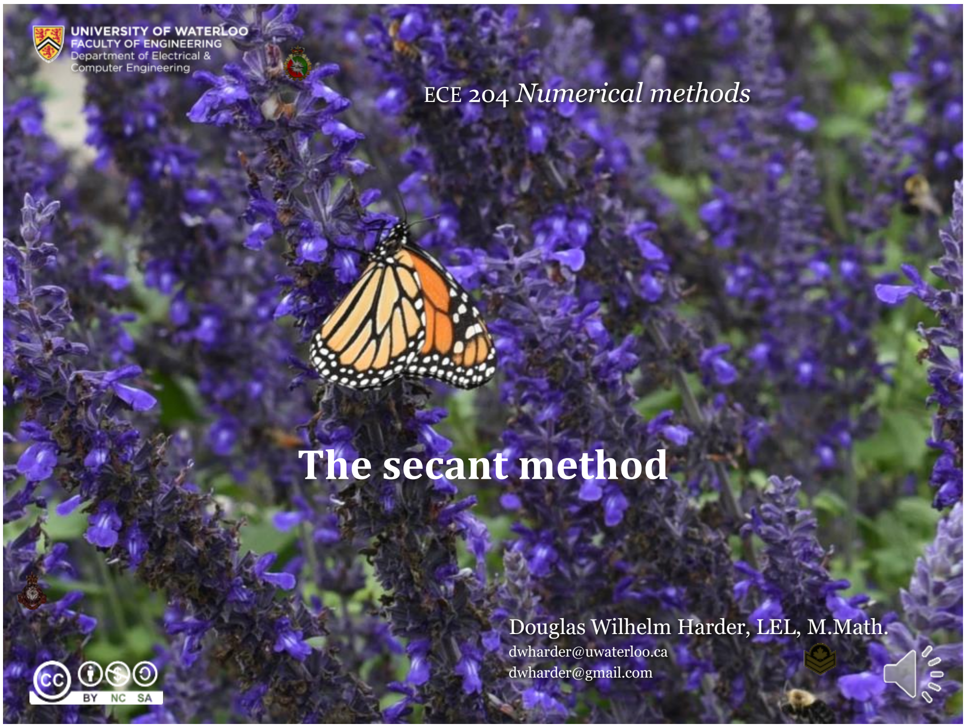


1

Introduction

- In this topic, we will
  - Remove the bracketing restriction on the bracketed secant method
  - We will look at an example and estimate how quickly it converges
  - Discuss how to best find how quickly the error goes to zero
  - Look at an implementation

2

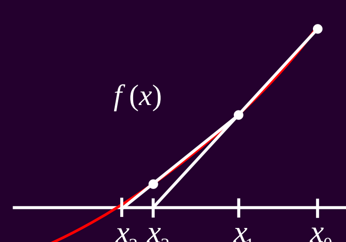
Removing bracketing

- Newton's method simply requires us to have an initial guess
  - Issue: it requires the derivative
- The bracketed secant method does not require the derivative
  - Issue: it requires us to bracket the root
  - Issue: the bracketing appears to slow down convergence: O(*h*)
- What if we don't have a bracket for the root and we don't have a derivative?

Removing bracketing

- Suppose we have a current guess x_k and a previous guess x_{k-1}
  - Could we not just find the root of the interpolating linear polynomial?

$$x_{k+1} \leftarrow x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$



Removing bracketing

- Benefit: We do not require that the root is bracketed
- Drawback: Like Newton's method,
  this method is not guaranteed to converge
- Question: Does this change the rate of convergence?

5

5

Example

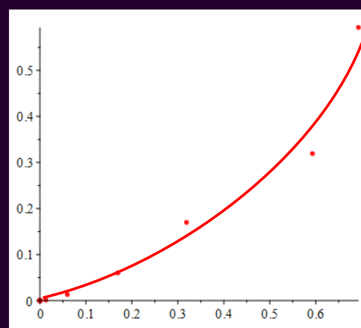
- Find the first root of $2e^{-2x} - e^{-x}$
  - The solution is ln(2)

k	x_k	$f(x_k)$	$\ln(2) - x_k$	$\dfrac{\ln(2) - x_k}{\ln(2) - x_{k-1}}$
0	0	1	0.6931	
1	0.1	0.7326	0.5931	
2	0.374005269572983	0.2586	0.3191	0.5380
3	0.523523434122555	0.1095	0.1696	0.5315
4	0.6333278455599131	0.03272	0.0598	0.3527
5	0.6801164924137422	0.006644	0.01303	0.2178
6	0.6920368309961099	0.0005561	0.001110	0.08521
7	0.6931256976122437	0.00001074	0.00002148	0.01935
8	0.6931471448088051	0.00000001788	0.00000003575	0.001664
9	0.6931471805587934	0.0000000000005759	0.000000000001152	0.00003222
10	0.6.931471805599452	0	0.0000000000000000001110	0.00009639

6

Rate of convergence

- How do we estimate the rate of convergence?
  - If we plot the current error versus the next error, we see a trend

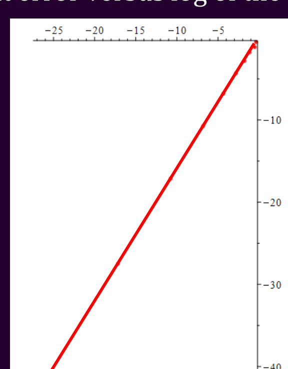


Rate of convergence

- How do we estimate the rate of convergence?
  - If an expression reduces according to $h_{k+1} = ah_k^{\,b}$ then

$$\ln(h_{k+1}) = \ln(ah_k^{\,b}) = \ln(a) + b\ln(h_k)$$

  - Plot log of the current error versus log of the next error



$0.02656 + 1.599h$

$h_{k+1} = 1.027\, h_k^{1.599}$

  - Beyond the scope of this course:
    - The secant method is $O(h^{\phi})$

Implementation

```
double secant( double f( double x ), double x0, double x1,
               double eps_step, double eps_abs,
               unsigned int max_iterations ) {
    double f0{ f( x0 ) };
    double f1{ f( x1 ) };

    assert( std::abs( f1 ) > std::abs( f2 ) );

    if ( f1 == 0.0 ) {
        return x1;
    }
```


Implementation

```
    for ( unsigned int k{0}; k < max_iterations; ++k ) {
        double x2{ x1 - f1*(x1 - x0)/(f1 - f0) };
        double f2{ f( x2 ) };

        if ( f2 == 0.0 ) {
            return x2;
        } else {
            if ( (std::abs( f2 ) < eps_abs)
                 && (std::abs(x2 - x1) < eps_step) ) {
                return x2;
            }
            x0 = x1;
            f0 = f1;
            x1 = x2;
            f1 = f2;
        }
    }
```

Implementation

```
        return NAN;
    }
```


Summary

- Following this topic, you now
  - Understand the more generalized secant method
    - It does not require bracketing the root
  - Are aware it may not converge, like Newton's method
  - Understand that we can deduce the rate of change by plotting a log-log plot and using least-squares best-fitting polynomials
  - Know that the secant method is $O(h^{\phi})$

References

[1] https://en.wikipedia.org/wiki/Secant_method

13

13

Acknowledgments

None so far.

14

14

Colophon

These slides were prepared using the Cambria typeface. Mathematical equations use Times New Roman, and source code is presented using `Consolas`.
Mathematical equations are prepared in MathType by Design Science, Inc.
Examples may be formulated and checked using Maple by Maplesoft, Inc.

The photographs of flowers and a monarch butter appearing on the title slide and accenting the top of each other slide were taken at the Royal Botanical Gardens in October of 2017 by Douglas Wilhelm Harder. Please see

https://www.rbg.ca/

for more information.

Disclaimer

These slides are provided for the ECE 204 *Numerical methods* course taught at the University of Waterloo. The material in it reflects the author's best judgment in light of the information available to them at the time of preparation. Any reliance on these course slides by any party for any other purpose are the responsibility of such parties. The authors accept no responsibility for damages, if any, suffered by any party as a result of decisions made or actions based on these course slides for any other purpose than that for which it was intended.