Newton's Method for Nonlinear Equations

One-Dimensional Root Finding

MMAE 350 — Computational Mechanics

Motivation

- Many engineering problems lead to equations of the form

$$f(x) = 0$$

- These equations are often *nonlinear*
- Closed-form solutions rarely exist
- We need a *systematic numerical method*

Motivation

- Many engineering problems lead to equations of the form

$$f(x) = 0$$

- These equations are often *nonlinear*
- Closed-form solutions rarely exist
- We need a *systematic numerical method*

Newton's method is one of the most important tools in computational mechanics.

Examples from Mechanics

Nonlinear equations arise when:

- Material behavior is nonlinear (plasticity, large strain)
- Geometry changes with deformation
- Loads depend on displacement

Examples:

- Nonlinear stress–strain laws
- Contact problems
- Large deformation equilibrium

The Root-Finding Problem

We want to solve:

$$f(x) = 0$$

- $f(x)$ is known
- x is unknown
- We want the value of x where the curve crosses zero

The Root-Finding Problem

We want to solve:

$$f(x) = 0$$

- $f(x)$ is known
- x is unknown
- We want the value of x where the curve crosses zero

This is called a *root* of the function.

Geometric Idea Behind Newton's Method

Start with an initial guess $x^{(0)}$.

Geometric Idea Behind Newton's Method

Start with an initial guess $x^{(0)}$.

- Approximate $f(x)$ near $x^{(k)}$ by its tangent line
- Use the tangent line to predict where $f(x) = 0$
- Use that prediction as the next guess

Geometric Idea Behind Newton's Method

Start with an initial guess $x^{(0)}$.

- Approximate $f(x)$ near $x^{(k)}$ by its tangent line
- Use the tangent line to predict where $f(x) = 0$
- Use that prediction as the next guess

Newton's method is *repeated linearization*.

Linearization via Taylor Series

Expand $f(x)$ about the current iterate $x^{(k)}$:

$$f(x) \approx f(x^{(k)}) + f'(x^{(k)})(x - x^{(k)})$$

Set the approximation equal to zero:

$$0 \approx f(x^{(k)}) + f'(x^{(k)})(x^{(k+1)} - x^{(k)})$$

Newton Update Formula

Solving for $x^{(k+1)}$ gives:

$$\boxed{x^{(k+1)} = x^{(k)} - \frac{f\left(x^{(k)}\right)}{f'\left(x^{(k)}\right)}}$$

Newton Update Formula

Solving for $x^{(k+1)}$ gives:

$$\boxed{x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}}$$

This is the **Newton update**.

Algorithm Summary

Newton's Method (1D)

1. Choose an initial guess $x^{(0)}$
2. For $k = 0, 1, 2, \ldots$:
   - Evaluate $f(x^{(k)})$
   - Evaluate $f'(x^{(k)})$
   - Update:

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}$$

3. Stop when $|f(x^{(k)})|$ is small

Convergence Behavior

If Newton's method converges:

- Convergence is typically **quadratic**
- Error decreases very rapidly near the root

Convergence Behavior

If Newton's method converges:

- Convergence is typically **quadratic**
- Error decreases very rapidly near the root

But:

- Requires a good initial guess
- Can fail if $f'(x)$ is small or zero

Why Initial Guess Matters

Newton's method is *local*.

- Works best when starting near the solution
- Poor guesses can lead to divergence
- In mechanics, physics often gives a good initial guess

Why Initial Guess Matters

Newton's method is *local*.

- Works best when starting near the solution
- Poor guesses can lead to divergence
- In mechanics, physics often gives a good initial guess

Example: Linear elasticity is often a good starting point.

Newton's Method in Mechanics Language

- $f(x) =$ residual (force imbalance)
- $f'(x) =$ tangent stiffness
- Newton update $=$ solve linearized equilibrium

Newton's Method in Mechanics Language

- $f(x)$ = residual (force imbalance)
- $f'(x)$ = tangent stiffness
- Newton update = solve linearized equilibrium

Each Newton step:

- Linearizes the problem
- Solves a linear system
- Updates the configuration

Connection to Finite Elements

In 1D:

$$f'(x) = \frac{df}{dx}$$

In many degrees of freedom:

$$\mathbf{K}_T \Delta \mathbf{u} = -\mathbf{R}$$

Connection to Finite Elements

In 1D:

$$f'(x) = \frac{df}{dx}$$

In many degrees of freedom:

$$\mathbf{K}_T \Delta \mathbf{u} = -\mathbf{R}$$

- Tangent stiffness matrix
- Residual force vector
- Same idea, larger system

Stopping Criteria

Common choices:

- $|f(x^{(k)})| <$ tolerance
- $|x^{(k+1)} - x^{(k)}| <$ tolerance

Stopping Criteria

Common choices:

- $|f(x^{(k)})| <$ tolerance
- $|x^{(k+1)} - x^{(k)}| <$ tolerance

In mechanics, residual-based criteria are most common.

What Can Go Wrong?

- Poor initial guess
- Very stiff or very soft nonlinearities
- Large step overshooting the solution

What Can Go Wrong?

- Poor initial guess
- Very stiff or very soft nonlinearities
- Large step overshooting the solution

Fixes:

- Damping (line search)
- Smaller load steps
- Better initial guesses

Takeaway

- Newton's method solves nonlinear equations by repeated linearization
- It is fast and powerful when used correctly
- It is the foundation of nonlinear finite element analysis

Takeaway

- Newton's method solves nonlinear equations by repeated linearization
- It is fast and powerful when used correctly
- It is the foundation of nonlinear finite element analysis

In this course: Newton's method is not just an algorithm — it is a modeling idea.