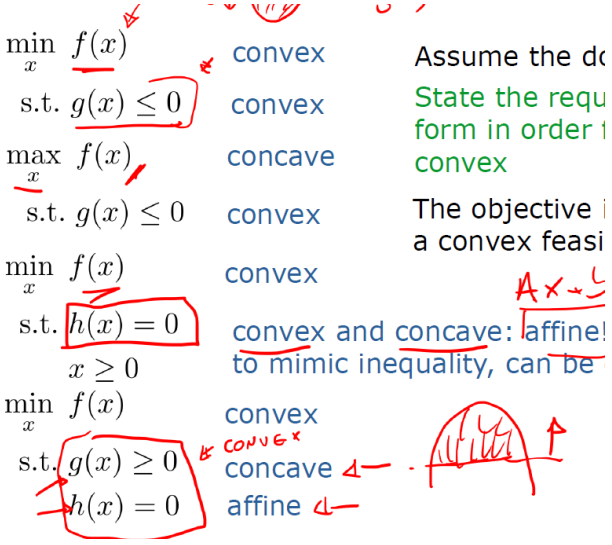
CVX

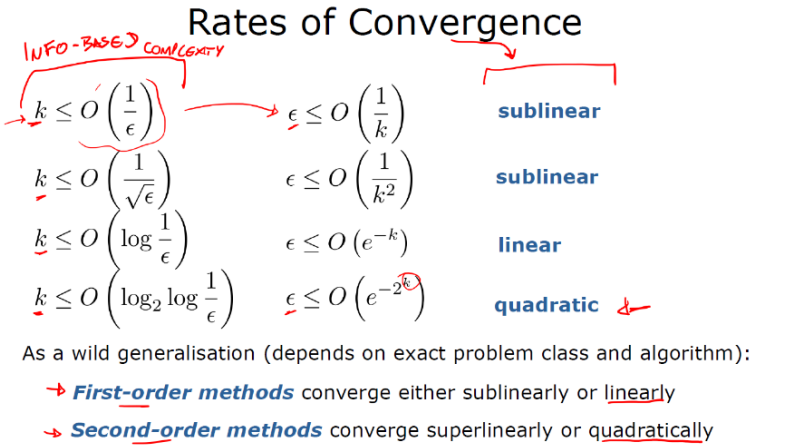
Some problems where the region doesn't have to be a polyhedron, and the optimal solution doesn't have to be on an extreme point, simplex won't work.



A function f is convex if, for any two points x and y, the line segment between these points lies above the function graph. Mathematically:

$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta)f(y),$$

where $\theta \in [0,1]$. This definition is central for proving that the solution is globally optimal in convex problems.



Information-Based Complexity measures the number of iterations required to achieve a solution within a specified tolerance. This is a practical metric for understanding how efficiently an algorithm can approximate the optimal value.

The **Jacobian matrix** generalizes the gradient for vector-valued functions, containing partial derivatives of each output component with respect to each input variable. Every Row is the transpose of the derivative of that function f1 with respect to xi.

The **Hessian matrix** provides a second-order derivative for scalar functions, showing curvature. Convex iff function lies on or above all of its tangents, explained below. Single variate – only second derivative, multi variate – strictly convex, Hessian matrix is **positive definite** everywhere. Definiteness of Hessian at a stationary point indicates whether or not it is a minimum (pos definite), maximum (neg definite) or saddle point (indefinite).

Local Minimum Condition: In convex functions, if x* is a local minimum, then it is also a global minimum. If the function is convex, first order condition becomes a **sufficient** condition for global minimum, if not then hessian being positive semidefinite is necessary.

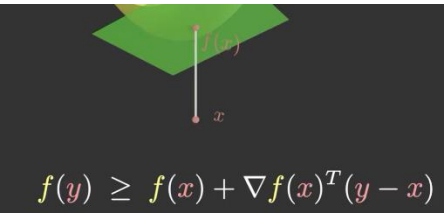
In multi-variable constraints, visualizing sublevel (f(x)≤c) and super-level (g(x)≥c) sets helps in understanding the feasible region. For convex functions, sublevel sets are convex, which supports the optimization process by maintaining a convex feasible region.

Equality and inequality constraints together define what is known as the feasible set. Latter divide the region as represented by a hyperplane, with a positive/negative half space or null region i.e. the hyperplane itself. Large family of optimization problems, mainly convex optimization problems can be solved efficiently in a unified manner.

Polyhedral, which is intersection of half spaces, is a convex set. Region above the graph is called epigraph, and epi of a convex function is convex (hypo for concave). Scalar multi or addition are convex. An optimization problem is convex if the objective function and constraints are convex or linear. (obj being lin and constraint being convex counts). If f is convex but g is concave, problem can still be convex.

Principle of duality – Take a hyperplane that supports my set in such a way that the set falls entirely on the positive side of this hyperplane. That means a set is convex when you recover the set in intersection of all the positive sides of all the hyperplanes.

Extra - While a twice-differentiable function with a positive definite Hessian everywhere is strictly convex (e) and will have at most one stationary point (a), which is a global minimum (b), it does not guarantee that Newton's method will converge in one step. Newton's method converges in one step only if the function is quadratic. For non-quadratic functions, it typically requires multiple iterations to converge.



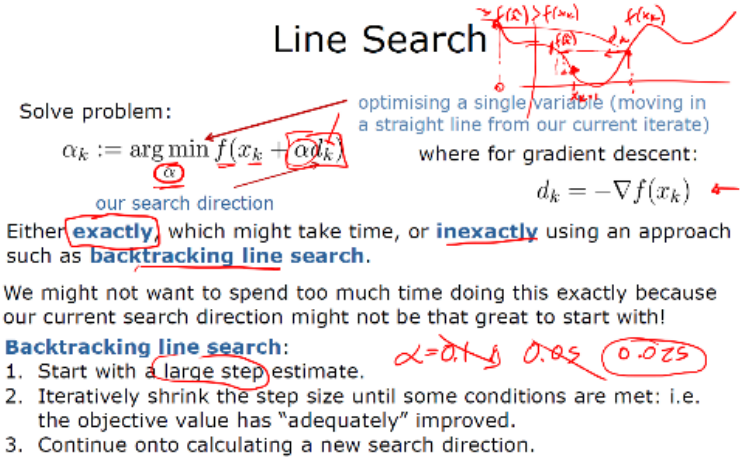
Tangent Hyperplane is a good approximation of the graph of f at point x. Dual def of convex fn, f is convex if and only if it's graph is above its tangent hyperplane. So, for gradient 0, the hyperplane is horizontal – so global minimizer. So gradient 0 and solve for x. So non constant linear functions are unbounded. There can be more complex problems, to even solve the gradient, so there are other methods.

Gradient descent – we can evaluate the cost function at an arbitrary point, and with a process called back propagation, we evaluate the negative gradient and take a small step in that direction, doing it iteratively (in stochastic, we can take small random subset at each iteration, in adaptive, picks a different step size for each component).

$$\theta^{k+1} = \theta^k - \eta \nabla f(\theta^k)$$

When you minimize a function, what you are looking for critical points of that function, where derivate equals zero.

How step size is selected is called stepped, like fixed or rule based(might use info from previous iterations). In line based, updates based on gathering information by looking at previously un-explored points.



Newton's method-Hard to plot functions that have more than 2 variables(maybe even million), iterative optimisation algo, start with intial guess x0(unknown, draw from some random distraubution), and the closer the intial guess is to the true minimiser the better. We pick a direction d0(descent direction) and we follow it to get x1, track f(x1) for progress in decline. How to pick good descent direction – gradient gives us direction of biggest increase. $X_{k+1} = x_k + d_k$.

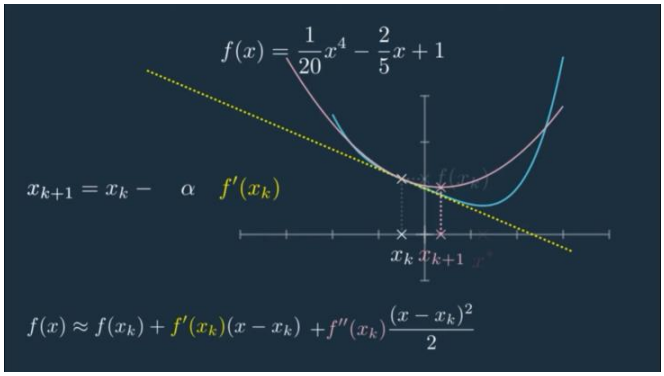
Newton's method is commonly used for **finding the zeros** of non-linear functions. Start with Taylor series of a **multivariate scalar function.** At some iteration x_k, when x tends towards x_k, the rate of change of the function x around the point x_k.

$$f'(x_k) \approx \frac{f(x) - f(x_k)}{x - x_k}$$

Rearranging terms, we see that the derivative is the slope of the tangent line to the graph of f at x_k. If the slope is positive, go as much as left in the tangent line, vice versa for negative. Both cases, $X_{k+1} = x_k$ - alpha(step size) * $f'(x_k)$, subtracting a positive multiple of the derivative slope from the current iterate. We can't make alpha too large, we might overshoot and miss the minimizer, too small and it might take a lot more iterations, trying to tune it in a optimal way is what leads us to newton's method. From taylor series, we can get an even better approximation by including the second derivative(which is quadratic), and take the minimizer of this quadratic function as our next iterate. So alpha is now inverse of second derivative by algebra, and we don't have to pick alpha manually anymore, making it second order method from using second derivative information. For multivariate, replace the first derivative with the gradient and the second derivative with the inverse of the hessian. A hessian is only invertable if it is positive definite, i.e. all it's eigen values are positive HV = lamda v.

$$f(x) \approx f(x_k) + f'(x_k)(x - x_k)$$

3. **When the Function Is Exactly Quadratic:**
   - If $f(x)$ is quadratic, such as $f(x) = \frac{1}{2}x^T Q x + c^T x$, where Q is **positive definite**, then Newton's Method will converge to the minimum in **one step**.
   - In this case:
     - The gradient $\nabla f(x) = Qx + c$.
     - The Hessian $H_f(x) = Q$ (a constant matrix).
4. **Direct Solution for x When $f(x)$ Is Quadratic:**
   - Applying Newton's formula:
     $$H_f(x_k)^{-1} \nabla f(x_k) = Q^{-1}(Qx_k + c) = x_k + Q^{-1}c$$
     - Solving this directly moves x_{k+1} to the exact minimum point:
       $$x_{k+1} = -Q^{-1}c$$
     - Therefore, if $f(x)$ is truly quadratic, **Newton's Method reaches the minimum in a single step.**



Newton's Method **approximates** the function locally as quadratic to get the direction and step size. However, this assumption may not always be accurate, which is why techniques like **line search** are used to adjust the step size as needed.

For non differentable functions like mod x, it can be dealt with a change in formulation when we allow constraints.

Advantage – Quadratic convergence, which means at every iteration we double the number of exact digits in our approximation. Disadvantage – Scalablity, n^2 number of memory units for n variables for hessian, and for inverse near cube of that.

In order to have a **unique** solution for the Newton step, we need the Hessian to be invertible:

$$x_{k+1} - x_k = -H_f(x_k)^{-1}\nabla f(x_k)$$

Penalty – For converting constrained function to an unconstrained one by adding the constraint to the obj with a penalty that penalises for not satisfying that constraint. i.e if y is the constraint, then if y<=0, then penalty is 0 for a minimising problem, otherwise infinity. The worst-case penalty would make obj infinity, and it would be impossible to solve. Approximating the penalty function with another function that is smooth and does not take infinite values like P(y) = u * y, where u slope is nonnegative. So now penalty will be proportional to how much you violated the constraint. But we are rewarding the obj when P(y)<0, which we don't want, because it would affect the optimal value. If we consider all the slopes of this linear function, and take the max instead, we recover the original zero infinity function. After rearranging max and min, we do the minimisation of the obj function by setting the gradient to 0 and solve for x. Maximizing with respect to u(making it the dual problem)

$$\max_{u \geq 0} \min_{x} \; x_1 + x_2 + u\,(x_1^2 + x_2^2 - 1)$$

Infimum of a set: largest lower bound on values in set, doesn't have to be part of the set. **Supremum** of a set: smallest upper bound on values in set

Dual Optimisation problem – Constrained to unconstrained, multiplying with scalar slopes, taking the max(we want the inequality scalar to be positive and the equality scalar to be either positive or negative, since we want to penalise both). This is the lagrangian of the convex problem. This dual problem gives us a lower bound on the optimal value of the primal problem (because we are minimising first), which is the weak duality theorem. For strong duality, slater's condition. Lagrangian Relaxation is like duality in LP, where we relax constraints to create a problem that is easier to solve. The dual function inf L(x, v, u) = d(v,u) is always concave. The **dual problem** maximizes d(v, u) subject to u>=0

$$\max_{v_i, u_j \geq 0} \min_{x \in \mathbb{R}^n} f(x) + \sum_i v_i\, h_i(x) + \sum_j u_j\, g_j(x)$$

1. **Purpose**: Slater's Condition helps ensure **strong duality** in convex optimization problems with inequality constraints, meaning that the optimal values of the primal and dual problems are equal.
2. **Condition**: Slater's Condition states that for a convex optimization problem:
   - If all inequality constraints $g_i(x) \leq 0$ are **convex and nonlinear** (they aren't just affine/linear constraints), then **strong duality holds** if there exists a point x^* strictly within the feasible region.
   - Specifically, this point x^* must satisfy:

     $$g_i(x^*) < 0 \quad \text{for all } i, \quad \text{and} \quad h_j(x^*) = 0 \quad \text{for all equality constraints } h_j(x).$$
3. **Implication**:
   - If such a point x^* exists (one that strictly satisfies the inequality constraints), then **strong duality** is guaranteed.
   - In other words, the optimal values of the primal and dual problems will be equal, and duality theory can be applied effectively.

So, **finding an x^* that satisfies all constraints strictly does not automatically mean strong duality holds**, but if such an x^* exists and the problem is convex, Slater's Condition guarantees strong duality. This is a sufficient condition (not necessary)

Dual Gradient Ascent – Dual Gradient Ascent is a method for solving the dual problem by iteratively adjusting the dual variable u to maximize the dual function. The dual gradient ascent method iterates between solving the inner level problem for the primal variables (provided the current iterate for the dual variables), and performing a gradient ascent step on the dual variables.

$$x_{k+1} := \arg\min_x \mathcal{L}(x, \mu_k)$$
$$\mu_{k+1} := \mu_k + \alpha_k g(x_{k+1})$$

KKT Conditions – The **KKT Conditions** represent a set of first-order necessary conditions for finding local optima in constrained problems, applicable when certain regularity conditions are met, like slater's. The gradient wrt x needs to be zero. Complementary slackness ensures that if a constraint $g(x)$ is inactive (i.e., it does not bind the solution), its corresponding Lagrange multiplier must be zero.If only one constraint, then the derivate of f and g are inversely proportional i.e. d(f) = -ud(g). So, x should be feasible, so g(x)>=0, u>=0, penalty terms ug(x) = 0, so it doesn't affect the optimal value. Useful for designing general purpose optimisation solvers.

x^* to be a local minimum, there exist some KKT multipliers μ λ

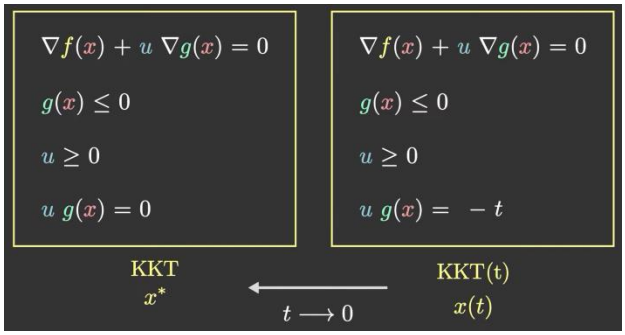
$$\nabla_x \mathcal{L}(x^*, \mu, \lambda) = 0 \quad \text{(stationarity)}$$
$$g(x^*) \leq 0 \quad h(x^*) = 0 \quad \text{(primal feasibility)}$$
$$\mu \geq 0 \quad \text{(dual feasibility)}$$
$$\mu_j g_j(x^*) = 0 \quad \text{(complementary slackness)}$$

Interior Point Method – Way to solve KKT conditions. Interior Point Methods are optimization techniques used to solve linear and nonlinear constrained problems. They work by keeping the solution iteratively "inside" the feasible region and gradually moving towards the boundary, as opposed to boundary methods like Simplex. Interior Point Methods avoid directly hitting constraint boundaries by using a "barrier" approach, which penalizes the solution as it nears the boundary, making it computationally efficient.

We are going to solve a modified perturbed version called KKT(t) (which are much easier to solve. where t is +ve parameter which controls the degree of perturbation. As this parameter goes to 0, x(t) converges to x*. u = -t/gx. Plugging into the gradient equation, we get -t log, which acts as a barrier function at the objective, approaching infinity as the variables approach the boundary of a constraint, discouraging any further movement in that direction, smoothing out the sharp transition of a constraint. The **barrier method** (interior point method) solves a sequence of barrier problems with t converging to zero.

$$\nabla(f(x) - t \log(-g(x))) = 0$$

We can now perform newton's method to perform this minimisation. Initially, don't t to be too small otherwise it will result in a discontinuous function with the log, and newton would be too slow to converge, so start with a big t, meaning f would be overshadowed, and only g would need to be minimised. X(t) would be the analytic center of the feasible region. Once we have a solution, try for a smaller t, i.e. t' = 0.9t, apply newton again, but x(t) would be the starting point, so computing x(t') would be faster. Working towards t = 0 formed by the path of xs of t, called the central path (trajectory that solutions follow in the interior point method as they approach the optimal solution), falling in the interior of the feasible region.



Second order methods can often achieve higher rates of convergence over first order methods. However, this comes at the cost of more computation per iteration, e.g., in calculating Hessians and solving a large linear system of equations. First order methods might be necessary for applications where the problem is very large, where the Hessian may struggle to fit into memory.

To determine if the function $f(x, y) = x^2 + xy - 2$ is convex, we use the **Hessian matrix** approach. We start by calculating the first partial derivatives of f:

$$f_x = 2x + y \quad \text{and} \quad f_y = x$$

Next, we compute the second partial derivatives to form the Hessian matrix H:

$$H = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}$$

To assess the definiteness of H, we find its eigenvalues by solving the characteristic polynomial:

$$\det(H - \lambda I) = -\lambda^2 + 2\lambda - 1 = 0$$

Using μ_1, μ_2, μ_3 for the inequality constraints above respectively. The stationarity condition consists of the following two equations:

$$-\mu_1 \frac{1}{x^2} + 4\mu_2 x - \mu_3 = 0$$
$$2(y + 2) - \mu_1 = 0$$

The constraints are:

$$\frac{1}{1.5811} - 0.6325 = 0.000$$
$$2 \cdot 1.5811^2 - 5 = 0.000$$
$$1 - 1.5811 = -0.5811$$

All constraints are satisfied (**primal feasibility**). The last constraint is not active, so due to the complementary slackness condition, its dual variable must be zero $\mu_3 = 0$ (**complementary slackness**). Plugging in the candidate point to the complementary slackness terms we get the following expressions for the other two duals:

$$\mu_2 = 2(y + 2)\frac{1}{4x^3} = 0.333$$
$$\mu_1 = 2(y + 2) = 5.265$$

All duals are positive (**dual feasibility**) and the stationarity condition is satisfied with these dual values (**stationarity**).

The problem is convex and all functions are smooth over the domain of interest, so the KKT conditions are sufficient for a global optimal solution.

For a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$, state a condition that f needs to satisfy in order to be a convex function if f is not differentiable. State a different condition for the case where f *is* differentiable.

For the first part you could use: the line segments between any two points on graph of f, must be above or equal to f. A second option would be to state that the epigraph of f must be a convex set.

When f is differentiable, different conditions from the above are: the Hessian of f must be positive semidefinite, or the graph of f must lie on or above all its tangent planes.

$$x^{(k)} = \arg\min_x f(x) + \eta_k \| \max(0, g(x)) \|_2^2 + \eta_k ||h(x)||_2^2$$

$$\min_{x \in \mathbb{R}^n} f(x)$$
$$\text{s.t. } g(x) \le 0$$

The Penalty Method turns a constrained optimisation problem into an unconstrained one, for a penalty parameter η_k, by applying a quadratic penalty to constraint violations:

$$\min_{x \in \mathbb{R}^n} f(x) + \eta_k \sum_i^m \max(0, g_i(x))^2$$

A sequence of the above penalty problems are solved for increasing η_k, until the constraint violations are within some tolerance.