Lecture 11.
Math 685

Nonlinear optimization.

1. **Steepest descent:**

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

$$\alpha_k = \min_{\alpha} f(x_k - \alpha \nabla f(x_k))$$ – line search

Linear convergence: $e_{k+1} \sim C \cdot e_k$, $C < 1$

$e_k = \|x_k - x^*\|$

C can be very close to 1.

An automatic way to compute ∇f: $\dfrac{f(x_k + h) - f(x_k)}{h}$

forward difference

central difference: $\dfrac{f(x_k+h) - f(x_k-h)}{2h}$

2. **Newton's method.**

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

$g(x) = \nabla f(x)$ then $x_{k+1} = x_k - \dfrac{g(x_k)}{g'(x_k)}$

root finding problem

Implementation: $x_{k+1} = x_k + s_k$

where $H_f(x_k) s_k = -\nabla f(x_k)$

s_k solves this linear system.

⊕ quadratic convergence

$$e_{k+1} \sim \underset{\substack{\uparrow \\ \text{conv. factor} \\ C<1}}{C} \cdot e_k^2 \qquad r=2 \text{ rate of convergence}$$

⊖ very sensitive to initial guess.

Needed: descent condition

$$\nabla f(x_*)^T s_k < 0 \qquad x_* \text{ - exact sol.}$$

3. <u>quasi-Newton</u>

$$B_k^{-1} \approx H_f^{-1}(x_k)$$

Hessian approximation (much less cost per iteration)

more robust than Newton

But descent direction can be lost.

less fast (at most superlinear convergence), similar to secant method vs. Newton for rootfinding.

4. <u>CG</u> $f(x_k) - f(x^*) \leq \left(\frac{\sqrt{cond}-1}{\sqrt{cond}+1}\right)^2 (f(x_{k-1}) - f(x^*))$

doesn't store B_k approximation, but computes it implicitely.

Solves quadratic problems exactly in $\leq n$ steps where n = dimension of the problem.

(Similar to BFGS with exact line search).

5. Nonlinear least squares

$$\varphi(x) = \frac{1}{2}\vec{r}^T\vec{r} \to \min$$

This puts the problem into the context of nonlin. optim.

Gauss - Newton:

$$J^T(x_k)J(x_k)s_k = -J^T(x_k)r(x_k)$$

(after discarding 2nd order terms)

If GN method fails, Levenberg - Marquardt is the method of choice:

$$(J^T(x_k)J(x_k) + \underbrace{\mu_k I}_{\substack{\text{perturbation} \\ \text{regularization}}})s_k = -J^T(x_k)\cdot r(x_k)$$