(5) Nonlinear equations

Develop machinery to solve \( f(x) = 0 \) \( \Rightarrow \) find \( x \).

\[ f : \mathbb{R}^n \rightarrow \mathbb{R}^n \]

What if we're really looking for \( f(x) = y \)?

Intuition: Each of the \( m \) equations describes a surface.

Existence? Not easy—many different ways.

- Intermediate value theorem (1D only, unhelpful in nD)

- Inverse function theorem

\[ f \in C, \text{ have solution } f(x) = 0, \ f'(x) \text{ invertible } \Rightarrow \text{ no extremum} \]

\[ \Rightarrow \text{ solutions also exist for } f(x) = y \text{ with } y \text{-close.} \]

Playing solvable for these \( y \)
- Contraction mapping thm.

\[ g: \mathbb{R}^n \to \mathbb{R}^n \text{ contractive} : \exists \alpha < 1 \text{ s.t.} \]

\[ \| g(x) - g(y) \| \leq \alpha \| x - y \| \]

**Fixed point of** \( g \): \( g(x) = x \)

**Real-world example:** map

Closed set \( S \subseteq \mathbb{R}^n \), \( g(S) \subseteq S \)

\( \Rightarrow \) there exists a unique fixed point.

**Uniqueness?**

- Hard to say anything general
- No (usually)

**Sensitivity**

\[ \text{cond (root finding) = cond (evaluation of inverse)} \]

**Multiple roots (1D)**

\[
\begin{align*}
\text{Set } & \quad f(x) = 0 \\
& \quad f'(x) = 0 \\
& \quad \vdots \\
& \quad f^{(m-1)}(x) = 0 \\
& \quad f^{(m)}(x) \neq 0
\end{align*}
\]

**W15 p101**
5.2 Convergence rates of iterative procedures

Let $e_k = \hat{u}_k - u$ be the error in the $k$th iterate $\hat{u}_k$.

Seen convergence behavior: $\|e_{k+1}\| < C \cdot \|e_k\|$

"Gain constant # of digits each step"

Faster convergence $\|e_{k+1}\| < C \cdot \|e_k\|^2$

"Doubling # of digits each step"

An iterative method converges with rate $r$ if

$$
\lim_{k \to \infty} \frac{\|e_{k+1}\|}{\|e_k\|^r} = C < \infty
$$

$r = 1$ \rightarrow "Linear"

$r = 2$ \rightarrow "Quadratic"

$r = 3$ \rightarrow "Cubic"

$> 1$: superlinear

WS16p2a
Recap: conv. rates

\[ \|e_{k+1}\| \leq C \|e_k\|^{r} \rightarrow 0 \]

rate matters when \( \|x_k\| \) is small

1\( \Rightarrow \frac{\|e_{k+1}\|}{\|e_k\|} \leq C \cdot \delta > 0 \) (but that doesn't mean anything!)

3\( \Rightarrow 0 < C_{\text{low}} \leq \frac{\|e_{k+1}\|}{\|e_k\|} \leq C_{\text{high}} > 0 \)

4\( \lim_{k \to \infty} \frac{\|e_{k+1}\|}{\|e_k\|} = C > 0 \)

5.3 Stopping criteria

- \( \|f(x)\| < \text{tol} \) "residual"
- \( \|x_{k+1} - x_k\| < \text{tol} \)
- \( \|x_{k+1} - x_{k+1}\| / \|x_k\| < \text{tol} \)

Why is none of these foolproof?
S.3 Methods in 1D

Bisection

Demo

Rate of convergence? Constant?

Fixed point iteration

$x_0 = \langle \text{some starting value} \rangle$

$x_{n+1} = g(x_n)$

Demo

When does FPI converge?

Assume:

- $g$ smooth
- $|g'(x)| < 1$ at fixed point $x^* = g(x)$

Error:

$e_{n+1} = x_{n+1} - x^* = g(x_n) - g(x^*)$

Mean Value Theorem:

$g(x_n) - g(x^*) = g'(\theta_n) (x_n - x^*)$

$\Rightarrow g' \left( \frac{e_n}{c} \right) \leq |g'(x)| < 1 \text{ in region around } x^*$

$\Rightarrow$ assume entire iteration in that region

$\Rightarrow e_{n+1} = C |e_n| < C^2 |e_{n-1}| < \ldots < C^k e_0$

$\Rightarrow e_n \rightarrow 0$ linearly with rate $C$. 
If \( g'(x^*) = 0 \), then convergence is faster than linear.

\[
g(x_k) - g(x^*) = g''(x^*) \frac{(x_k - x^*)^2}{2} \quad \text{(Taylor)}
\]

\Rightarrow \text{quadratic convergence!}

Would like to systematically find FPI \( y \) s.t. \( g'(x^*) = 0 \).

**Newton's method**

Idea: Approximate \( f \) at current iterate.

\[
f(x + h) \approx \frac{f(x) + f'(x) h}{f'(x)} \quad \Rightarrow \quad \text{find root of } g
\]

\[
f(x) + f'(x) h = 0 \quad \Rightarrow \quad h = -\frac{f(x)}{f'(x)}
\]

\[
\begin{align*}
x_0 & = \text{initial guess} \\
x_{k+1} & = x_k + h = x_k - \frac{f(x)}{f'(x)}
\end{align*}
\]

Can view Newton as FPI:

\[
g(x) = x - \frac{f(x)}{f'(x)} \quad \quad g'(x) = \frac{f(x) f''(x)}{f'(x)^2}
\]

\[
f(x^*) = 0 \quad , \quad f'(x^*) = 0 \Rightarrow g'(x^*) = 0
\]

What does this mean? What does this imply?
Drawback? → Have to have derivative! 
   → locally convergent

Secant method

Recall: Use approximate derivative in Newton:

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

Rate of convergence: \((1 + \sqrt{5})/2 \approx 1.618\).

Drawbacks? → slower conv. 
   → two starting guesses
   → locally convergent

Dems: Newton and secant

Where should you "trust" a Newton/secant approximation?

→ limit steps to that region
→ "trust region" methods

WS8p1
5.4 Methods in nD ("Systems of equations")

Fixed point iteration

\[ x_0 = \text{starting guess} \]
\[ x_{n+1} = g(x_n) \]

Converges (locally) if \( g'(g(x^*)) < 1 \).

Jacobian matrix: \( f_j(x) = \begin{pmatrix} \end{pmatrix} \)

Also: \( Dg = g' = f_g \)

\( f_g(x^*) = 0 \Rightarrow \) at least quadratic convergence.

Newton

\[ f(x + s) = f(x) + \frac{d}{dt} f(x) |_{t=0} \]

\( \Rightarrow \)
\[ \frac{d}{dt} f_x \big|_{t=0} x = -f(x) \]
\[ x = \left( \frac{d}{dt} f_x \big|_{t=0} \right)^{-1} f(x) \]

\[ x_0 = \text{starting guess} \]
\[ x_{n+1} = x_n - \left( \frac{d}{dt} f_x \big|_{t=0} \right)^{-1} f(x_n) \]
Downsides: 
  → locally convergent
  → computing/inverting \( \mathbf{J} \) expensive

Fix: Use approximate Jacobian matrix satisfying

\[
\tilde{\mathbf{J}}(x_k) = \mathbf{J}(x_{km}) - \mathbf{J}(x_k) 
\]

How many equations?
How many unknowns?

One choice: Broyden's method (minimizes change to \( \mathbf{J} \))