Last Time:

- Newton's Method
- Root Finding
- Minimization

Today:

- Constrained Minimization

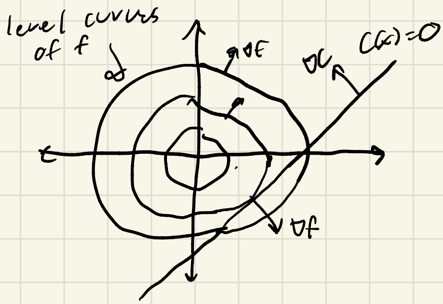
Equality Constraints:

$$\min_x \quad f(x) \qquad \leftarrow f(x): \mathbb{R}^n \to \mathbb{R}$$

$$\text{s.t.} \quad c(x) = 0 \qquad \leftarrow c(x): \mathbb{R}^n \to \mathbb{R}^m$$

* First-Order Necessary Conditions

1) Need $\nabla f(x) = 0$ in <u>free</u> directions
2) Need $c(x) = 0$



$f(x): \mathbb{R}^2 \to \mathbb{R}$

$c(x): \mathbb{R}^2 \to \mathbb{R}$

∇c = normal vector to $c(x)=0$

* Any non-zero component of ∇f must be normal to the constraint surface/manifold

$$\Rightarrow \nabla f + \lambda \nabla c = 0 \quad \text{for some } \lambda \in \mathbb{R}$$

λ: "Lagrange multiplier"/"Dual variable"

- In general:

$$\frac{\partial f}{\partial x} + \lambda^T \frac{\partial c}{\partial x} = 0$$

- Based on this gradient condition, we difine:

$$\underbrace{L(x,\lambda)}_{\text{"Lagrangian"}} = f(x) + \lambda^T c(x)$$

Such that:

$$\left.\begin{aligned} \nabla_x L(x,\lambda) &= \nabla f + \left(\frac{\partial c}{\partial x}\right)^T \lambda = 0 \\ \nabla_\lambda L(x,\lambda) &= c(x) \end{aligned}\right\} \text{"KKT conditions"}$$

- We can solve this jointly in x and λ with Newton

$$\nabla_x L(x+\Delta x, \lambda+\Delta\lambda) \approx \nabla_x L(x,\lambda) + \frac{\partial^2 L}{\partial x^2}\Delta x + \underbrace{\frac{\partial^2 L}{\partial x \partial \lambda}}_{\left(\frac{\partial c}{\partial x}\right)^T} \Delta\lambda$$

$$\nabla_\lambda L(x+\Delta x, \lambda) \approx c(x) + \frac{\partial c}{\partial x}\Delta x$$

$$\underbrace{\begin{bmatrix} \frac{\partial^2 L}{\partial x^2} & \left(\frac{\partial c}{\partial x}\right)^T \\ \frac{\partial c}{\partial x} & 0 \end{bmatrix}}_{\text{"KKT System"}} \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} -\nabla_x L(x,\lambda) \\ -c(x) \end{bmatrix}$$

* Gauss-Newton Method:

$$\frac{\partial^2 L}{\partial x^2} = \nabla^2 f + \underbrace{\frac{\partial}{\partial x}\left[\left(\frac{\partial c}{\partial x}\right)^T \lambda\right]}$$

this term is expensive to compute

- We often drop the 2nd term ("constraint curvature")
- This is called "Gauss-Newton"
- Equivalent to linearizing first, then doing Newton

* Example
  - Start at $[0, 0]$, $[-3, 2]$

    Newton gets stuck, Gauss-Newton doesn't

* Take Away Message:
  - May need to regularize $\frac{\partial^2 L}{\partial x^2}$ in Newton, even $\nabla^2 f \geq 0$
  - Gauss-Newton is often used in practice because it converges almost as fast and is cheaper per iteration

---

* Inequality Constraints:

$$\min_x \quad f(x)$$
$$\text{s.t.} \quad c(x) \geq 0$$

- We'll just look at the case of only inequalities for now
- In general, these methods are combined with the previous ones to handle both equality and inequality constraints in the same problem

* First-Order Necessary Conditions:

1) $\nabla f = 0$ in the <u>free</u> directions
2) $c(x) \geq 0$

KKT conditions
$$\begin{cases} \nabla f - \left(\frac{\partial c}{\partial x}\right)^T \lambda = 0 & \leftarrow \text{"Stationarity"} \\ c(x) \geq 0 & \leftarrow \text{"primal feasibility"} \\ \lambda \geq 0 & \leftarrow \text{"dual feasibility"} \\ \lambda^T c(x) = 0 & \leftarrow \text{"Complementarity"} \end{cases}$$

- Intuition
  - If constraint is "active" $(c(x) = 0) \Rightarrow \underbrace{\lambda \geq 0}_{\text{same as equality case}}$
  - If constraint is "inactive" $(c(x) > 0) \Rightarrow \underbrace{\lambda = 0}_{\text{same as unconstrained case}}$
  - Complementarity encodes "on/off switching" of constraints

* Algorithms:

- Much harder than equality case
- Can't directly apply Newton to KKT conditions
- Many options w/ trade offs

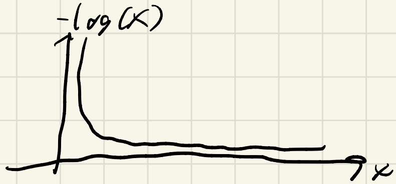
* Active-Set:

  - Have some way of knowing active/inactive constraints
  - Solve equality constrained problem
  - Very fast if you have a good heuristic
  - Can be bad otherwise

* Barrier/Interior-Point

  - Replace inequalities with a "barrier function" in objective that blows up at constraint boundary

$$\left.\begin{aligned}&\min_x \ f(x)\\&\text{s.t. } c(x)\geq 0\end{aligned}\right\} \rightarrow \min_x \ f(x) - \sum_{i=1}^{m} \frac{1}{\rho}\log(c_i(x))$$

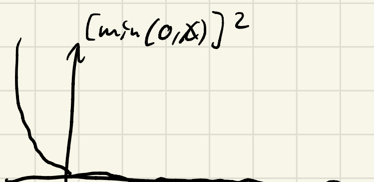


  - Gold standard for small-medium convex problems
  - Requires lots of hacks/tricks for non-convex problems

* Penalty

  - Replace inequality with an objective term that penalizes violations:

$$\left.\begin{aligned}&\min_x \ f(x)\\&\text{s.t. } c(x)\geq 0\end{aligned}\right\} \rightarrow \min_x \ f(x) + \frac{\rho}{2}\left[\min(0,c(x))\right]^2$$



- Easy to implement
- Has issues with ill-conditioning
- Difficult to achieve high accuracy

* Augmented Lagrangian

- Add a Lagrange multiplier estimate to penalty method:

$$\min_x \underbrace{f(x) - \tilde{\lambda}^T c(x) + \frac{\rho}{2}[\min(0, c(x))]^2}_{L_\rho(x,\lambda) \text{ "Augmented Lagrangian"}}$$

- Update $\tilde{\lambda}$ by "offloading" penalty term at each iteration:

$$\frac{\partial f}{\partial x} - \tilde{\lambda}^T \frac{\partial c}{\partial x} + \rho c(x)^T \frac{\partial c}{\partial x} = \frac{\partial f}{\partial x} - \underbrace{[\tilde{\lambda} - \rho c(x)]}^T \frac{\partial c}{\partial x} = 0$$

$$\Rightarrow \tilde{\lambda} \leftarrow \tilde{\lambda} - \rho c(x)$$

(for active constraints)

- Repeat until convergence:

1) $\min_x L_\rho(x, \tilde{\lambda})$

2) $\tilde{\lambda} \leftarrow \max(0, \tilde{\lambda} - \rho c(x))$ ← Clamp to guarantee non-negativity

3) $\rho \leftarrow \alpha \rho$

Typically $\alpha \approx 10$

- Fixes ill-conditioning of penalty method
- Converges fast (super linear) to moderate precision
- Works well on non-convex problems

* Example: Quadratic Program

$$\min_x \ \frac{1}{2} x^T Q x + q^T x \quad , \quad Q > 0 \quad \text{(convex)}$$

$$\text{s.t.} \quad Ax \leq b$$
$$Cx = d$$

- Very useful in control
- Can be solved very fast online (e.g ~ kHz)

* Some Notes:

- In general, you still need regularization and line searches in the constrained setting

- Line searches get a little more complicated

* Reference: Numerical Optimization

Nocedal + Wright