# Advanced Statistical Computing

Dr. Ke Deng
Center for statistical Science
Tsinghua University, Beijing

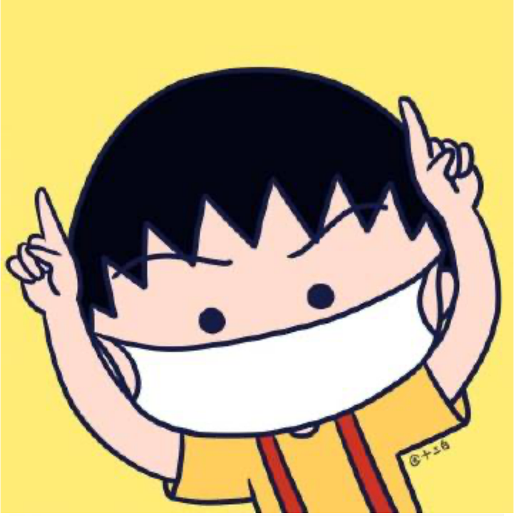
邓 柯
清华大学统计学研究中心

kdeng@tsinghua.edu.cn

# 融合式教学安排

| | 线下课堂 | 腾讯会议 | 网络学堂 | 微信群 |
|---|---|---|---|---|
| 发布公告 | | | | √ |
| 课件下载 | | | √ | |
| 直播视频 | | √ | | |
| 直播音频 | | √ | | |
| 课堂互动 | √ | | | |
| 课程作业 | | | √ | |
| 课后答疑 | √ | √ | | |

# 请在教室佩戴口罩

Advanced Statistical Computing
Lecture 0

# Introduction & Preliminary Knowledge

Dr. Ke Deng
Center for statistical Science
Tsinghua University, Beijing

邓 柯
清华大学统计学研究中心

kdeng@tsinghua.edu.cn

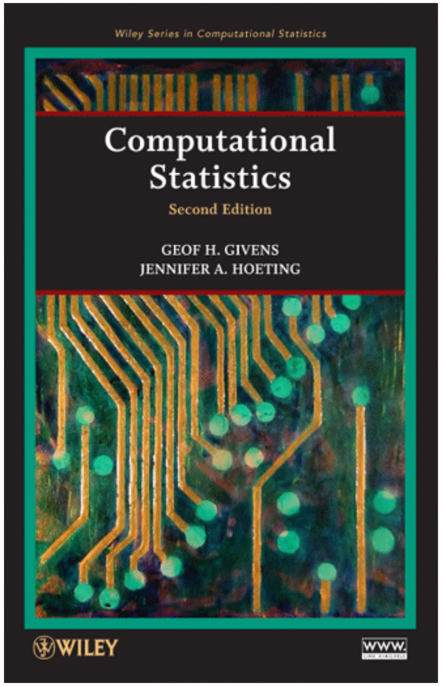
# Course Rules

| | | | |
|---|---|---|---|
| Homework | 30% | Class Participation | 5% |
| Midterm Exam | 30% | Final Exam | 35% |

5 "random" roll calls in the semester, 1 point each time

# Textbooks & Online Materials

[T1] Geof Givens & Jennifer Hoeting (2013) *Computational Statistics* (2nd Edition), Wiley. (http://onlinelibrary.wiley.com/book/10.1002/9781118555552)




**Online Materials, Data Sets & Programs**

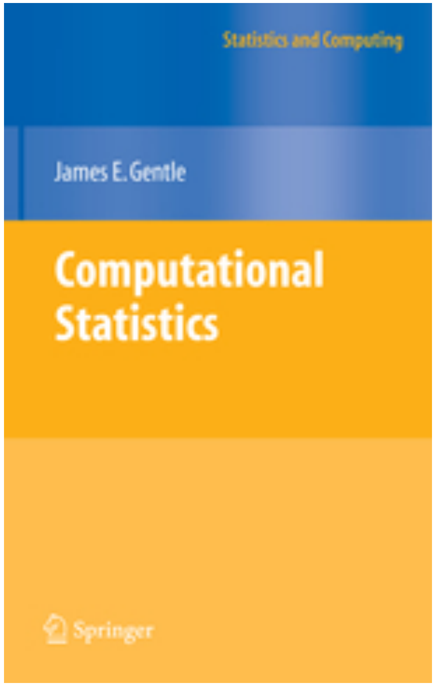
[O1] http://www.stat.colostate.edu/computationalstatistics/
[O2] http://statweb.stanford.edu/ susan/courses/s227/
[O3] http://www.stat.purdue.edu/chuanhai/teaching/Stat598D/

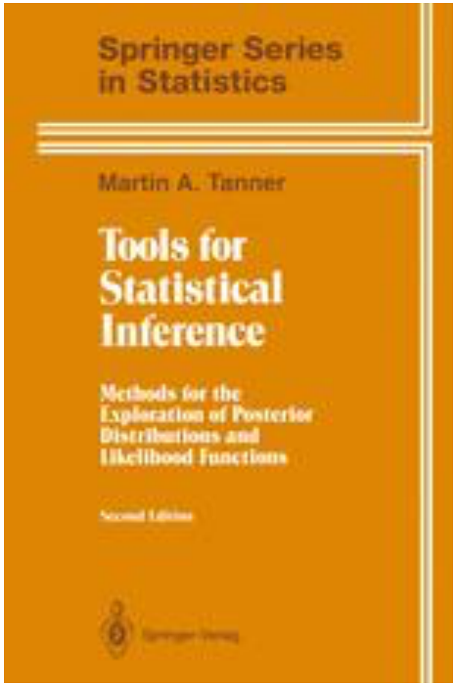
# Reference books

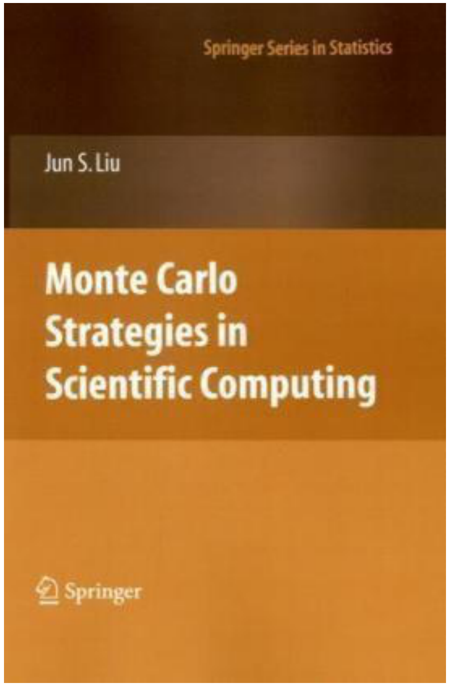
[R1] James E. Gentle (2009) *Computational Statistics*, Springer.

[R2] Martin A. Tanner (1996) *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions*, Springer.

[R3] Jun S. Liu (2001) *Monte Carlo Strategies in Scientific Computing*, Springer.



# Major Topics of the Course

- Optimization
  - Solving nonlinear equations
  - Combinational optimization
  - EM algorithm
- Integration & Monte Carlo Simulation
  - Numerical integration
  - Monte Carlo integration
  - Markov chain Monte Carlo
- Bootstrapping
- Density Estimation & Smoothing
- Practical Techniques

# Preliminary Knowledge

- Calculus
- Probability distributions & the exponential family
- Likelihood inference
- Bayesian inference
- Statistical limit theorem
- Markov chain
- Basic concepts of computing
- Programming languages for computing
  - **R**, MATLAB, C++, …

# "Big oh" & "Little oh"

- Problem setting:
  - Let $f$ and $g$ be functions defined on a common (possibly infinite) interval.
  - Let $z_0$ be a point in this interval or a boundary point of.
  - We require $g(z) \neq 0$ for all $z \neq z_0$ in a neighborhood of $z_0$.

- We say $f(z) = \mathcal{O}(g(z))$, if there exists a constant $M$ such that
$$|f(z)| \leq M|g(z)| \; as \; z \to z_0.$$
  - For example, $(n+1)/(3n^2) \; = \; \mathcal{O}(n-1)$ as $n \to \infty$.

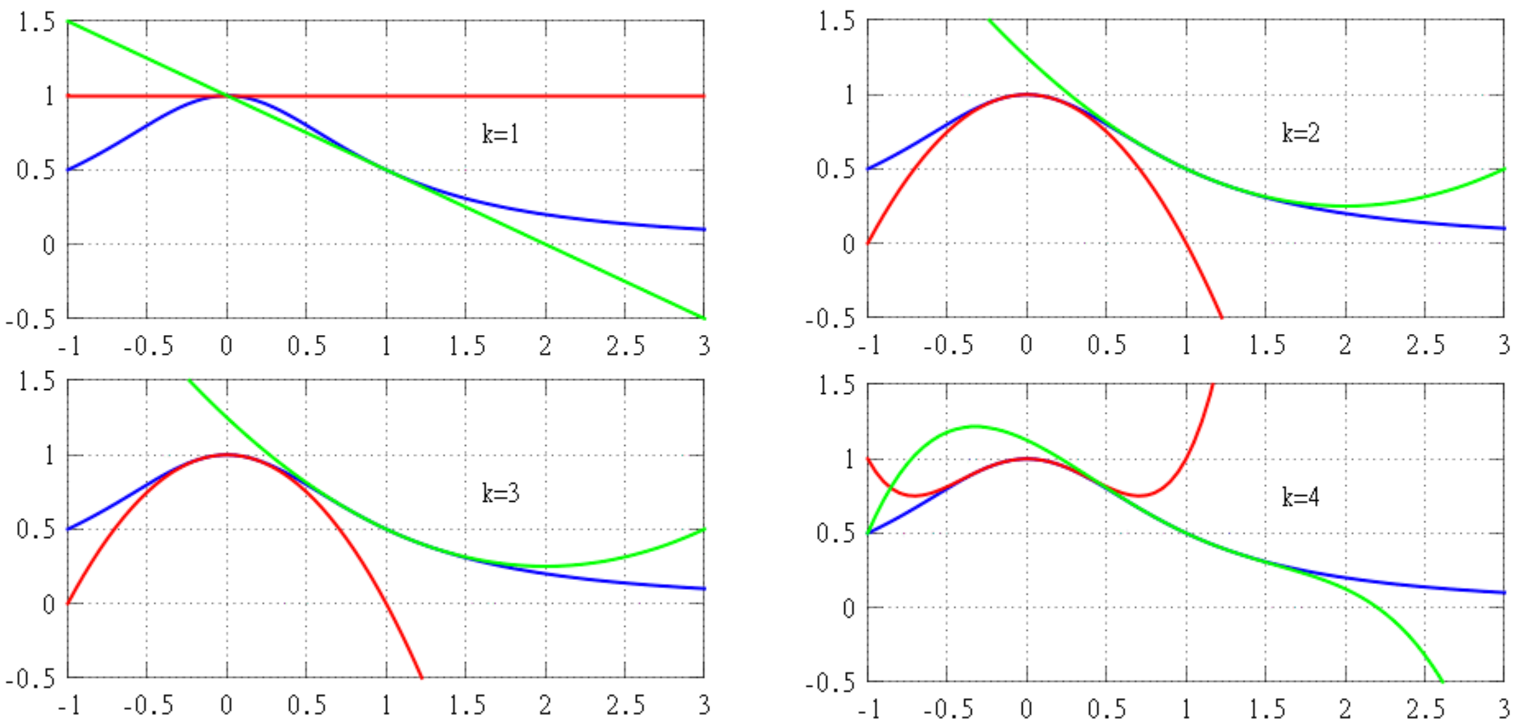
- we say $f(z) = o(g(z))$, if $\lim_{z \to z_0} f(z)/g(z) = 0$.
  - For example, $f(x_0 + h) - f(x_0) = hf'(x_0) + o(h)$ as $h \to 0$ if $f$ is differentiable at $x_0$.

Note: the same notation can be used for describing the convergence of a sequence $\{x_n\}$ as $n \to \infty$, by letting $f(n) = x_n$.

# Taylor Polynomial

Let $k \geq 1$ be an integer and let the function $f: \mathbf{R} \rightarrow \mathbf{R}$ be $k$ times differentiable at the point $a \in \mathbf{R}$. The Taylor polynomial of $f$ at $a$ is defined as:

$$P_k(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(k)}(a)}{k!}(x-a)^k$$




Approximation of $f(x) = 1/(1 + x^2)$ by its Taylor polynomials at $x = 0$ and $x = 1$

# Remainder Term of Taylor Polynomial

❖ Remainder term: $R_k(x) = f(x) - P_k(x)$

- Peano form: $R_k(x) = o(|x-a|^k), \quad x \to a.$

Let $h_k(x) = \begin{cases} \frac{f(x)-P(x)}{(x-a)^k} & x \neq a \\ 0 & x = a \end{cases}$

It's easy to show $\lim_{x \to a} h_k(x) = 0$ by repeated application of L'Hôpital's rule

# Remainder Term of Taylor Polynomial

❖ Remainder term: $R_k(x) = f(x) - P_k(x)$

- Peano form: $R_k(x) = o(|x-a|^k), \quad x \to a.$
- Mean-value forms: $R_k(x) = \dfrac{f^{(k+1)}(\xi)}{k!}(x-\xi)^k \dfrac{G(x)-G(a)}{G'(\xi)}$

Let $F(t) = f(t) + f'(t)(x-t) + \dfrac{f''(t)}{2!}(x-t)^2 + \cdots + \dfrac{f^{(k)}(t)}{k!}(x-t)^k$

$F(x) = f(x) \qquad F(a) = P_k(x)$

$R_k(x) = f(x) - P_k(x) = F(x) - f(a)$

$F'(t) = \dfrac{f^{(k+1)}(t)}{k!}(x-t)^k$

$$\frac{R_k(x)}{G(x)-G(a)} = \frac{F(x)-F(a)}{G(x)-G(a)} = \frac{F'(\xi)}{G'(\xi)}$$

$$R_k(x) = \frac{f^{(k+1)}(\xi)}{k!}(x-\xi)^k \frac{G(x)-G(a)}{G'(\xi)}$$

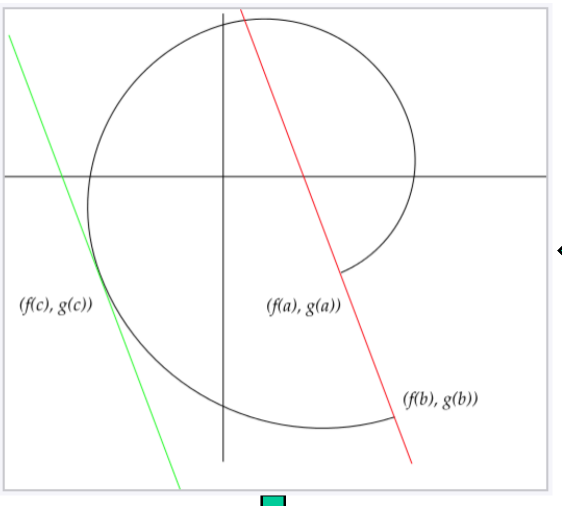
Cauchy's intermediate value theorem

For $\forall F, G \in C_1[a,x]$, $\dfrac{F'(\xi)}{G'(\xi)} = \dfrac{F(x)-F(a)}{G(x)-G(a)}$ → $\xi \in (a,x)$

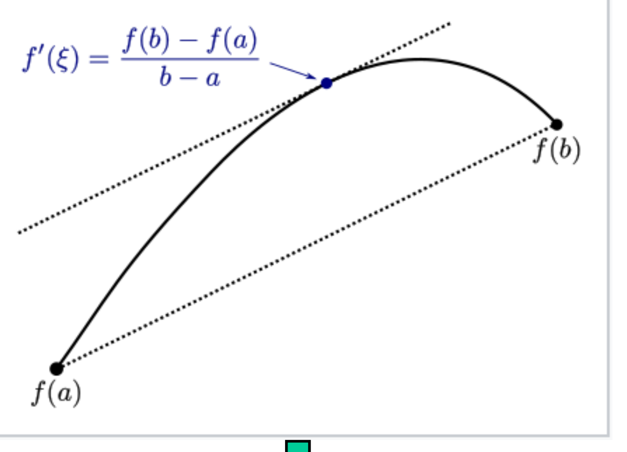
# Remainder Term of Taylor Polynomial

❖ Remainder term: $R_k(x) = f(x) - P_k(x)$

- Peano form: $R_k(x) = o(|x-a|^k), \quad x \to a.$
- Mean-value forms: $R_k(x) = \dfrac{f^{(k+1)}(\xi)}{k!}(x-\xi)^k \dfrac{G(x)-G(a)}{G'(\xi)}$




generalize




Cauchy's intermediate value theorem

For $\forall F, G \in C_1[a, x]$, $\dfrac{F'(\xi)}{G'(\xi)} = \dfrac{F(x)-F(a)}{G(x)-G(a)}$

Intermediate value theorem

For $\forall f \in C_1[a, x], f'(\xi) = \dfrac{f(b)-f(a)}{b-a}$

# Remainder Term of Taylor Polynomial

❖ Remainder term: $R_k(x) = f(x) - P_k(x)$

- Peano form: $R_k(x) = o(|x-a|^k), \quad x \to a.$

- Mean-value forms: $R_k(x) = \dfrac{f^{(k+1)}(\xi)}{k!}(x-\xi)^k \dfrac{G(x)-G(a)}{G'(\xi)}$

$G(t) = (t-x)^{k+1}$

  - Lagrange form: $R_k(x) = \dfrac{f^{(k+1)}(\xi_L)}{(k+1)!}(x-a)^{k+1}$

$G(t) = t - a$

  - Cauchy form: $R_k(x) = \dfrac{f^{(k+1)}(\xi_C)}{k!}(x-\xi_C)^k(x-a)$

# Remainder Term of Taylor Polynomial

- Remainder term: $R_k(x) = f(x) - P_k(x)$

  - Peano form: $R_k(x) = o(|x-a|^k), \quad x \to a.$
  - Mean-value forms: $R_k(x) = \dfrac{f^{(k+1)}(\xi)}{k!}(x-\xi)^k \dfrac{G(x)-G(a)}{G'(\xi)}$
    - Lagrange form: $R_k(x) = \dfrac{f^{(k+1)}(\xi_L)}{(k+1)!}(x-a)^{k+1}$
    - Cauchy form: $R_k(x) = \dfrac{f^{(k+1)}(\xi_C)}{k!}(x-\xi_C)^k(x-a)$
  - Integral form: $R_k(x) = \displaystyle\int_a^x \frac{f^{(k+1)}(t)}{k!}(x-t)^k\,dt$

# Taylor's Theorem

*Taylor' s theorem* provides a polynomial approximation to a function $f$:
Suppose $f$ has finite $(n+1)$th derivative on $(a,b)$ and continuous $n$th derivative on $[a,b]$. Then for any $x_0 \in [a,b]$ distinct from $x$, the Taylor series expansion of $f$ about $x_0$ is

$$f(x) = \sum_{i=0}^{n} \frac{1}{i!} f^{(i)}(x_0)(x-x_0)^i + R_n$$

where $f^{(i)}(x_0)$ is the $i$th derivative of $f$ evaluated at $x_0$, and

$$R_n = \frac{1}{(n+1)!} f^{(n+1)}(\xi)(x-x_0)^{n+1}$$

for some point $\xi$ in the interval between $x$ and $x_0$.

Note that: $R_n = \mathcal{O}(|x-x_0|^{n+1})$ as $|x-x_0| \to 0$.

# Taylor's Theorem (Multivariate version)

Suppose $f$ is a real-valued function of a $p$-dimensional variable $\boldsymbol{x}$, possessing continuous partial derivatives of all orders up to and including $n+1$ with respect to all coordinates, in an open convex set containing $\boldsymbol{x}$ and $\boldsymbol{x}_0 \neq \boldsymbol{x}$. Then

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \sum_{i=1}^{n} \frac{1}{i!} D^{(i)}(f; \mathbf{x}_0, \mathbf{x} - \mathbf{x}_0) + R_n,$$

where

$$D^{(i)}(f; \mathbf{x}, \mathbf{y}) = \sum_{j_1=1}^{p} \cdots \sum_{j_i=1}^{p} \left\{ \left( \frac{d^i}{dt_{j_1} \cdots dt_{j_i}} f(\mathbf{t}) \Big|_{\mathbf{t}=\mathbf{x}} \right) \prod_{k=1}^{i} y_{j_k} \right\}$$

$$R_n = \frac{1}{(n+1)!} D^{(n+1)}(f; \boldsymbol{\xi}, \mathbf{x} - \mathbf{x}_0)$$

for some point $\xi$ in the interval between $x$ and $x_0$.

Note that: $R_n = \mathcal{O}(|x - x_0|^{n+1})$ as $|x - x_0| \to 0$.

# Reference

- Reference: [T1] Chapter 1
- Further reading:
  - [R1] Chapter 1: Mathematical & Statistical Preliminaries

Advanced Statistical Computing
Lecture 1

# Optimization & Solving Nonlinear Equation (I)

Dr. Ke Deng
Center for statistical Science
Tsinghua University, Beijing

邓 柯
清华大学统计学研究中心

kdeng@tsinghua.edu.cn

# Major Topics of the Course

- Optimization
  - Solving nonlinear equations
  - Combinational optimization
  - EM algorithm
- Integration & Monte Carlo Simulation
  - Numerical integration
  - Monte Carlo integration
  - Markov chain Monte Carlo
- Bootstrapping
- Density Estimation & Smoothing
- Practical Techniques

# Optimization in Statistical Inference

- MLE : central to statistical inference
- Minimizing risk: in a Bayesian decision problem
- Solving nonlinear least squares problems
- Finding highest posterior density intervals
- …

Finding MLE by solving the score equation below:

$$\mathbf{l}'(\boldsymbol{\theta}) = \left(\frac{dl(\boldsymbol{\theta})}{d\theta_1}, \ldots, \frac{dl(\boldsymbol{\theta})}{d\theta_n}\right)^{\mathrm{T}} = \mathbf{0},$$

**Optimization** is intimately linked with **solving (non-linear) equations**

# Several Typical Scenarios

- Score equation with a straightforward analytic solution
  - Very rare in practice
- Linear score equation
  - Easy to solve
- Linear objective function with linear inequality constraints
  - More difficult, but can be solved by linear programming techniques
  - such as, the simplex method or interior point methods
- Nonlinear score equation with no analytic solution
  - optima are routinely found using a variety of effective off-the-shelf numerical optimization software

# Motivation of Study

A good question:

It seems like optimization is a solved problem whose study here might be a low priority, why do not we simply omit it?

Reasons for a careful study of this topic here:

- Optimization software is confronted with a new problem every time the user presents a new function to be optimized
- Even the best optimization software often initially fails to find the maximum for tricky likelihoods and requires tinkering to succeed
- Therefore, the user must understand enough about how optimization works to tune the procedure successfully

# Major Topics of the Course

- Optimization
  - Solving nonlinear equations
  - Combinational optimization
  - EM algorithm
- Integration & Monte Carlo Simulation
  - Numerical integration
  - Monte Carlo integration
  - Markov chain Monte Carlo
- Bootstrapping
- Density Estimation & Smoothing
- Practical Techniques

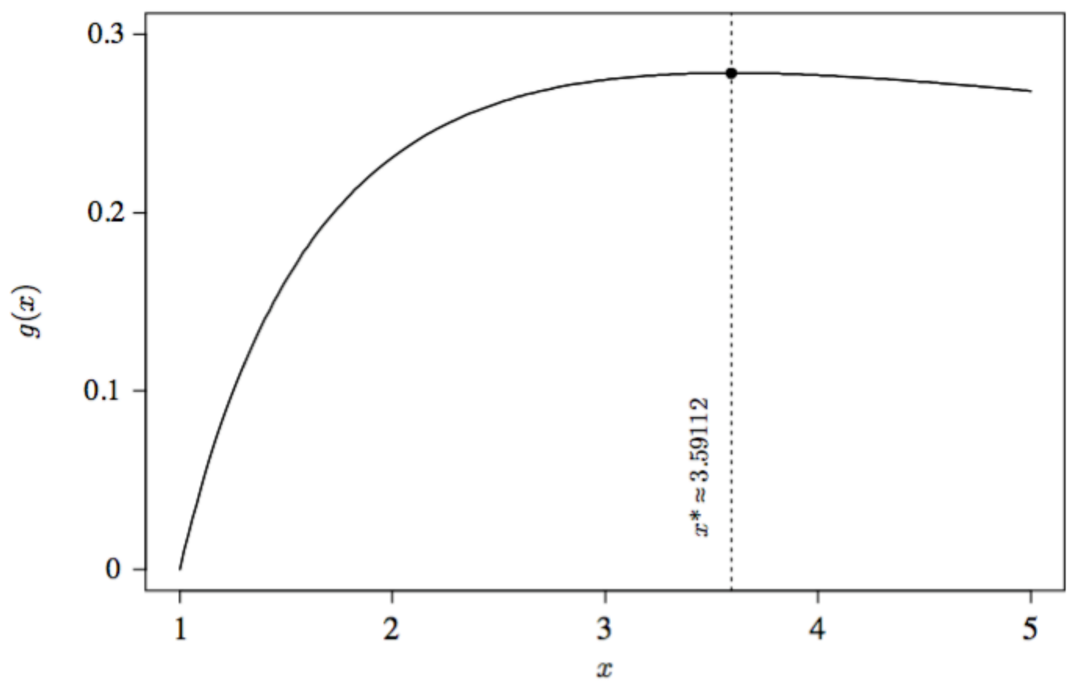
# Univariate Optimization Problem

❖ An example:

Target function to maximize: $g(x) = \dfrac{\log x}{1+x}$ ⟹ $g'(x) = \dfrac{1 + 1/x - \log x}{(1+x)^2} = 0$

no analytic solution

Fig. 2.1 The maximum occurs at $x^* \approx 3.59112$, indicated by the vertical line.




- Numerical optimization is in a great appeal
- Graphing $g(x)$ is often the first and very useful step
- We see that the maximum is around 3

# Three Primary Methods

- $M_1$: Bisection/Bracketing Method
- $M_2$: Newton's Method
  - $M_2'$ : Fisher scoring (special case of $M_2$)
  - $M_2''$: Secant method (approximation of $M_2$)
- $M_3$: Fixed-Point/Functional Iteration
  - $M_3'$ : Fixed-point method with naïve updating function
  - $M_3''$: Fixed-point method with scaled updating function

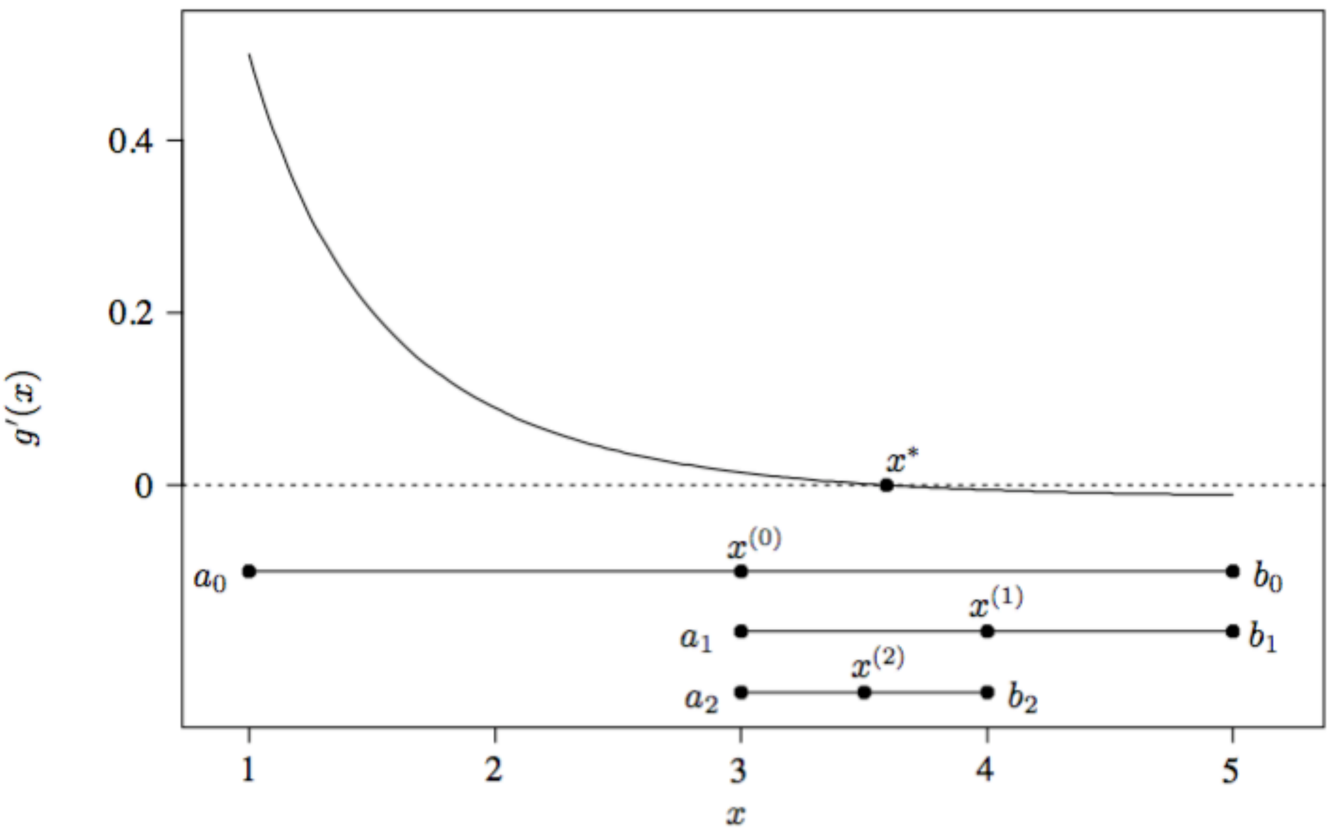
# $M_1$: Bisection Method

Target function:

$$g(x) = \frac{\log x}{1+x} \quad \Rightarrow \quad g'(x) = \frac{1 + 1/x - \log x}{(1+x)^2}$$

Starting values:

$$a_0 = 1, b_0 = 5, \text{and } x^{(0)} = 3$$



# $M_1$: Bisection Method

Suppose:

- $g'$ is continuous on $[a_0, b_0]$
- $g'(a_0)g'(b_0) \leq 0$

$\xrightarrow[\text{theorem}]{\text{intermediate value}}$

- there exists at least one $x^*$ in $[a_0, b_0]$ for which $g'(x^*) = 0$
- hence $x^*$ is a local optimum of $g$

Bisection method constructs a sequence of nested intervals to capture $x^*$:

$$[a_0, b_0] \supset [a_1, b_1] \supset [a_2, b_2] \supset \cdots \text{ and so forth}$$

Starting value: $x^{(0)} = (a_0 + b_0)/2$

Updating equations: $$[a_{t+1}, b_{t+1}] = \begin{cases} [a_t, x^{(t)}] & \text{if } g'(a_t)g'(x^{(t)}) \leq 0, \\ [x^{(t)}, b_t] & \text{if } g'(a_t)g'(x^{(t)}) > 0 \end{cases}$$

$$x^{(t)} = \frac{1}{2}(a_t + b_t) \Longleftrightarrow x^{(t)} = a_t + (b_t - a_t)/2$$

numerically more stable

**Note:** if $g$ has more than one root in the starting interval, it is easy to see that bisection will find one of them, but will not find the rest.

# Stopping Rule

- Two reasons to stop:
  - if the procedure appears to have achieved satisfactory convergence
  - or, if it appears unlikely to do so soon
- Two basic stopping criterion:
  - absolute convergence criterion

$$\left|x^{(t+1)} - x^{(t)}\right| < \epsilon$$

Tolerable imprecision

  - relative convergence criterion

$$\frac{\left|x^{(t+1)} - x^{(t)}\right|}{\left|x^{(t)}\right|} < \epsilon \quad \text{or} \quad \frac{\left|x^{(t+1)} - x^{(t)}\right|}{\left|x^{(t)}\right| + \epsilon} < \epsilon$$

Numerically more stable if $x^{(t)}$ is too close to zero

# Guarantee of Convergence in Theory

- Precision of bisection method at the *t*th step:

$$b_t - a_t = 2^{-t}(b_0 - a_0) \longrightarrow 0 \text{ when } n \longrightarrow \infty$$

- When $g'$ is continuous, the method ensures that $g'(a_t)g'(b_t) \leq 0$
- continuity therefore implies that $g'\big(x(\infty)\big)^2 \leq 0$, thus $g'\big(x(\infty)\big) = 0$
- Therefore, $x(\infty)$ must be a root of $g$, and the bisection method is guaranteed to converge to a root in $[a_0, b_0]$.

Note that:
In practice, numerical imprecision in a computer may thwart convergence.

# Practical Issues

- The outcome depends on:
  - $g$, the starting value, and the optimization algorithm tried
- A bad starting value can lead to
  - divergence, cycling, discovery of a misleading local optimum, …
- Find a good starting value near the global optimum by
  - graphing, preliminary estimates (e.g., method-of-moments estimates), educated guesses, and trial and error.
- If computing speed limits the total number of iterations
  - It is wise **not** to use one long run of the optimization procedure
  - A collection of runs from multiple starting values are preferred

> ➢ gain **confidence** in your result
>
> ➢ **avoid** being fooled by local optima or stymied by convergence failure.

# Extensions of Bisection

- Bisection method is an example of a bracketing method
  - a method that bounds a root within a sequence of nested intervals of decreasing length
- Bisection is quite a slow approach
  - It requires a rather large number of iterations to achieve a desired precision, relative to other methods discussed below
- Other bracketing methods include
  - Secant bracket ----→ equally slow after an initial period of greater efficiency
  - Illinois method, Ridders's method and Brent's method ----→ faster

# Advantages of Bracketing Method

Despite relatively slow, bracketing methods have advantages below:

- Perform robustly on most problems:
  - guarantee a root can be found as long as $g'$ is continuous on $[a_0, b_0]$.
- Avoid worries about $g''$:
  - the existence, behavior, or ease of deriving $g''$

Bracketing methods continue to be reasonable alternatives to the methods below that rely on greater smoothness of $g$.

Note: Although bisection is very simple in nature, it illustrates the main components of all iterative root-finding procedures.

# $M_2$: Newton's Method

❖ The motivating example: 

Score equation with no analytic solution

Target function to maximize: $g(x) = \dfrac{\log x}{1+x} \Longrightarrow g'(x) = \dfrac{1 + 1/x - \log x}{(1+x)^2} = 0$

- The key idea:
  - Approximated nonlinear score equation with a linear equation nearby $x^{(t)}$ :

    $0 = g'(x^*) \approx g'(x^{(t)}) + (x^* - x^{(t)})g''(x^{(t)})$

  - Get solution: $x^* = x^{(t)} - \dfrac{g'(x^{(t)})}{g''(x^{(t)})}$

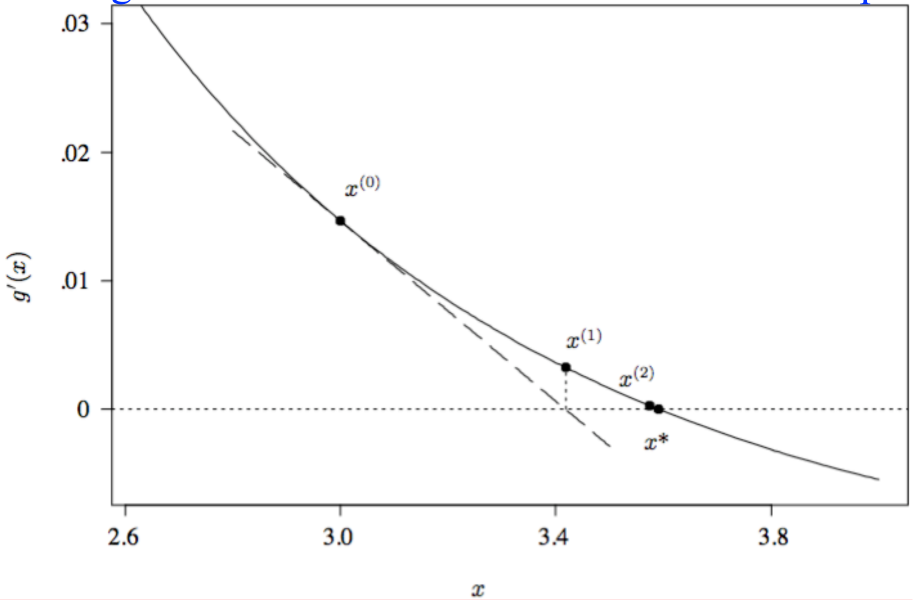
- Updating equation:

$$x^{(t+1)} = x^{(t)} - \frac{g'(x^{(t)})}{g''(x^{(t)})} = x^{(t)} + h^{(t)}$$

$-\dfrac{g'(x^{(t)})}{g''(x^{(t)})}$

- Technical conditions:
  - $g'$ is continuously differentiable
  - $g''(x^*) \neq 0$

Fig. 2.3 Newton's method for the above example



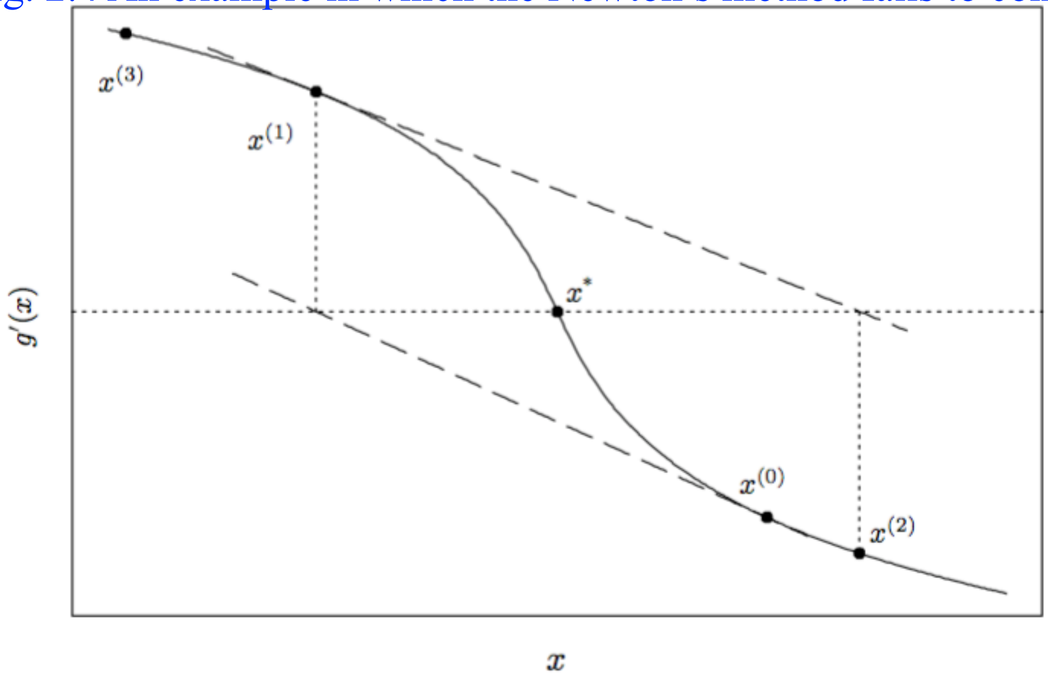

$h^{(t)} = \dfrac{(x^{(t)}+1)\left(1 + 1/x^{(t)} - \log x^{(t)}\right)}{3 + 4/x^{(t)} + 1/(x^{(t)})^2 - 2\log x^{(t)}}$ for the motivating example

# Newton's Method may Fail to Converge

Whether Newton's method converges depends on

- the shape of *g*
- the starting value

Fig. 2.4 An example in which the Newton's method fails to converge



# Convergence of Newton's Method

- Given the two technical conditions for Newton's method to work:
  - ➢ $g'$ is continuously differentiable
  - ➢ $g''(x^*) \neq 0$

- The 1st order Taylor expansion of $g'$ nearby $x^{(t)}$:

Require existence of $g'''$

$$0 = g'(x^*) = g'(x^{(t)}) + (x^* - x^{(t)})g''(x^{(t)}) + \frac{1}{2}(x^* - x^{(t)})^2 g'''(q)$$

$$\lim_{t\to\infty} \frac{|\epsilon^{(t+1)}|}{|\epsilon^{(t)}|^\beta} = c \ \ \& \ \beta = 2$$

Quadratic convergence

$$\underbrace{x^{(t)} + h^{(t)}}_{x^{(t+1)}} - x^* = (x^* - x^{(t)})^2 \frac{g'''(q)}{2g''(x^{(t)})} \Rightarrow \epsilon^{(t+1)} = (\epsilon^{(t)})^2 \frac{g'''(q)}{2g''(x^{(t)})}$$

- For a neighborhood of $x^*$: $\mathcal{N}_\delta(x^*) = [x^* - \delta, x^* + \delta]$

$$c(\delta) = \max_{x_1, x_2 \in \mathcal{N}_\delta(x^*)} \left| \frac{g'''(x_1)}{2g''(x_2)} \right| \to \left| \frac{g'''(x^*)}{2g''(x^*)} \right| \Rightarrow \left| c(\delta)\epsilon^{(t+1)} \right| \leq \left( c(\delta)\epsilon^{(t)} \right)^2$$

- If the starting value is not too bad, say

$$\left|\epsilon^{(0)}\right| = \left|x^{(0)} - x^*\right| \leq \delta \Rightarrow \left|\epsilon^{(t)}\right| \leq \frac{(c(\delta)\delta)^{2^t}}{c(\delta)} \longrightarrow 0$$

# Convergence of Newton's Method

**Theorem (a).** If $g'''$ is continuous and $x^*$ is a simple root of $g'$, then there exists a neighborhood of $x^*$ for which Newton's method converges to $x^*$ when started from any $x^{(0)}$ in that neighborhood.

**Theorem (b).** If g′ is twice continuously differentiable, is convex, and has a root, then Newton's method converges to the root from *any* starting point.

**Theorem (c).** When starting from somewhere in an interval $[a, b]$, the Newton's method will converge from any $x^{(0)}$ in the interval if

1. $g''(x) \neq 0$ on $[a, b]$,
2. $g'''(x)$ does not change sign on $[a, b]$,
3. $g'(a)g'(b) < 0$, and
4. $\left|g'(a)/g''(a)\right| < b - a$ and $\left|g'(b)/g''(b)\right| < b - a$

# $M_2'$: Fisher Scoring

- A variate of Newton's method in finding MLE
- Updating equation in original Newton's method:

$$x^{(t+1)} = x^{(t)} - \frac{g'(x^{(t)})}{g''(x^{(t)})} \xrightarrow{g\,=\,\text{log–likelihood}\ l} \theta^{(t+1)} = \theta^{(t)} - \frac{l'(\theta^{(t)})}{l''(\theta^{(t)})}$$

- Basic property of Fish Information:

$$-l''(\theta) \longrightarrow \boxed{I(\theta)} \quad \Longrightarrow \quad \boxed{\theta^{(t+1)} = \theta^{(t)} + l'(\theta^{(t)})I(\theta^{(t)})^{-1}}$$

Fisher information

- Alternative updating equation

**Notes**

- Fisher scoring and Newton's method have the same asymptotic properties
- For individual problems one may be computationally/analytically easier than the other
- Generally, Fisher scoring works better in the beginning to make rapid improvements, while Newton's method works better for refinement near the end.

# $M_2''$: Secant Method

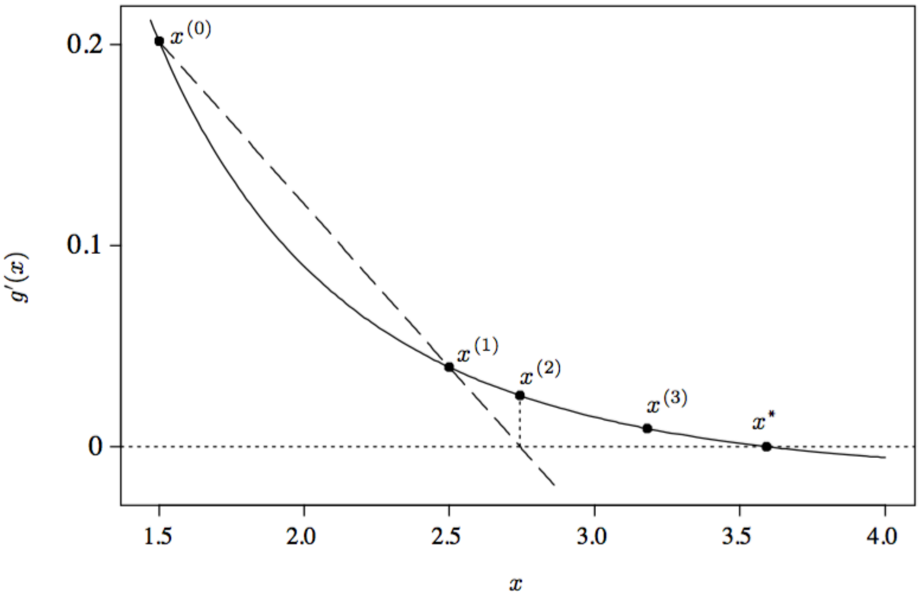
- An approximation of Newton's method when $g''$ is difficult to calculate
- Updating equation in original Newton's method:

$$x^{(t+1)} = x^{(t)} - \frac{g'(x^{(t)})}{g''(x^{(t)})}$$

Replace $g''$ by discrete-difference
$[g'(x^{(t)}) - g'(x^{(t-1)})]/(x^{(t)} - x^{(t-1)})$

- Alternative updating equation

$$x^{(t+1)} = x^{(t)} - g'(x^{(t)}) \frac{x^{(t)} - x^{(t-1)}}{g'(x^{(t)}) - g'(x^{(t-1)})}$$



# Convergence of Secant Method

- Convergence:

  Under conditions akin to those for Newton's method, the secant method will converge to the root $x^*$

- Convergence order

$$\epsilon^{(t+1)} = x^{(t+1)} - x^*$$

convergence with order $\beta$

$$\lim_{t\to\infty} \frac{|\epsilon^{(t+1)}|}{|\epsilon^{(t)}|^\beta} = c$$

**Q1**: what's the convergence order of Secant Method?
**Q2**: does it converge faster or slower than the Newton's?

$$x^{(t+1)} = x^{(t)} - g'(x^{(t)}) \frac{x^{(t)} - x^{(t-1)}}{g'(x^{(t)}) - g'(x^{(t-1)})}$$

$$\epsilon^{(t+1)} = \left[ \frac{x^{(t)} - x^{(t-1)}}{g'(x^{(t)}) - g'(x^{(t-1)})} \right] \left[ \frac{g'(x^{(t)})/\epsilon^{(t)} - g'(x^{(t-1)})/\epsilon^{(t-1)}}{x^{(t)} - x^{(t-1)}} \right] \left[ \epsilon^{(t)} \epsilon^{(t-1)} \right]$$

# Convergence of Secant Method

- Convergence:

  Under conditions akin to those for Newton's method, the secant method will converge to the root $x^*$

- Convergence order

$$\epsilon^{(t+1)} = x^{(t+1)} - x^*$$

convergence with order $\beta$

$$\lim_{t\to\infty} \frac{|\epsilon^{(t+1)}|}{|\epsilon^{(t)}|^\beta} = c$$

**Q1**: what's the convergence order of Secant Method?
**Q2**: does it converge faster or slower than the Newton's?

$$x^{(t+1)} = x^{(t)} - g'(x^{(t)}) \frac{x^{(t)} - x^{(t-1)}}{g'(x^{(t)}) - g'(x^{(t-1)})}$$

$A^{(t)} \rightarrow 1/g''(x^*)$ as $x(t) \rightarrow x^*$ for continuous $g''$

$$\epsilon^{(t+1)} = \left[\frac{x^{(t)} - x^{(t-1)}}{g'(x^{(t)}) - g'(x^{(t-1)})}\right] \left[\frac{g'(x^{(t)})/\epsilon^{(t)} - g'(x^{(t-1)})/\epsilon^{(t-1)}}{x^{(t)} - x^{(t-1)}}\right] \left[\epsilon^{(t)}\epsilon^{(t-1)}\right]$$

2nd order Taylor expansion of $g'(x^{(t)})$ on $x^*$

$$\approx d^{(t)}\epsilon^{(t)}\epsilon^{(t-1)}$$

$$B^{(t)} \approx g'''(x^*) \frac{\epsilon^{(t)} - \epsilon^{(t-1)}}{2(x^{(t)} - x^{(t-1)})} = \frac{g'''(x^*)}{2}$$

$$d^{(t)} \rightarrow \frac{g'''(x^*)}{2g''(x^*)} = d \text{ as } t \rightarrow \infty$$

$$\epsilon^{(t+1)} \approx d^{(t)}\epsilon^{(t)}\epsilon^{(t-1)}$$

# Convergence of Secant Method

- Convergence:

  Under conditions akin to those for Newton's method, the secant method will converge to the root $x^*$

- Convergence order

$$\epsilon^{(t+1)} = x^{(t+1)} - x^*$$

$$x^{(t+1)} = x^{(t)} - g'(x^{(t)}) \frac{x^{(t)} - x^{(t-1)}}{g'(x^{(t)}) - g'(x^{(t-1)})}$$

$$\epsilon^{(t+1)} \approx d^{(t)} \epsilon^{(t)} \epsilon^{(t-1)}$$

$$d^{(t)} \to \frac{g'''(x^*)}{2g''(x^*)} = d \text{ as } t \to \infty$$

convergence with order $\beta$

$$\lim_{t\to\infty} \frac{|\epsilon^{(t+1)}|}{|\epsilon^{(t)}|^\beta} = c$$

$$\lim_{t\to\infty} |\epsilon^{(t)}|^{1-\beta+1/\beta} = \frac{c^{1+1/\beta}}{d}$$

$$1 - \beta + 1/\beta = 0$$

$$\beta = (1+\sqrt{5})/2 \approx 1.62$$

**Q1**: what's the convergence order of Secant Method?
**Q2**: does it converge faster or slower than the Newton's?

# $M_3$: Fixed-Point Iteration

- Fixed-point of a function $G$
  - a point whose evaluation by function $G$ equals itself, i.e., $G(x) = x$.
- Fixed-point strategy for finding roots of $g'$
  - determine a function $G$ for which $g'(x) = 0$ if and only if $G(x) = x$
  - transform the problem:

    finding a root of $g'$ $\Longrightarrow$ finding a fixed point of $G$
  - hunt for a fixed point:

    updating iteratively by $x^{(t+1)} = G(x^{(t)})$
- Specify function $G$
  - any suitable function $G$ may be tried
  - the most obvious choice is $G(x) = g'(x) + x$ $\Longrightarrow$

updating equation
$$x^{(t+1)} = g'\left(x^{(t)}\right) + x^{(t)}$$

# Convergence of Fixed-Point Method

- The convergence of this method depends on whether $G$ is *contractive*
- To be contractive on $[a, b]$, $G$ must satisfy

  **1.** $G(x) \in [a, b]$ whenever $x \in [a, b]$, and

  **2.** $|G(x_1) - G(x_2)| \leq \lambda |x_1 - x_2|$ for all $x_1, x_2 \in [a, b]$ for some $\lambda \in [0, 1)$.

  Lipschitz condition　　　　Lipschitz constant

**Contractive Mapping Theorem:** If $G$ is contractive on $[a, b]$, then there exists a unique fixed point $x^*$ in this interval, and the fixed-point algorithm will converge to it from any starting point in the interval. The error at step $t$ is

$$|x^{(t)} - x^*| \leq \frac{\lambda^t}{1-\lambda} |x^{(1)} - x^{(0)}|$$

**Convergence:** not universally assured, unless the Lipschitz condition holds, e.g.,

$$|G'(x)| \leq \lambda < 1 \text{ for all } x \text{ in } [a, b]$$

**The order of convergence:** depends on $\lambda$ if fixed-point iteration converges

# Choose the Form of $G$

- The effectiveness of fixed-point iteration is highly dependent on the chosen form of $G$

- For example, consider finding the root of $g'(x) = x + \log x$
  - $G(x) = (x + e^{-x})/2$ converges quickly
  - $G(x) = e^{-x}$ converges more slowly
  - $G(x) = -\log x$ fails to converge at all

❖ Exercise 1:

Please verify the above statements by comparing the order of convergence for different specifications of function $G$

# Adjust Updating Function $G$ by Scaling

- Fixed-point method

  $G(x) = g'(x) + x$ for interval $[a, b]$

  using the naïve updating function

- Convergence condition:

  $G'(x) = |g''(x) + 1| < 1$ on [a, b]

  may not hold for many cases

- Solution: we rescale $g'(x)$ by

  $G_\alpha(x) = \alpha g'(x) + x$

- Technical requirement
  - $g''$ is bounded
  - $g''$ does not change sign on $[a, b]$
  - $\alpha \neq 0$

- The rescaled function fits the framework

  $\alpha g'(x) = 0$ if and only if $g'(x) = 0$

- We can choose $\alpha$ to guarantee convergence

  $G_\alpha'(x) = |\alpha g''(x) + 1| < 1$

Notes

- Although one could carefully calculate a suitable $\alpha$, it may be easier just to try a few values
- If the method converges quickly, then the chosen $\alpha$ was suitable

# Scaled Fixed-Point Method: an Example

Target function:

$$g(x) = (\log x)/(1+x)$$

Updating equation:

$$G(x) = \alpha g'(x) + x$$

Scale parameter:

$$\alpha = 4$$

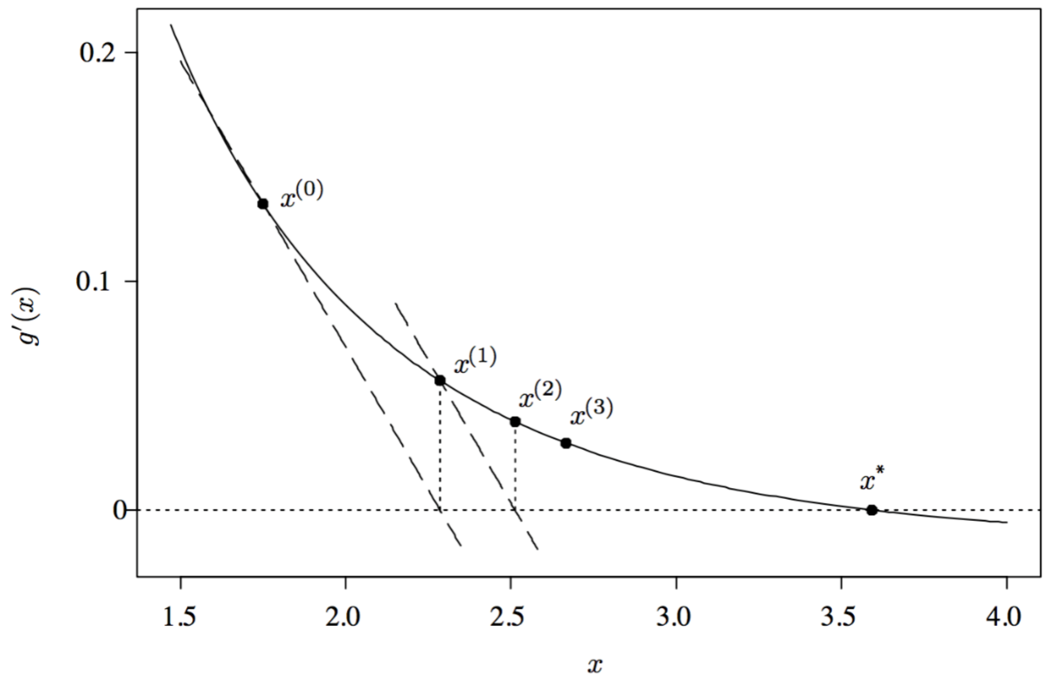
- Exercise 2:

  Deduce the concrete form of $|\alpha g''(x) + 1|$ to show why let $\alpha = 4$ is a good choice?

- Exercise 3:

  The two dash lines in the right figure looks parallel. Can you show whether this conjecture is correct?

Fig. 2.6 First three steps of scaled fixed-point iteration



# Summary of Methods

- $M_1$: Bisection/Bracketing Method
- $M_2$: Newton's Method
  - $M_2'$ : Fisher scoring (special case of $M_2$)
  - $M_2''$: Secant method (approximation of $M_2$)
- $M_3$: Fixed-Point/Functional Iteration
  - $M_3'$ : Fixed-point method with naïve updating function
  - $M_3''$: Fixed-point method with scaled updating function

**Note:** Newton's method $M_2$ and its variations are all special case of $M_3$

# Reference

- Reference: [T1] Chapter 2.1
- Further reading:
  – [R1] Chapter 6: Solution of Nonlinear Equations & Optimization

# Further Reading

- Reference: [T1] Chapter 2.1
- Further reading:
  - [R1] Chapter 6: Solution of Nonlinear Equations & Optimization