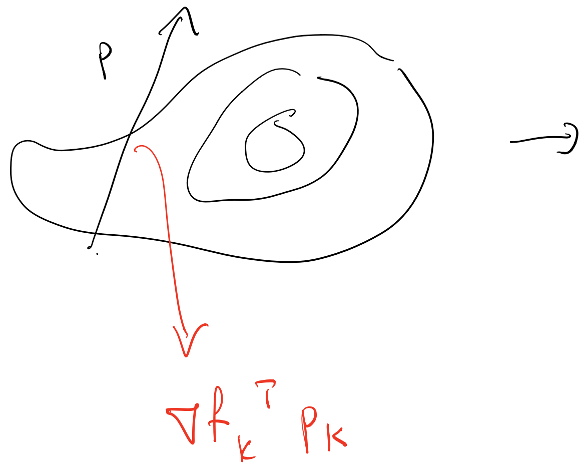
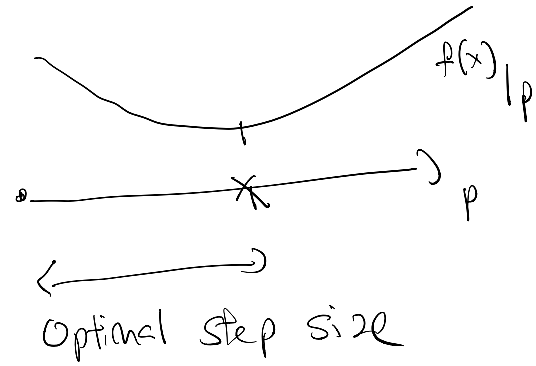
1. Choosing the step size
   - theory for quadratic surfaces
   - $\boxed{\text{wolf conditions}}$
2. 1st order methods
   - Batch, mini-batch & stochastic GD
   - Momentum, RMSProp, ADAM
   - $\boxed{\text{Gradient descent for least squares}}$
3. 2nd order methods
   - Quasi-Newton
   - $\boxed{\text{Levenberg-Marquardt (Trust region)}}$
   - Minimizing Negative log likelihoods
4. Constrained optimization
   - Transformation
   - Penalties
   - $\boxed{\text{Lagrange multipliers}}$
   - Karush Kuhn Tucker (KKT)
5. Code examples

Choosing step size in line search



In direction of p, this is a 1D optimization problem



① 1D minimization

→ Golden section

→ Quadratic interpolation

→ Newton

Generally too expensive

Line search

$$x_{k+1} = x_k + \underbrace{\alpha}_{\text{step size}} \underbrace{p_k}_{\text{search direction}}$$

Want $\underbrace{\nabla f_k^T p_k}_{\text{directional derivative}} \underbrace{< 0}_{\text{descent}}$

Search directions are typically $p_k = -B^{-1} \nabla f_k$

$B = I$ steepest descent

$B = H$ Newton method

$B =$ Iteratively updated approximation of H

Quasi-Newton

$$\nabla f_k^T p_k = -\nabla f_k^T B^{-1} \nabla f_k$$

< 0 if B is positive definite

Step size

(1) Find a bracket & generate candidate α_k within bracket

Check candidates against termination conditions

(2) Termination conditions $\quad 0 < c_1 < c_2 < 1$

Wolfe conditions

(1) $f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T p_k$

(2) $\nabla f(x_k + \alpha p_k)^T p_k \geq c_2 \nabla f_k^T p_k$

Let $\varphi(\alpha) = f(x_k + \alpha p_k)$

$\varphi'(\alpha) = \nabla f(x_k + \alpha p_k)^T p_k$

$\varphi'(0) = \nabla f_k^T p_k$

$\varphi'(\alpha) \geq c_2 \varphi'(0)$

From Nocedal & Wright (Numerical optimization)

Don't go too far for too little

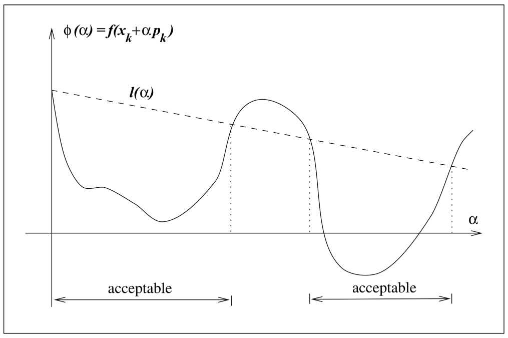


Figure 3.3 Sufficient decrease condition.

Don't stop when you're winning

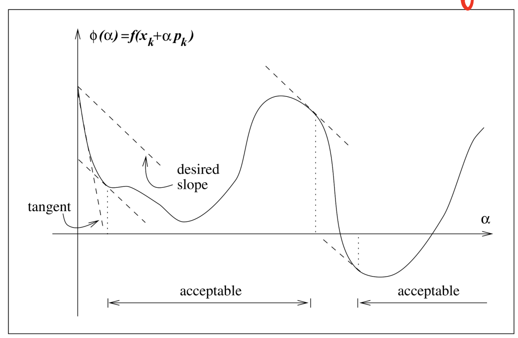


Figure 3.4 The curvature condition.

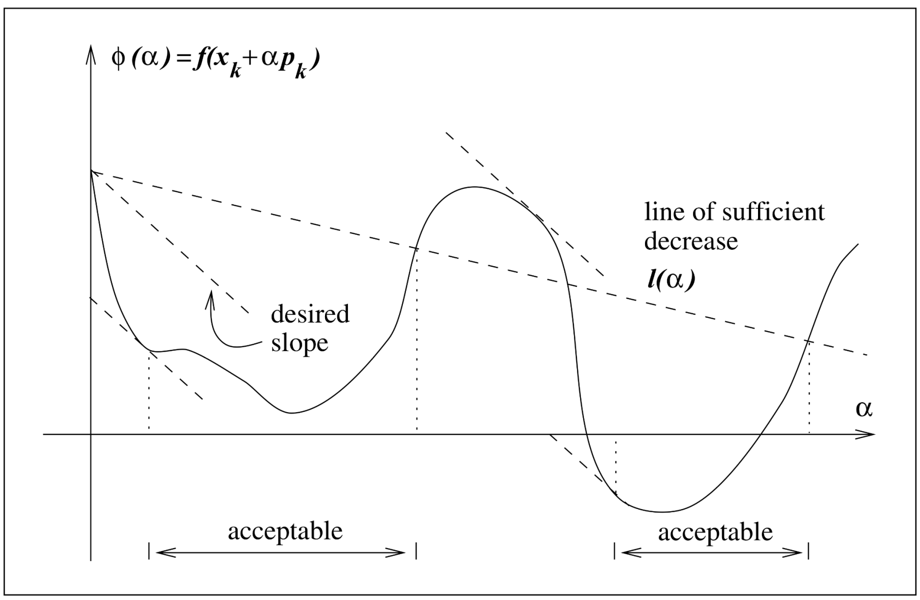


Figure 3.5 Step lengths satisfying the Wolfe conditions.

Gradient descent for least squares

$$\text{Min} \quad L(x) = \|y - Ax\|^2$$

$$= (y-Ax)^T(y-Ax)$$

$$= (y^T - x^T A^T)(y - Ax)$$

$$= y^T y - y^T A x - x^T A^T y + x^T A^T A x$$

$$= y^T y - x^T A^T y - x^T A^T y + x^T A^T A x$$

$$\frac{\partial L}{\partial x^T} = -2A^T y + 2A^T A x$$

$$= \underbrace{A^T A x - A^T y}_{\text{gradient}}$$

$$\boxed{\begin{array}{c} x^T(Bx) \\ \shortparallel \\ (Bx)^T x \\ \shortparallel \\ x^T B^T x \\ \shortparallel \\ x^T B x \end{array}}$$

$$\underbrace{x_{k+1} = x_k + \alpha\left(A^T y - A^T A x_k\right)}_{\text{gradient descent for least squares.}}$$

Levenberg-Marquardt

$$x_{k+1} = x_k - \alpha \nabla f_k^T p_k \qquad \text{Gradient descent}$$

$$x_{k+1} = x_k - \alpha B^{-1} \nabla f_k^T p_k \qquad \text{Newton}$$

$$x_{k+1} = x_k - (B + \lambda I)^{-1} \nabla f_k^T p_k \qquad \text{LM}$$

$\lambda \to 0 \Rightarrow$ Newton

$\lambda \to \infty \Rightarrow$ Gradient descent

Note $\lambda \to \infty$ guarantees $(B + \lambda I)^{-1}$ is positive definite

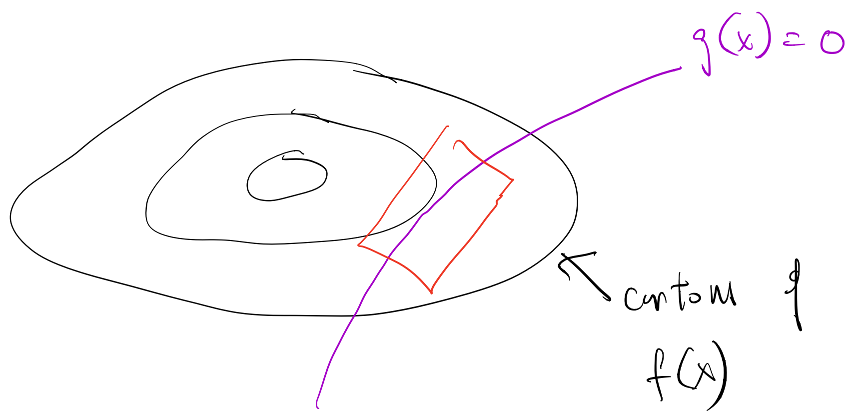
$\boxed{\lambda \text{ defines the "trust" region size}}$

Used for nonlinear least squares

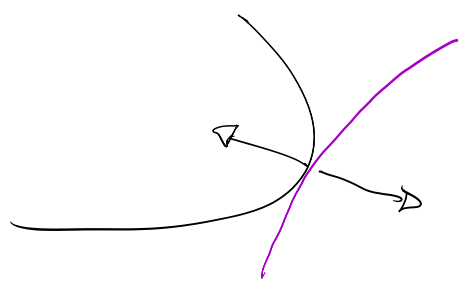
"curve_fit" in scipy.optimize

Constrained optimization

Minimize $f(x)$ with constraint $g(x) = 0$



On contour line, $f(x)$ does not change



g must be PARALLEL to f at minimum point.

otherwise could $\downarrow f$ by moving a small distance along g.

if f and g are tangent at minimum

∇f and ∇g must be parallel

$$\Rightarrow \quad \nabla f = \lambda \nabla g$$

$f(x) = xy + yz$ Objective function

$g(x) = x + 2y - 6$
$h(x) = x - 3z$ } constraints

Set up Lagrangian

$$F = xy + yz - \lambda(x + 2y - 6) - \mu(x - 3z)$$

$\nabla F = 0$

$$\left.\begin{aligned} y - \lambda - \mu &= 0 \\ x + z - 2\lambda &= 0 \\ y + 3\mu &= 0 \\ x + 2y - 6 &= 0 \\ x - 3z &= 0 \end{aligned}\right\}$$

5 equations in 5 unknowns

Solve using linear algebra

Review examples in notebooks