# Iterative methods for nonlinear equations

MATH 6610 Lecture 23

November 6, 2020

# Nonlinear equations

Given $f : \mathbb{R}^n \to \mathbb{R}^n$ a general nonlinear function, consider solving for $x$:

$$f(x) = 0.$$

This problem is in general both theoretically and computationally difficult.

- Existence and uniqueness can be difficult to establish
- Iterative algorithms are the typical strategy
- Algorithm success varies wildly depending on the initial iterate

# Nonlinear equations

Given $f : \mathbb{R}^n \to \mathbb{R}^n$ a general nonlinear function, consider solving for $x$:

$$f(x) = 0.$$

This problem is in general both theoretically and computationally difficult.

- Existence and uniqueness can be difficult to establish
- Iterative algorithms are the typical strategy
- Algorithm success varies wildly depending on the initial iterate

Even with $n = 1$ this is a relatively difficult problem.

There are some standard algorithms for addressing this problem.

We'll only look at a few, but there are numerous methods.

# Linearizations

In some cases nonlinear equations can be *linearized*, which informally means that solutions to the nonlinear equation

$$f(x) = 0,$$

can be expressed with linear objects and operators.

# Linearizations

In some cases nonlinear equations can be *linearized*, which informally means that solutions to the nonlinear equation

$$f(x) = 0,$$

can be expressed with linear objects and operators.

A well-known example of linearizations is finding roots of a polynomial:

$$f(x) := x^m + \sum_{j=0}^{m-1} a_j x^j = 0.$$

This is a nonlinear equation for any $m > 1$.

# Companion matrices

$$f(x) := x^m + \sum_{j=0}^{m-1} a_j x^j = 0.$$

Define $C \in \mathbb{C}^{m\times m}$ by

$$C = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{m-1} \end{pmatrix}.$$

This matrix $C$ is a *companion matrix*.

# Companion matrix linearizations

A computation shows that if $x \in \mathbb{C}$ is a(ny) root of $f$, then

$$v = \begin{pmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^{m-1} \end{pmatrix}$$

is an eigenvector of $C$ with eigenvalue $x$.
In other words, the spectrum of $C$ are exactly the set of points that solve $f(x) = 0$.

# Companion matrix linearizations

A computation shows that if $x \in \mathbb{C}$ is a(ny) root of $f$, then

$$v = \begin{pmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^{m-1} \end{pmatrix}$$

is an eigenvector of $C$ with eigenvalue $x$.
In other words, the spectrum of $C$ are exactly the set of points that solve $f(x) = 0$.

$$f(x) := x^m + \sum_{j=0}^{m-1} a_j x^j = 0 \quad \Leftrightarrow \quad x \in \Lambda(C).$$

While this provides a way to compute roots via eigenvalue problems, often $C$ is ill-conditioned.
In particular, $C$ is not a normal matrix, so the eigenvalue problem is often poorly conditioned.

# Bisection

$$f(x) = 0 \qquad (n = 1)$$

Perhaps the simplest procedure is bisection: assume $f$ is continuous, and that we have two values $x_L$ and $x_R$ such that

$$x_L < x_R, \qquad\qquad f(x_L)f(x_R) < 0,$$

i.e., $f(x_L)$ and $f(x_R)$ have different signs.

# Bisection

$$f(x) = 0 \qquad (n = 1)$$

Perhaps the simplest procedure is bisection: assume $f$ is continuous, and that we have two values $x_L$ and $x_R$ such that

$$x_L < x_R, \qquad\qquad f(x_L)f(x_R) < 0,$$

i.e., $f(x_L)$ and $f(x_R)$ have different signs.

In this situation, there must be some solution $x^* \in (x_L, x_R)$ (Intermediate Value Theorem).

The bisection algorithm zeros in on one such solution by creating a smaller and samller bracketing interval:

1. Define $x_M := \frac{1}{2}(x_L + x_R)$, and compute $f(x_M)$.
2. If $f(x_M)f(x_L) < 0$: set $x_R \leftarrow x_M$ and return to step 1.
3. If $f(x_M)f(x_R) < 0$: set $x_L \leftarrow x_M$ and return to step 1.
4. If $f(x_M) = 0$: then $x^* = x_M$.

# Fixed point iteration, I



$$f(x) = 0$$

Fixed point iteration is an alternative, simple strategy that rewrites the equation above as

$$x = g(x),$$

where, for example, $g(x) = x - f(x)$.

# Fixed point iteration, I

$$f(x) = 0$$

Fixed point iteration is an alternative, simple strategy that rewrites the equation above as

$$x = g(x),$$

where, for example, $g(x) = x - f(x)$.

Under certain assumptions, the Banach fixed point theorem

- guarantees a unique solution to $x = g(x)$,
- that the solution is the limit of the sequence $\{x_n\}$ defined by $x_n \coloneqq g(x_{n-1})$.

# Fixed point iteration, I

$$x = g(x),$$

In order to leverage the Banach fixed point theorem results, $g$ must be a contraction:

- There is some region $D \subseteq \mathbb{R}^n$ such that $g : D \to D$.
- There is some $\lambda \in [0, 1)$ such that $g$ satisfies $\|g(x) - g(y)\| \leqslant \lambda \|x - y\|$ for every $x, y \in D$.

# Fixed point iteration, I



$$x = g(x),$$

In order to leverage the Banach fixed point theorem results, $g$ must be a contraction:

- There is some region $D \subseteq \mathbb{R}^n$ such that $g : D \to D$.
- There is some $\lambda \in [0, 1)$ such that $g$ satisfies $\|g(x) - g(y)\| \leqslant \lambda \|x - y\|$ for every $x, y \in D$.

Note that the contraction property is satisfied if, for example,

$$\sup_{x \in D} \left\| \frac{\mathrm{d}g}{\mathrm{d}x} \right\| < 1,$$

where $\frac{\mathrm{d}g}{\mathrm{d}x}$ is the Jacobian of $g$.

# Fixed point iteration, I

$$x = g(x),$$

In order to leverage the Banach fixed point theorem results, $g$ must be a contraction:

- There is some region $D \subseteq \mathbb{R}^n$ such that $g : D \to D$.
- There is some $\lambda \in [0, 1)$ such that $g$ satisfies $\|g(x) - g(y)\| \leqslant \lambda \|x - y\|$ for every $x, y \in D$.

Note that the contraction property is satisfied if, for example,

$$\sup_{x \in D} \left\| \frac{\mathrm{d}g}{\mathrm{d}x} \right\| < 1,$$

where $\frac{\mathrm{d}g}{\mathrm{d}x}$ is the Jacobian of $g$.

Several methods for solving nonlinear equations are variants of fixed point iteration which, given $f$, make special choices for $g$ to ensure the contraction property.

# Newton's Method ($n = 1$)

Newton's Method solves

$$f(x) = 0 \qquad (n = 1)$$

by casting the problem as the following fixed point iteration:

$$x = g(x) := x - \frac{f(x)}{f'(x)}$$

Note that any solution to $x = g(x)$ also satisfies $f(x) = 0$.

# Newton's Method ($n = 1$)

Newton's Method solves

$$f(x) = 0 \qquad (n = 1)$$

by casting the problem as the following fixed point iteration:

$$x = g(x) := x - \frac{f(x)}{f'(x)}$$

Note that any solution to $x = g(x)$ also satisfies $f(x) = 0$.

Newton's method applies fixed point iteration:

$$x_n := g(x_{n-1}) = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})},$$

where $x_0$ must be chosen.

# Effectiveness of Newton's Method



Newton's Method, under certain assumptions, attains *quadratic* convergence, i.e.,

$$|x - x_n| \leqslant C \, |x - x_{n-1}|^2 \, ,$$

where $x$ is a root of $f(x)$, and $C$ satisfies $C|x - x_0| < 1$.

# Effectiveness of Newton's Method

Newton's Method, under certain assumptions, attains *quadratic* convergence, i.e.,

$$|x - x_n| \leqslant C\,|x - x_{n-1}|^2\,,$$

where $x$ is a root of $f(x)$, and $C$ satisfies $C|x - x_0| < 1$.

Failure of Newton's Method often results from a poor choice of $x_0$, or from $f$ not satisfying technical conditions that would ensure success of the method.

However, if $x_0$ is "close enough" to $x$, then Newton's methods often performs extremely well.

# Effectiveness of Newton's Method

Newton's Method, under certain assumptions, attains *quadratic* convergence, i.e.,

$$|x - x_n| \leqslant C\,|x - x_{n-1}|^2\,,$$

where $x$ is a root of $f(x)$, and $C$ satisfies $C|x - x_0| < 1$.

Failure of Newton's Method often results from a poor choice of $x_0$, or from $f$ not satisfying technical conditions that would ensure success of the method.

However, if $x_0$ is "close enough" to $x$, then Newton's methods often performs extremely well.

Some methods combine slower and less sophisticated methods, like bisection, to first obtain a guess that is "close" to $x$.

Subsequently, a faster method, like Newton's Method, is used to converge quickly to the solution.

# Newton's Method $(n > 1)$



$$f(x) = 0 \qquad (n > 1)$$

A multivariate form of Newton's Method looks similar to the one-dimensional case:

$$x = g(x) := x - \left(\frac{\mathrm{d}f}{\mathrm{d}x}\right)^{-1} f(x),$$

and the iterates are defined as $x_n = g(x_{n-1})$.

Note in particular that this requires inversion of a (potentially large) matrix at every step.