Lecture 4: Extending Newton's method

COMP5930M Scientific Computation

Today

Recap

We have seen two basic methods for scalar nonlinear equations

- Bisection
  Robust, slow, requires valid starting bracket
- Newton
  Fast, unreliable, requires derivative,
  requires good initial guess

We would like the best features of both

The Secant Method

Replace the derivative required by Newton's Method with a numerical approximation

$$\frac{dF}{dx}(x_n) \approx \frac{F(x_n + \delta) - F(x_n)}{\delta}$$

Secant method:

$$x_{n+1} = x_n - \frac{\delta F(x_n)}{F(x_n + \delta) - F(x_n)}$$

The Secant Method

Replace the derivative required by Newton's Method with a numerical approximation

$$\frac{dF}{dx}(x_n) \approx \frac{F(x_n + \delta) - F(x_n)}{\delta}$$

Secant method:

$$x_{n+1} = x_n - \frac{\delta F(x_n)}{F(x_n + \delta) - F(x_n)}$$

Issues:

- Choosing δ
- Convergence rate?
- Effect on the overall algorithm

The Secant Method

Replace the derivative required by Newton's Method with a numerical approximation

$$\frac{dF}{dx}(x_n) \approx \frac{F(x_n+\delta) - F(x_n)}{\delta}$$

Secant method:

$$x_{n+1} = x_n - \frac{\delta F(x_n)}{F(x_n+\delta) - F(x_n)}$$

Issues:

- Choosing δ
- Convergence rate?
- Effect on the overall algorithm

Numerical derivatives

- Allowing for finite-precision arithmetic we can show that the optimal value of $\delta \propto \sqrt{eps}$
- *eps* is the machine precision, typically 10^{-16}, which gives roughly 10^{-8} accuracy for the derivative
- In practice $\delta = 10\sqrt{eps}$ is often a default value

In practice

- The previous method requires an additional function evaluation at each step - inefficient
- Instead use $F(x_{n-1})$, the previous iterate

$$\frac{dF}{dx}(x_n) \approx \frac{F(x_n) - F(x_{n-1})}{x_n - x_{n-1}}$$

- Less accurate (δ larger) but more efficient
- Requires two data points to start the algorithm

Secant method

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{F(x_n) - F(x_{n-1})} F(x_n)$$

- Order of convergence can be shown to be $q = \frac{\sqrt{5}+1}{2} \approx 1.618$
- The performance in practice is similar to Newton's method
- Still lacks robustness

Dekker's Method

Combine the advantages of bisection and Newton/secant

Algorithm outline:

1. Define an initial bracket $[x_0, x_1]$ s.t. $F(x_0)F(x_1) < 0$.
2. If $|F(x_0)| < |F(x_1)|$ we swap their order, $x_0 \leftrightarrow x_1$. Take a step with the secant method:

$$x_2 = x_1 - \frac{x_1 - x_0}{F(x_1) - F(x_0)} F(x_1).$$

Dekker's Method

Combine the advantages of bisection and Newton/secant

Algorithm outline:

1. Define an initial bracket $[x_0, x_1]$ s.t. $F(x_0)F(x_1) < 0$.
2. If $|F(x_0)| < |F(x_1)|$ we swap their order, $x_0 \leftrightarrow x_1$. Take a step with the secant method:

$$x_2 = x_1 - \frac{x_1 - x_0}{F(x_1) - F(x_0)} F(x_1).$$

3a. If $x_2 \in [x_0, x_1]$ define a new bracket $[x_0, x_2]$ or $[x_2, x_1]$ depending on if $F(x_0)F(x_2) \leq 0$ or $F(x_2)F(x_1) < 0$.

Dekker's Method

Combine the advantages of bisection and Newton/secant

Algorithm outline:

1. Define an initial bracket $[x_0, x_1]$ s.t. $F(x_0)F(x_1) < 0$.
2. If $|F(x_0)| < |F(x_1)|$ we swap their order, $x_0 \leftrightarrow x_1$.
   Take a step with the secant method:

$$x_2 = x_1 - \frac{x_1 - x_0}{F(x_1) - F(x_0)} F(x_1).$$

3a. If $x_2 \in [x_0, x_1]$ define a new bracket $[x_0, x_2]$ or $[x_2, x_1]$ depending on if $F(x_0)F(x_2) \leq 0$ or $F(x_2)F(x_1) < 0$.

3b. If $x_2 \notin [x_0, x_1]$ reject x_2 and use one step of the bisection method to define a new bracket

Dekker's Method

Combine the advantages of bisection and Newton/secant

Algorithm outline:

1. Define an initial bracket $[x_0, x_1]$ s.t. $F(x_0)F(x_1) < 0$.
2. If $|F(x_0)| < |F(x_1)|$ we swap their order, $x_0 \leftrightarrow x_1$.
   Take a step with the secant method:

$$x_2 = x_1 - \frac{x_1 - x_0}{F(x_1) - F(x_0)} F(x_1).$$

3a. If $x_2 \in [x_0, x_1]$ define a new bracket $[x_0, x_2]$ or $[x_2, x_1]$ depending on if $F(x_0)F(x_2) \leq 0$ or $F(x_2)F(x_1) < 0$.

3b. If $x_2 \notin [x_0, x_1]$ reject x_2 and use one step of the bisection method to define a new bracket

4. Test if converged. If not, increment n and iterate from 2.

Dekker's Method

Combine the advantages of bisection and Newton/secant

Algorithm outline:

1. Define an initial bracket $[x_0, x_1]$ s.t. $F(x_0)F(x_1) < 0$.
2. If $|F(x_0)| < |F(x_1)|$ we swap their order, $x_0 \leftrightarrow x_1$.
   Take a step with the secant method:

$$x_2 = x_1 - \frac{x_1 - x_0}{F(x_1) - F(x_0)} F(x_1).$$

3a. If $x_2 \in [x_0, x_1]$ define a new bracket $[x_0, x_2]$ or $[x_2, x_1]$ depending on if $F(x_0)F(x_2) \leq 0$ or $F(x_2)F(x_1) < 0$.

3b. If $x_2 \notin [x_0, x_1]$ reject x_2 and use one step of the bisection method to define a new bracket

4. Test if converged. If not, increment n and iterate from 2.

Comments

- Requires a search for an initial bracket
- Will use (fast) secant whenever possible
- Will use robust (slow) bisection
  whenever secant appears unreliable
- The basis for Matlab's `fzero()` function

Summary

- Scalar nonlinear equations can be solved in a robust way
- The initial guess (or bracket) is often a bigger challenge

Next week

Systems of nonlinear equations