'(x_0) = 0$ plus $f''(x_0) > 0$ guarantees a minimum. In n variables, $\nabla f(x_0) = 0$ only tells you x_0 is a **critical point** — it could be a minimum, a maximum, or a **saddle point**. The classification rule uses the *definiteness* of $H(x_0)$: - $H(x_0) \succ 0$ (positive definite, all eigenvalues $\lambda_i > 0$) ⇒ local minimum - $H(x_0) \prec 0$ (all $\lambda_i < 0$) ⇒ local maximum - $H(x_0)$ has both positive and negative eigenvalues ⇒ **saddle point**
>
> A saddle point looks like a minimum if you only move along some directions (those with $\lambda_i > 0$) and like a maximum along others ($\lambda_i < 0$). Example: $f(x, y) = x^2 - y^2$ at the origin — moving along the x-axis you see a minimum, along the y-axis a maximum, yet $\nabla f(0, 0) = (0, 0)$. In high dimensions, the probability that *all* eigenvalues share the same sign shrinks fast, which is why saddle points dominate the critical-point landscape as n grows — a fact with real consequences for algorithms that rely only on $\nabla f = 0$ as a stopping criterion.

Taylor series expansion. Near a point x, a smooth function can be approximated as

$$f(x + h) \approx f(x) + \nabla f(x)^T h + \frac{1}{2} h^T H(x) h + O(\|h\|^3).$$

This quadratic approximation is exactly what Newton's method minimizes exactly at each step, and it is the reason convergence-rate proofs almost always start here.

Chain rule (multivariable). If $x = x(t)$, then

$$\frac{d}{dt} f(x(t)) = \nabla f(x(t))^T \frac{dx}{dt}.$$

This is essential when optimizing composite objectives (e.g. loss functions built from several nested functions).

Directional derivative. The rate of change of f at x in the direction of a unit vector u is

$$D_u f(x) = \nabla f(x)^T u.$$

This is maximized when u is parallel to $\nabla f(x)$, which is precisely why the gradient direction is the direction of steepest ascent.

2. Linear Algebra

Vector spaces, subspaces, and span. A vector space over R is a set closed under addition and scalar multiplication. Decision variables $x \in R^n$ live in such a space, and the *span* of a set of

vectors $\{v_1, \ldots, v_k\}$ — all their linear combinations $\sum_i c_i v_i$ — describes every direction reachable by combining them.

Linear independence, basis, and dimension. Vectors $v_1, \ldots, v_k$ are linearly independent if $\sum_i c_i v_i = 0$ forces every $c_i = 0$. A *basis* of a space is a linearly independent set that spans it, and the number of vectors in any basis is the space's *dimension*.

> *Example.* In R^3, the vectors $v_1 = (1, 0, 0)$, $v_2 = (0, 1, 0)$, $v_3 = (1, 1, 0)$ are **not** independent, since $v_3 = v_1 + v_2$. They span only the xy-plane (a 2-dimensional subspace), not all of R^3 — so they cannot be a basis for R^3, only for that plane.

Matrix operations, rank, and nullity. For an $m \times n$ matrix A: the **rank** is the dimension of its column space (the number of linearly independent columns), and the **rank–nullity theorem** states

$$rank(A) + dim(kerA) = n,$$

where $kerA = \{x: Ax = 0\}$ is the null space. In optimization, rank determines whether a linear system $Ax = b$ arising inside an algorithm (e.g. the KKT system, or $H\Delta x = -\nabla f$ in Newton's method) has a unique solution, no solution, or infinitely many.

> *Example.* $A = (1\ 2\ 2\ 4)$ has rank 1 (row 2 is twice row 1), so $Ax = 0$ has a whole line of solutions, not just $x = 0$ — meaning $H\Delta x = -\nabla f$ would have infinitely many Newton steps to choose from if H looked like this, a sign that the model is under-determined in some direction.

Eigenvalues and eigenvectors. For a square matrix A, a nonzero vector v satisfying

$$Av = \lambda v$$

is an eigenvector with eigenvalue λ, found as roots of the characteristic polynomial $det(A - \lambda I) = 0$.

> *Example.* For $A = (3\ 1\ 1\ 3)$, solve $det(3 - \lambda\ 1\ 1\ 3 - \lambda) = (3 - \lambda)^2 - 1 = 0$, giving $\lambda = 4$ (eigenvector $(1, 1)$) and $\lambda = 2$ (eigenvector $(1, -1)$). Since both eigenvalues are positive, A is positive definite — if A were a Hessian, this critical point would be a genuine local minimum.

Quadratic forms. For symmetric A, the scalar $x^T Ax = \sum_{i,j} A_{ij} x_i x_j$ is a *quadratic form*. Its sign behavior as x ranges over R^n is fully controlled by the eigenvalues of A: writing x in the eigenbasis, $x^T Ax = \sum_i \lambda_i y_i^2$ where y are the coordinates of x in that basis. This immediately

shows why the *sign of the eigenvalues* — not the matrix entries directly — determines definiteness.

Positive definite / semi-definite matrices. $A \succ 0$ if $x^T Ax > 0$ for all $x \neq 0$ (equivalently, all eigenvalues $\lambda_i > 0$); $A \succcurlyeq 0$ if $x^T Ax \geq 0$ (all $\lambda_i \geq 0$). A quick practical test for 2×2 (or via leading principal minors, **Sylvester's criterion**, for general n): $A \succ 0 \Leftrightarrow A_{11} > 0$ and $det(A) > 0$.

> *Example.* $A = (2\ 1\ 1\ 5)$: $A_{11} = 2 > 0$ and $det(A) = 10 - 1 = 9 > 0$, so $A \succ 0$. Contrast with $A = (2\ 3\ 3\ 1)$: $det(A) = 2 - 9 = -7 < 0$, so A is *indefinite* — exactly the saddle-point signature from Aha #1.

Norms. $\| x\|_1 = \sum_i |x_i|$, $\| x\|_2 = \sqrt{\sum_i x_i^2}$, $\| x\|_\infty = max_i |x_i|$. These measure step sizes and convergence, and shape constraint sets.

> *Example — why the shape of the unit ball matters.* The set $\{x: \| x\|_2 \leq 1\}$ is a smooth disk; $\{x: \| x\|_1 \leq 1\}$ is a diamond with sharp corners on the axes; $\{x: \| x\|_\infty \leq 1\}$ is a square. When an ℓ_1-constrained (diamond-shaped) feasible region is intersected with a sloped linear objective's level sets, the optimum is pulled toward a **corner on an axis** — i.e., toward a solution where some coordinates are exactly zero. This geometric fact, not a coincidence, is *why* ℓ_1-regularized optimization (e.g. LASSO) tends to produce sparse solutions, while ℓ_2-regularization (a smooth disk, no corners) does not.

Orthogonality and projections. Vectors u, v are orthogonal if $u^T v = 0$. The projection of b onto the column space of A solves the **least-squares normal equations**

$$A^T A\, x = A^T b,$$

which is exactly the closed-form solution mentioned for least-squares problems in Part C.

Systems of linear equations & Gaussian elimination. Solving $Ax = b$ by row reduction is the computational engine inside the Simplex method (pivoting between vertices) and inside every Newton step (solving $H\Delta x = -\nabla f$ rather than explicitly inverting H, which is more numerically stable).

> *Example.* Solve $(2\ 1\ 1\ 3)\Delta x = (-4\ \ -5)$ (a Newton step). Row-reducing gives $\Delta x = (-1.4, -1.2)$ — this is precisely the kind of 2×2 (or, in practice, $n \times n$) solve repeated at every iteration of Newton's method.

3. Real Analysis

Sequences and limits. A sequence $\{x_k\}$ converges to x^* if for every $\varepsilon > 0$ there exists N such that $\| x_k - x^* \| < \varepsilon$ for all $k > N$. Every iterative optimization algorithm — gradient descent, Newton's method, Simplex — generates a sequence of iterates $x_0, x_1, x_2, \ldots$, and "does the algorithm work?" is precisely the question "does this sequence converge, and to what?"

Continuity. f is continuous at x_0 if for every $\varepsilon > 0$ there exists $\delta > 0$ such that $\| x - x_0 \| < \delta \Rightarrow \left| f(x) - f(x_0) \right| < \varepsilon$. Informally: small changes in input produce small changes in output — without this, gradient information at a point would say nothing reliable about nearby points, and no local algorithm could make progress.

Compactness. A set $S \subseteq R^n$ is compact if and only if it is **closed and bounded** (Heine–Borel theorem). Closed means it contains its own boundary; bounded means it fits inside some ball of finite radius.

Weierstrass Extreme Value Theorem. If f is continuous on a nonempty compact set S, then f attains both a maximum and a minimum on S. This is the theoretical guarantee that makes searching for an optimum a well-posed question in the first place.

> *Example — why compactness cannot be dropped.* Consider $f(x) = x$ on the **open** interval $S = (0, 1)$. Here f is continuous and S is bounded but *not closed* (it's missing its endpoints), so S is not compact. The infimum of f over S is 0, but no $x \in S$ actually achieves it — you can always pick a smaller x closer to 0 without ever reaching it. The "optimal solution" simply doesn't exist. This is exactly why optimization problem statements always specify a *closed* feasible region (e.g. $x \geq 0$ rather than $x > 0$): dropping closedness can silently make a problem unsolvable even when it looks perfectly reasonable.
>
> *A second example.* $f(x) = 1/x$ on $S = (0, \infty)$ is continuous but S is unbounded, so again S is not compact — and indeed f has no minimum on S (it approaches 0 but never reaches it, and blows up to $+\infty$ near $x = 0$).

Open vs. closed sets, boundary points. A point is a *boundary point* of S if every neighborhood around it contains points both inside and outside S. This matters directly in constrained optimization: an unconstrained interior optimum satisfies $\nabla f = 0$, but a constrained optimum sitting *on the boundary* of the feasible region need not have $\nabla f = 0$ at all — it only needs to have no feasible descent direction, which is exactly what the KKT conditions in Part C are built to detect.

> *Example.* Minimize $f(x) = x^2$ over $S = [1, 3]$. The unconstrained minimizer of f is $x = 0$, where $f'(0) = 0$ — but $0 \notin S$. Restricted to S, the minimum occurs at the boundary point $x = 1$, where $f'(1) = 2 \neq 0$. This is the simplest possible illustration of why constrained optimality conditions cannot just be "set the derivative to zero."

Fixed-point / contraction viewpoint (a bridge to convergence proofs). Many optimization algorithms can be viewed as generating $x_{k+1} = T(x_k)$ for some map T (e.g. a gradient step). The **Banach Fixed-Point Theorem** states that if T is a *contraction* — meaning $\| T(x) - T(y) \| \leq c \| x - y \|$ for some $c < 1$ — then the sequence $x_{k+1} = T(x_k)$ converges to a unique fixed point $x^* = T(x^*)$ from any starting point, geometrically fast. This is the abstract engine underneath convergence proofs for gradient descent on well-behaved (strongly convex, smooth) functions: showing a gradient step is a contraction is often literally how the proof goes.

4. Convexity

Convex sets. A set S is convex if for all $x, y \in S$ and $\lambda \in [0, 1]$,

$$\lambda x + (1 - \lambda) y \in S,$$

i.e. the entire line segment joining any two points of S stays inside S.

> *Examples.* A ball $\{x: \| x \| \leq r\}$, a half-space $\{x: a^T x \leq b\}$, and a hyperplane $\{x: a^T x = b\}$ are all convex. A polyhedron $\{x: Ax \leq b\}$ (an intersection of finitely many half-spaces) is convex too — and more generally, **the intersection of any collection of convex sets is convex** (if x, y lie in every set in the collection, the segment between them lies in every set, hence in the intersection). This single fact is why stacking multiple linear constraints together, as in an LP, still produces a convex feasible region no matter how many constraints you add.
>
> *A non-example.* The set $S = \{(x, y): xy \geq 1, x > 0\}$ (the region above one branch of a hyperbola) is **not** convex: take $x = (1, 1)$ and $y = (4, 0.25)$, both on the boundary curve $xy = 1$; their midpoint $(2.5, 0.625)$ has product $1.5625 \geq 1$ so it happens to lie inside here, but picking two points farther apart on the curve (e.g. $(0.1, 10)$ and $(10, 0.1)$) gives a midpoint $(5.05, 5.05)$ which is fine too — the curve $xy = 1$ is in fact convex on $x > 0$ as a *region above* it; the classic non-convex example instead is an annulus (ring) $\{x: 1 \leq \| x \| \leq 2\}$, where two points on opposite sides of the inner hole have a segment that cuts through the excluded center.

Convex functions. f is convex if for all x, y in its domain and $\lambda \in [0, 1]$,

$$f(\lambda x + (1 - \lambda) y) \leq \lambda f(x) + (1 - \lambda) f(y),$$

i.e. the chord between any two points on the graph lies on or above the graph. f is **strictly convex** if the inequality is strict for $x \neq y, \lambda \in (0, 1)$ — this rules out flat segments and guarantees a minimizer, if one exists, is unique.

First-order characterization. A differentiable f is convex if and only if its graph lies entirely above every tangent line:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad for\ all\ x, y.$$

This says the linear (first-order Taylor) approximation at any point *underestimates* f everywhere — a remarkably strong global statement built from purely local (derivative) information.

Second-order (Hessian) characterization. A twice-differentiable f is convex on a convex set if and only if $H(x) \succcurlyeq 0$ for every x in that set — exactly the positive-semidefiniteness condition from the Linear Algebra section, now reinterpreted as "non-negative curvature in every direction, everywhere."

Strong convexity. f is m-strongly convex ($m > 0$) if $f(x) - \frac{m}{2} \| x\|^2$ is convex, equivalently $H(x) \succcurlyeq mI$ for all x. This is a quantitative strengthening of convexity — it guarantees not just a unique minimizer, but a *curved-enough* bowl around it, which is exactly the condition needed to prove gradient descent converges at a fast (linear) rate, tying back to the contraction-mapping idea from Real Analysis.

> **Mathematical detail — Convexity collapses "local" and "global" into the same concept.** Suppose f is convex and x^* is a local minimum, i.e. there is a neighborhood N of x^* with $f(x^*) \leq f(x)$ for all $x \in N$. Suppose, for contradiction, some point z exists with $f(z) < f(x^*)$. Consider the segment $x(\lambda) = x^* + \lambda(z - x^*)$ for small $\lambda \in (0, 1]$; by convexity,
>
> $$f(x(\lambda)) \leq (1 - \lambda) f(x^*) + \lambda f(z) < f(x^*).$$
>
> For sufficiently small λ, $x(\lambda) \in N$, contradicting that x^* is a local minimum. Hence **no such z exists**, and x^* must be a *global* minimum. This is why so much of optimization theory tries to reformulate a problem as a convex one: convexity converts a global search into a local, "just walk downhill" problem, since there are no other local minima or saddle points to get trapped in.

Worked examples of checking convexity.

- $f(x) = x^2$: $f''(x) = 2 > 0$ everywhere, so f is (strictly) convex. Confirms the familiar bowl shape.
- $f(x, y) = x^2 + y^2$: $H = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \succ 0$, so f is strictly convex — a paraboloid bowl in every direction.
- $f(x, y) = x^2 - y^2$: $H = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$ is indefinite (one positive, one negative eigenvalue), so f is **neither convex nor concave** — this is the saddle function from Aha #1, viewed now through the convexity lens.
- $f(x) = \| x \|$ (any norm) is convex, by the triangle inequality: $\| \lambda x + (1 - \lambda) y \| \leq \lambda \| x \| + (1 - \lambda) \| y \|$ directly matches the convexity definition — no calculus required, and it holds even though $\| x \|$ isn't differentiable at 0 (convexity doesn't require differentiability).
- $f(x) = log(e^{x_1} + \cdots + e^{x_n})$ (**log-sum-exp**) is convex — a common building block in machine-learning objectives — even though this is not obvious from the definition alone; it is usually shown via the Hessian.

Operations that preserve convexity (useful for quickly recognizing convex problems without redoing a Hessian check each time):

- **Nonnegative weighted sums**: if $f_1, \ldots, f_k$ are convex and $w_i \geq 0$, then $\sum_i w_i f_i$ is convex.
- **Pointwise maximum**: if $f_1, \ldots, f_k$ are convex, then $g(x) = max_i f_i(x)$ is convex (even though each "piece" may be linear, the max of several tangent planes is a convex, piecewise-linear surface — this is exactly how many robust and minimax formulations stay convex).
- **Composition with an affine map**: if f is convex, then $g(x) = f(Ax + b)$ is convex.

Brief Mentions

- **Newton's method can converge to a maximum or saddle**, not just a minimum, because it only seeks $\nabla f = 0$.
- **Gradient descent zig-zags on elongated, ill-conditioned bowls** — even for simple convex quadratics.
- **Simplex has exponential worst-case complexity yet is extremely fast in practice** — a famous gap between worst-case theory and empirical performance.
- **Duality: every optimization problem has a "mirror" problem**, and under convexity the two meet at the same optimal value.
- **Adding constraints can never improve the optimal value** for a minimization problem — only preserve or worsen it, since the feasible region can only shrink.
- **No free lunch**: no single algorithm dominates across all problem classes; a method fast on smooth convex problems can fail entirely on non-convex or discontinuous ones.