Lecture 17

Last time

- What's to come
- One-dimensional Newton's method.
- Newton's in $\mathbb{R}^d$.

Today

- Convergence guarantee
- Computational complexity
- Quasi-Newton intro.

Local Convergence guarantees

Recall that given a matrix $A \in \mathbb{R}^{d \times d}$,

$$\|A\| = \max_{\|x\|_2 = 1} \|Ax\|_2.$$

Operator norm, or spectral norm.

Moreover if A is symmetric, then

$$\|A\| = \max_i \{|\lambda_i(A)|\}.$$ Eigenvalue.

Theorem: Let $F: \mathbb{R}^d \to \mathbb{R}^d$ be cont. diff. and assume $F(x^*)$ for some $x^* \in \mathbb{R}^d$ and $\nabla F(x^*)$ is nonsingular. Suppose that $\exists\, r > 0$ such that $\nabla F(x)$ is L-Lipschitz on $B(x^*, r)$.

Then for some $\varepsilon > 0$, we have that if $x_0 \in B(x^*, \varepsilon)$, then the iterates of Newton-Raphson satisfy

$x_k \in B(x^*, \varepsilon)$, $\nabla F(x_k)$ is nonsingular and

$$\|x_{k+1} - x^*\| \le c\,\|x_k - x^*\|^2,$$

for some fixed $c>0$. +

Proof: First let's state a Lemma

Lemma :): Assume $A, B \in \mathbb{R}^{d\times d}$. If A is nonsingular and $\|A^{-1}(B-A)\| < 1$, then B is nonsingular with

$$\|B^{-1}\| \le \frac{\|A^{-1}\|}{1 - \|A^{-1}(B-A)\|}.$$

+

With this lemma we can show a bound on $\|\nabla F(x_0)\|$.

Since $\nabla F(x^*)$ is invertible, we define $M = \|\nabla F(x^*)^{-1}\|$.

WLOG assume that $\forall\ x \in B(x, r)$, $\nabla F(x)$ is invertible.

Define $\varepsilon = \min\{r, 1/2ML\}$. Then, we have

$$\begin{aligned}
&\|\nabla F(x^*)^{-1}(\nabla F(x_0) - \nabla F(x^*))\| \\
&\quad\le \|\nabla F(x^*)^{-1}\|\,\|\nabla F(x_0) - \nabla F(x^*)\| \\
&\quad\le M L \|x_0 - x^*\| \le ML\varepsilon \le 1/2.
\end{aligned}$$

Thus, by Lemma ☺, $\nabla F(x_0)$ is invertible and $\|\nabla F(x_0)\| \leq 2M$.

Next we show quadratic improvement

$$\|x_1 - x^*\| = \|x_0 - x^* - \nabla F(x_0)^{-1} F(x_0)\|$$

$$= \|\nabla F(x_0)^{-1} (\nabla F(x_0)(x_0 - x^*) - F(x_0))\|$$

$$\leq \|\nabla F(x_0)^{-1}\| \, \| F(x^*) - \underbrace{(F(x_0) + \nabla F(x_0)(x_0 - x^*))}_{\text{Linear approx}} \|$$

Taylor Approximation

$$\leq 2M \frac{L}{2} \|x_0 - x^*\|^2.$$

We can inductively apply the same argument if $\|x_1 - x^*\| \leq \varepsilon$. Note that

$$\|x_1 - x^*\| \leq ML \|x_0 - x^*\|^2$$

$$\leq (ML\varepsilon) \cdot \varepsilon$$

$$\leq \varepsilon / 2.$$

Proof of Lemma ☺: Notice that B is invertable if, and only if, $A^{-1}B$ is invertable. It suffices to prove that $\|A^{-1}Bx\| > 0 \quad \forall x \in \mathbb{R}^d \setminus \{0\}$.

$$\|A^{-1}Bx\| = \|(I + A^{-1}(B - A))x\|$$

$$\geq \|Ix\| - \|A^{-1}(B-A)x\|$$

$$\geq (1 - \|A^{-1}(B-A)\|)\|x\| > 0.$$

To prove the bound on the norm,

$$\|B^{-1}\| \left(1 - \|A^{-1}(B-A)\|\right)$$

Cauchy-Schwarz

$$\le \|B^{-1}\| - \|A^{-1}(B-A)B^{-1}\|$$

Reverse triangle ineq.

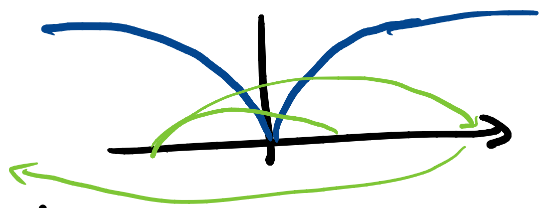
$$\le \|B^{-1} + A^{-1} - B^{-1}\|$$

$$= \|A^{-1}\|. \qquad \square$$

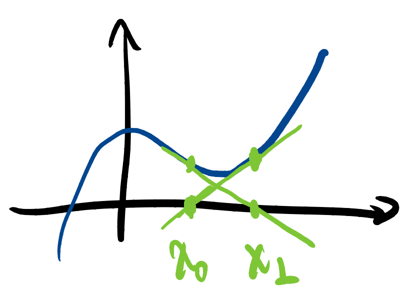
Caveats

- In HW4 you'll prove that
  - ★ Newton-Raphson might diverge

    $F(x) = |x|^{1/4}$



  - ★ It might also cycle



FRACTALS IN $\mathbb{C}$!

- When $\nabla F(x^*)$ is singular, then we either diverge or converge slower, $F(x) = x^2$, you can easily check

that
$$F(x_k) = \frac{1}{2^k} x_0.$$
linear rate.

▷ The method is sign invariant, Thus, the iterates are the same if we consider F or $-F$.
Not desirable when $F = \nabla f$.

Something really nice about this method is that it is affine invariant.

If $A \in \mathbb{R}^{d \times d}$ is invertible

$$F(x) = 0 \quad \equiv \quad F(Ay) =: G(y) = 0$$

$$x_0, x_1, \ldots \quad \equiv \quad y_0, y_1, \ldots$$

$$x_k = Ay_k.$$

Iteration cost / Computational complexity

Let's see the scaling of each operation and what we could do with a laptop:

In spirit

- Compute a gradient $O(d)$ memory / time

Newton's for smooth problems / Effectiveness

We can compute $d \sim 10^8 - 10^{10}$ with laptop

- Compute a Hessian $O(d^2)$ memory / time
  $d \sim 10^4 - 10^5$
- Solve $\nabla F(x_k)\, p = F(x_k)$ Worse than $O(d^2)$
  $d \sim 10^2 - 10^3$

If we solve directly $O(d^3)$.

Matrix factorization / triangular solve

People use indirect methods, e.g. conjugate gradient.

The cost of inverting a matrix at each iteration truly prevents us from scaling.

A potential alternative is Quasi-Newton methods.

Quasi-Newton Methods.

In the next couple of classes we'll cover

- Issues with eigenvalues
- Modified Newton
- Convergence guarantees
- Computational concerns
- Approximating Hessians / Secant Methods

▷ Quasi-Newton Methods (BFGS)

▷ Quasi-Newton superlinear convergence

Issues with Eigenvalues

As we discussed last time, Newton-Raphson moves to a critical point of

$$f_k(x) = f(x) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k)$$

Working with $\nabla^2 f(x)$, might be prohibitive, but we could consider general 2nd-order models:

$$m_k(x) = f_k + g_k^T(x - x_k) + \frac{1}{2}(x - x_k)^T B_k (x - x_k)$$

Constants are not relevant.

When $\left.\begin{aligned} f_k &= f(x_k) \\ g_k &= \nabla f(x_k) \\ H_k &= \nabla^2 f(x_k) \end{aligned}\right\}$ Newton's

When $\left.\begin{aligned} f_k &= f(x_k) \\ g_k &= \nabla f(x_k) \\ H_k &= \left(\frac{1}{\alpha_k}\right) I \end{aligned}\right\}$ Gradient descent.

When

$$\left.\begin{aligned} f_k &= f(x_k) \\ g_k &= \frac{\partial f}{\partial x_i}(x)\, e_i \\ H_k &= \left(\frac{1}{\alpha_k}\right) I \end{aligned}\right\} \text{Coordinate descent.}$$

Thus, a natural strategy is to consider

x_{k+1} is such that $\nabla m_k(x_{k+1}) = 0.$

which in turn reduces to

$$x_{k+1} = x_k - B_k^{-1} g_k.$$

when B_k is invertible.

Natural questions:

- How do we pick B_k so that we have descent?
- Can we make it cheaper per-iteration?