§4.1 Roots of Equations

- Given $f(x)$, determine values of x for which $f(x)=0$; these values of x are called roots or zeros of $f(x)$.
- Roots may be real or complex numbers but in this course we focus on computing real roots.
- How many real roots may $f(x)$ have?

THE UNIVERSITY of TENNESSEE KNOXVILLE
Big Orange. Big Ideas.

Roots of Equations

- A function $f(x)$ can have any number of real roots or none at all.
- Examples:

$$f(x)=\sin(x)-x=0$$

$$f(x)=\tan(x)-x=0$$

- Root-finding methods are iterative in nature, i.e., they require a starting point or guess.
- Common approach: <u>bracket</u> the root estimate.

THE UNIVERSITY of TENNESSEE KNOXVILLE
Big Orange. Big Ideas.

§4.2 Incremental Search Method

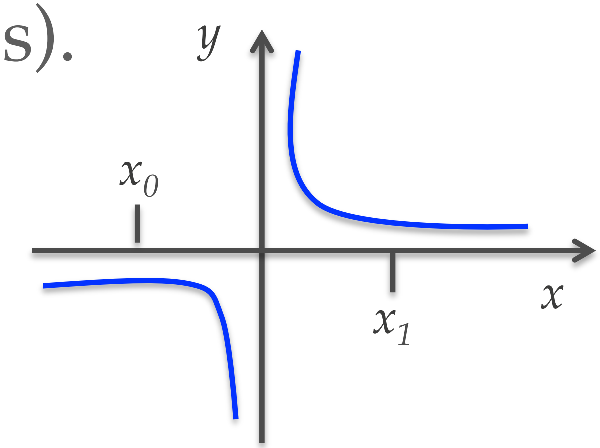
- Idea: if $f(x_1)$ and $f(x_2)$ have opposite signs, then there is at least one root in (x_1,x_2).
- Continue to evaluate $f(x)$ at increments of x (say Δx) for which there is a sign change in $f(x)$.
- Caveats to this approach?

THE UNIVERSITY of TENNESSEE KNOXVILLE
Big Orange. Big Ideas.

Incremental Search Method

- Possible to miss close roots if search increment Δx is larger than the root separation.
- Will no detect a double root.
- Singularities of $f(x)$ could mistakenly taken as roots (e.g., asymptotes).

$f(x_0)$ is negative and $f(x_1)$ is positive but $x=0$ is not a root.



THE UNIVERSITY of TENNESSEE UT KNOXVILLE
Big Orange. Big Ideas.

Incremental Search Method

- Review `rootsearch.py` on p. 147; searches for a zero of the user-supplied function $f(x)$ in the interval (a,b).
- Returns bounds (x_1,x_2) of root or “None”; dx= Δx

THE UNIVERSITY of TENNESSEE UT KNOXVILLE
Big Orange. Big Ideas.

§4.3 Bisection

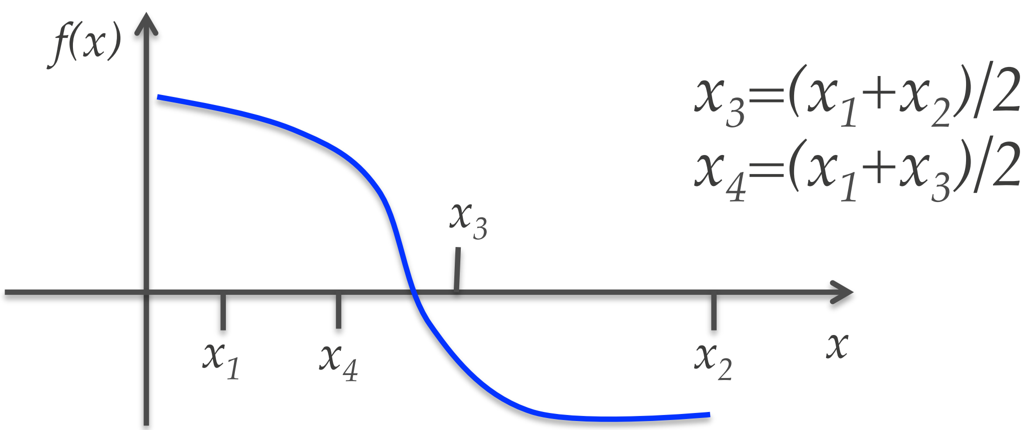
- Interval halving method that is very reliable (but not very fast).
- <u>Idea</u>: If there is a root in (x_1,x_2), then $f(x_1)\times f(x_2)<0$. Compute $f(x_3)$, where $x_3=(x_1+x_2)/2$ and test if $f(x_2)\times f(x_2)<0$?
- Repeat until the current interval (bounding the root) has a very small length, i.e., $|x_2-x_1|\leq \varepsilon$.

THE UNIVERSITY of TENNESSEE UT KNOXVILLE
Big Orange. Big Ideas.

Bisection

- Number of iterations to converge given an original interval (a,b) is $n = \dfrac{\ln(|\Delta x|/\varepsilon)}{\ln 2}$, $\Delta x = b - a$.

See **`bisect.py`** on p.149 and Example 4.3 code on pp. 150-151.



THE UNIVERSITY of TENNESSEE KNOXVILLE
Big Orange. Big Ideas.

§4.5 Newton-Raphson Method

- Best method for root-finding: simple and fast.
- <u>Caveat</u>: requires the derivative $f'(x)$ as well as the function $f(x)$.
- Derivation is from Taylor Series expansion of $f(x)$ about x: $\mathrm{f}(\mathrm{x}_{i+1}) = \mathrm{f}(\mathrm{x}_i) + \mathrm{f}'(\mathrm{x}_i)(\mathrm{x}_{i+1} - \mathrm{x}_i) + \mathrm{O}(\mathrm{x}_{i+1} - \mathrm{x}_i)^2.$
- If x_{i+1} is a root of $f(x)$ then the equation above becomes: $0 = \mathrm{f}(\mathrm{x}_i) + \mathrm{f}'(\mathrm{x}_i)(\mathrm{x}_{i+1} - \mathrm{x}_i) + \mathrm{O}(\mathrm{x}_{i+1} - \mathrm{x}_i)^2.$

THE UNIVERSITY of TENNESSEE UT KNOXVILLE
Big Orange. Big Ideas.

Newton-Raphson Method

- Assuming x_i is close to x_{i+1}, we can simplify the equation further: $0 = f(x_i) + f'(x_i)(x_{i+1} - x_i)$, or $0 = f(x_i) + f'(x_i)x_{i+1} - f'(x_i)x_i$.
- Newton-Raphson Formula: $x_{i+1} = x_i - \dfrac{f(x_i)}{f'(x_i)}$.
- If x is the <u>exact root</u> that x_i approximates, then the error in x_i is $E_i = x - x_i$, and we can represent the error in x_{i+1} as …

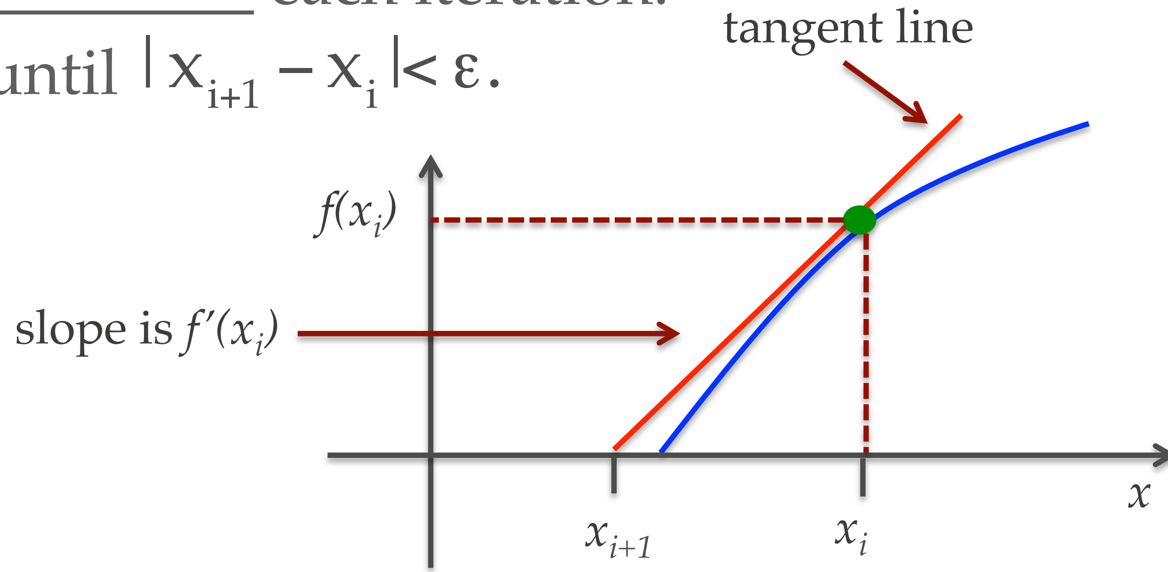
THE UNIVERSITY of TENNESSEE KNOXVILLE
Big Orange. Big Ideas.

Newton-Raphson Method

- Error in x_{i+1} is defined by: $E_{i+1} = \dfrac{f''(x_i)}{2f'(x_i)} E_i^2$.

  Converges quadratically, i.e.,
  the number of significant digits in the current approximation ___________ each iteration.

  Typically iterate until $|x_{i+1} - x_i| < \varepsilon$.



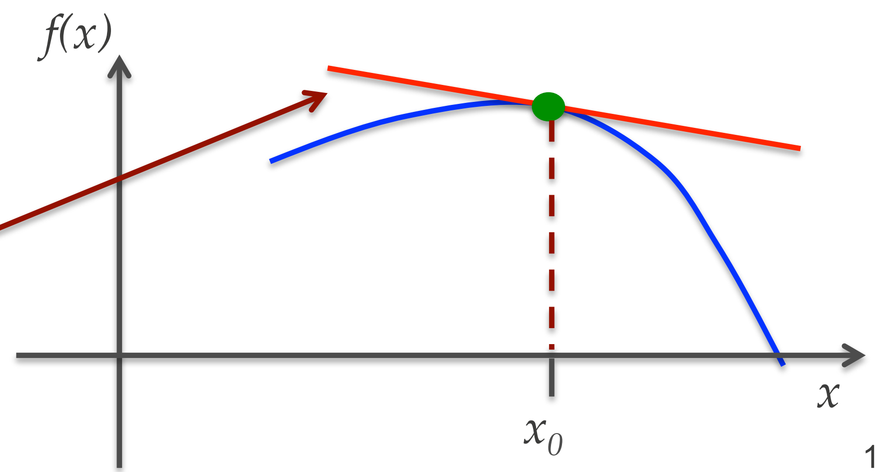
THE UNIVERSITY of TENNESSEE UT KNOXVILLE
Big Orange. Big Ideas.

Newton-Raphson Method

- Algorithm:
  1. Determine initial guess x for root of $f(x)=0$;
  2. Compute $\Delta x = f(x) / f'(x)$;
  3. Set $x = x + \Delta x$, repeat Steps 2 and 3 until $|\Delta x| < \varepsilon$.

Case when N-R diverges (does not converge):

Tangent line is not a good approx. to $f(x)$



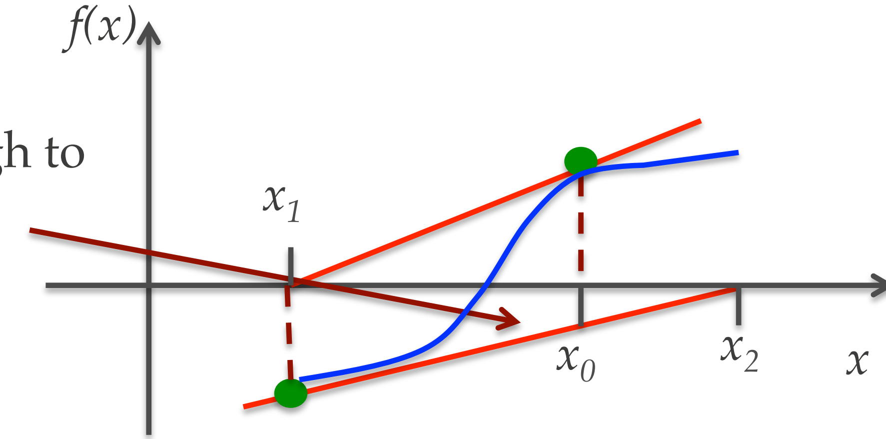
THE UNIVERSITY of TENNESSEE UT KNOXVILLE
Big Orange. Big Ideas.

Newton-Raphson Method

- See `newtonRaphson.py` on pp. 158-159 in textbook. Assumes an initial interval (a,b) for the root; if it *wanders* outside of (a,b), the bisection method is used to get closer to root (hybrid method).

Another case when N-R diverges (does not converge):

If x_0 is not close enough to the root, N-R may not converge (globally).



THE UNIVERSITY of TENNESSEE UT KNOXVILLE
Big Orange. Big Ideas.

Newton-Raphson Method

- See also Example 4.7 (`example4_7.py`) on pp. 159-160; find the smallest zero of

$$f(x)=x^4-6.4x^3+6.45x^2+20.538x-31.752$$

(should be 2.1)

§4.6 Systems of Equations

- Consider *n* simultaneous nonlinear equations:

$$\vec{f}(\vec{x}) = \vec{0} \quad \begin{cases} f_1(x_1, x_2, \ldots, x_n) = 0 \\ f_2(x_1, x_2, \ldots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) = 0 \end{cases}$$

Need to extend N-R to solve for $\vec{x}^T = (x_1, x_2, \ldots, x_n)$.

THE UNIVERSITY of TENNESSEE KNOXVILLE

Big Orange. Big Ideas.

§4.6 Systems of Equations

- Consider *n* simultaneous nonlinear equations:

$$\vec{f}(\vec{x}) = \vec{0} \quad \left\{ \begin{array}{l} f_1(x_1, x_2, \ldots, x_n) = 0 \\ f_2(x_1, x_2, \ldots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) = 0 \end{array} \right.$$

Need to extend N-R to solve for $\vec{x}^T = (x_1, x_2, \ldots, x_n)$.

THE UNIVERSITY of TENNESSEE UT KNOXVILLE
Big Orange. Big Ideas.

Systems of Equations

- Taylor Series expansion of $f_i(\vec{x})$ about $\vec{x}$ is

$$f_i(\vec{x}+\Delta\vec{x}) = f_i(\vec{x}) + \sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j}\Delta\vec{x}_j + O(\Delta\vec{x}^2).$$

Dropping the last term yields:

$$f_i(\vec{x}+\Delta\vec{x}) = f_i(\vec{x}) + J(\vec{x})\Delta\vec{x},$$

where $J(\vec{x})$, $J_{ij} = \dfrac{\partial f_i}{\partial x_j}$, is the *n*-by-*n* Jacobian matrix.

THE UNIVERSITY of TENNESSEE UT KNOXVILLE

Big Orange. Big Ideas.

Systems of Equations

- Assume that $\vec{x}$ is the current approx. to the solution of $\vec{f}(\vec{x}) = \vec{0}$, and let $\vec{x} + \Delta\vec{x}$ be an improved approximation; how do we determine $\Delta\vec{x}$?

$$\text{Set } \vec{f}(\vec{x} + \Delta\vec{x}) = \vec{0} \rightarrow J(\vec{x})\Delta\vec{x} = -\vec{f}(\vec{x}).$$

THE UNIVERSITY of TENNESSEE UT KNOXVILLE
Big Orange. Big Ideas.

Systems of Equations

- N-R Method for solving a nonlinear system of equations:
  1. Estimate $\vec{x}$
  2. Evaluate $\vec{f}(\vec{x})$
  3. Compute $J(\vec{x})$
  4. Solve $J(\vec{x})\Delta\vec{x} = -\vec{f}(\vec{x})$ for $\Delta\vec{x}$
  5. Set $\vec{x} = \vec{x} + \Delta\vec{x}$ and repeat Steps 2—5.

Stopping criterion: $\left\|\Delta\vec{x}\right\|_2 < \varepsilon$.

THE UNIVERSITY of TENNESSEE UT KNOXVILLE
Big Orange. Big Ideas.

Systems of Equations

- What happens if $\dfrac{\partial f_i}{\partial x_j}$ is difficult to impractical to compute?

Can approximate it using finite differences:

$$\frac{\partial f_i}{\partial x_j} \approx \frac{f_i(\vec{x}+\vec{e}_j h) - f_i(\vec{x})}{h}, \vec{e}_j = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix} \longleftarrow \text{jth component}$$

Systems of Equations

- See **`newtonRaphson2.py`** on p. 163 for code to implement the N-R method for nonlinear systems of equations; for an example of its use see the **`Example4_9.py`** code (pp. 165-166).

  Two interesting applications of the N-R method are found in Problems 16 (p. 167, structural mechanics) and 26 (pp. 170-171, circle drawing).

THE UNIVERSITY of TENNESSEE KNOXVILLE
Big Orange. Big Ideas.