Nonlinear Equations of One Variable I

- Solve $f(x) = 0, x \in R^1$
- This means to find a root
- The difference from solving linear systems: nonlinear but not linear

Bisection method I

- The procedure:
  - Given a_1, b_1 with
    $$f(a_1)f(b_1) < 0$$
    Set $i = 1$
  - At the current $[a_i, b_i]$, compute
    $$p_i = \frac{a_i + b_i}{2}$$
    If
    $$f(p_i) = 0, \text{ stop}$$

Bisection method II

Set

$$a_{i+1} = a_i \text{ and } b_{i+1} = p_i, \text{ if } f(a_i)f(p_i) < 0$$

and

$$a_{i+1} = p_i \text{ and } b_{i+1} = b_i, \text{ otherwise}$$

- $i \leftarrow i + 1$, go to the next iteration

Bisection method III

- The key is to keep having

$$f(a_i)f(b_i) < 0$$

  Between such a_i and b_i, for any continuous function, a root exists in $[a_i, b_i]$
- Example: (code from the book "Numerical methods" (second edition) by Faires and Burden)

Bisection method IV

```
>> bisect21
This is the Bisection Method.
Input the function F(x) in terms of x
For example: cos(x)
  'cos(x)'
Input endpoints A < B on separate lines
 0
 1
F(A) and F(B) have same sign
Input endpoints A < B on separate lines
 0
 0.5
F(A) and F(B) have same sign
Input endpoints A < B on separate lines
 0
 2
```

Bisection method V

```
Input tolerance
 0.001
Input maximum number of iterations - no decimal
 50
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 2
Bisection Method
  I    P                    F(P)
```

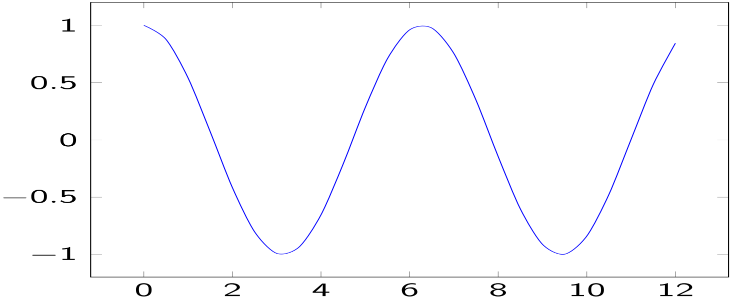
Bisection method VI

```
 1     1.00000000e+00      5.4030231e-01
 2     1.50000000e+00      7.0737202e-02
 3     1.75000000e+00     -1.7824606e-01
 4     1.62500000e+00     -5.4177135e-02
 5     1.56250000e+00      8.2962316e-03
 6     1.59375000e+00     -2.2951658e-02
 7     1.57812500e+00     -7.3286076e-03
 8     1.57031250e+00      4.8382678e-04
 9     1.57421875e+00     -3.4224165e-03
10     1.57226562e+00     -1.4692977e-03
11     1.57128906e+00     -4.9273569e-04

Approximate solution P =  1.57128906
with F(P) =  -0.00049274
Number of iterations =  11 Tolerance =  1.00000
```

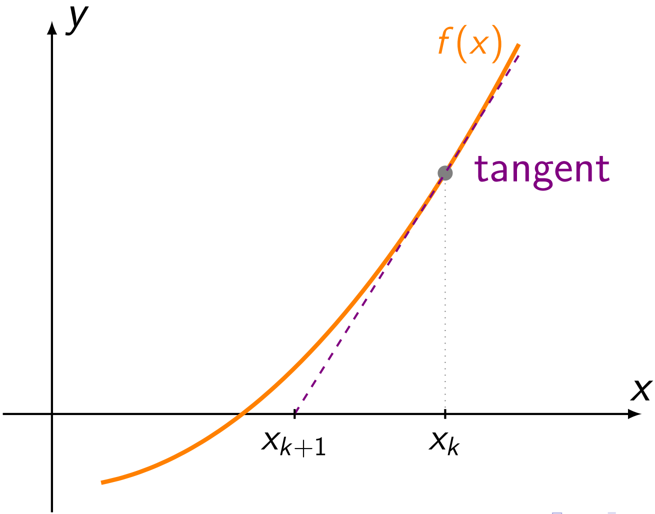
Bisection method VII

The $\cos x$ function



Our procedure finds a solution close to $\pi/2$

Newton method I



Newton method II

Latex source of the above figure was modified from https://tex.stackexchange.com/questions/551160/plot-that-demonstrate-newtons-method

- Solve

$$f(x) = 0$$

- Find the tangent line at x_k:

$$\frac{y - f(x_k)}{x - x_k} = f'(x_k)$$

x_k: the current iterate

Newton method III

- Obtain the intersection between the tangent line and the x-axis. Let $y = 0$

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

- Newton method can also be derived from the viewpoint of solving an optimization problem

$$\min f(x)$$

Newton method IV

This is (roughly) equivalent to solving $f'(x) = 0$

$$\begin{aligned} f(x+d) &= f(x) + f'(x)d + \frac{1}{2}d^2 f''(x) + \cdots \\ &\approx f(x) + f'(x)d + \frac{1}{2}d^2 f''(x) \end{aligned}$$

- Second-order approximation

$$\min_d f(x) + f'(x)d + \frac{1}{2}d^2 f''(x)$$

Newton method V

Then

$$f''(x)d + f'(x) = 0$$

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

$$x_{k+1} = x_k - \frac{f'(x)}{f''(x)}.$$

- Newton method may not converge. That is,

$$\{x_k\} \text{ may diverge}$$

Newton method VI

- Example of using Newton method: (code from the book "Numerical methods" (second edition) by Faires and Burden)

```
>> newton
newton
This is Newtons Method
Input the function F(x) in terms of x
For example: cos(x)
 'cos(x)'
'cos(x)'
Input the derivative of F(x) in terms of x
 '-sin(x)'
'-sin(x)'
Input initial approximation
 1
```

Newton method VII

```
1
Input tolerance
 0.001
0.001
Input maximum number of iterations - no deci
 50
50
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
1
Select amount of output
1. Answer only
2. All intermediate approximations
```

Newton method VIII

```
Enter 1 or 2
 2
2
Newtons Method
  I   P                 F(P)
  1   1.64209262e+00   -7.1235903e-02
  2   1.57067528e+00    1.2104963e-04
  3   1.57079633e+00   -5.9124355e-13

Approximate solution = 1.5707963268e+00
with F(P) = -5.9124355058e-13
Number of iterations = 3
Tolerance = 1.0000000000e-03
```

- Another run using a different initial point

Newton method IX

```
>> newton
This is Newtons Method
Input the function F(x) in terms of x
For example: cos(x)
 'cos(x)'
Input the derivative of F(x) in terms of x
 '-sin(x)'
Input initial approximation
 0.1
Input tolerance
 0.001
Input maximum number of iterations - no deci
 50
Select output destination
1. Screen
2. Text file
```

Newton method X

```
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 2
Newtons Method
  I   P                   F(P)
  1   1.00666444e+01   -8.0097972e-01
  2   1.14045284e+01    3.9764987e-01
  3   1.09711401e+01   -2.4431758e-02
  4   1.09955792e+01    4.8638065e-06
  5   1.09955743e+01   -4.2862638e-16
Approximate solution = 1.0995574288e+01
with F(P) = -4.2862637970e-16
Number of iterations = 5
```

Newton method XI

```
Tolerance = 1.0000000000e-03
>>
```

- Our procedure finds a solution close to $\pi/2$
- The number of Newton iterations is smaller than bisection