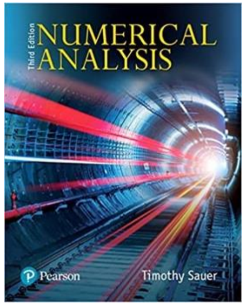
计算方法

刘景铖

计算机软件新技术国家重点实验室
南京大学

课程基本信息

- 教师：刘景铖
- Email：liu@nju.edu.cn
- Office hour：周二 16:00-18:00，计算机系516
- 课程主页：https://tcs.nju.edu.cn/wiki
- 数值分析（Numerical Analysis）（原书第2版）Timothy Sauer, 机械工业出版社



参考：

- Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics. Justin Solomon. CRC Press
- Lx=b,拉普拉斯方程求解以及它的应用（Lx=b, Laplacian Solver and Their Algorithmic Applications）Nisheeth K. Vishnoi

课程QQ群

- 作业每两周发布一次
- 第一次作业今天将会发布
- 作业Email: nm_nju_2026@163.com
- QQ群可以讨论课程相关的问题
- 作业可以小组讨论（<=3人）
  - 清晰标明合作者，及各自的贡献
  - 使用了的参考资料必须注明
  - 但写作必须由自己独立完成，不可照抄
  - 类似地，别人提供的想法也必须注明
- 分辨合作与作弊之间的区别
  - 参与讨论之前，请先花时间自己进行思考
  - 避免参与你不能提供贡献的讨论
  - 作弊不仅让你丧失一次学习的机会
  - 还会影响你以后的自信心
- 对学术不诚信的行为零容忍
- 迟交作业需要提前预约

计算方法 2026

群号: 522312428

对作弊的认定参照http://www.acm.org/publications/policies/plagiarism_policy

浮点数误差

数学公式上的准确，在有限精度的计算中未必准确

```
double x=0.3*3 + 0.1;
double y=1.0;
if (x==y) cout<<"Equal";
else cout<<"Not equal";
```

A common fix: 使用 abs(x-y)<eps 作为浮点数相等的测试

浮点数误差的累积

以计算向量**x[i]的欧氏**长度为例

```
for (int i=0; i < n; i++) sum+=x[i]*x[i];
return sqrt(sum);
```

更好的做法：

```
maxElement=max(x[i]);  y[i]=x[i]/maxElement;
for (int i=0; i < n; i++) sum+=y[i]*y[i];
return sqrt(sum) * maxElement;
```

Rounding vs truncation: 同样是求和，就近取整与截断误差哪个好？

更一般地，可以使用Kahan summation (compensated summation)

[Python 3.12]**开始sum使用的是其**变种之一

课后阅读：课本第0章

IEEE 754标准浮点数也让William Kahan得到1989图灵奖

Photo by George M. Bergman

欧洲空间局的Ariane 5火箭

Ariane flight V88 vehicle no. 501

- 1996年首次发射
- 升空37秒后失控并解体

原因：64位浮点数转换为16位整数时导致整数溢出

Design flaw!

在之前的Ariane 4中没有出错，仅仅是因为速度低

另一案例：防空导弹

整数溢出

CWE-190: Integer Overflow or Wraparound

近年来的一个著名例子： CVE-2023-32434 具体分析

该漏洞在卡巴斯基2023年发现

针对苹果iOS设备的大规模持续渗透行动中：攻击链上4个主要漏洞之一，位于内核级

拓展阅读：Operation Triangulation

计算方法

- 计算方法并不仅仅是一些“工程小技巧”
- 作为一门学科，它有着一系列的核心思想：
  - 收敛性：能合理近似的算法，至少需要收敛
  - 复杂性
  - 条件性 (鲁棒性)
  - 压缩性
  - **正交性**
  - ……
- 误差与偏差，也不仅仅来源于浮点数或离散化处理

数学建模与计算建模

- **回顾：物理中的理想气体**、光滑表面等等假设
- All models are wrong, but some are useful

2.3 Parsimony

Since all models are wrong the scientist cannot obtain a "correct" one by excessive elaboration. On the contrary following William of Occam he should seek an economical description of natural phenomena. Just as the ability to devise simple but evocative models is the signature of the great scientist so overelaboration and overparameterization is often the mark of mediocrity.

2.4 Worrying Selectively

Since all models are wrong the scientist must be alert to what is importantly wrong. It is inappropriate to be concerned about mice when there are tigers abroad.

Box, George E. P. (1976), "Science and statistics" (PDF), Journal of the American Statistical Association

如何定量地研究这些偏差？

一个定量分析的简单例子

- 给定函数f
- 输入： $x+\Delta x$
- 计算：$f(x)$

$$f(x+\Delta x)-f(x)\approx \Delta x\, f'(x)$$

（**中值定理**）

$$\text{设}f\in C^1[x,x+\Delta x],\text{则}\exists\xi\in(x,x+\Delta x),$$
$$f(x+\Delta x)-f(x)=\Delta x\cdot f'(\xi)$$

一个定量分析的简单例子

- 给定函数f
- 输入： $x + \Delta x$ 绝对误差
- 计算：$f(x)$

$$f(x+\Delta x) - f(x) \approx \Delta x\, f'(x)$$

（**中值定理**）

$$\text{设} f \in C^1[x, x+\Delta x], \text{则} \exists \xi \in (x, x+\Delta x),$$
$$f(x+\Delta x) - f(x) = \Delta x \cdot f'(\xi)$$

一个定量分析的简单例子

- 给定函数f
- 输入： $x+\Delta x$
- 计算：$f(x)$

$$f(x+\Delta x)-f(x)\approx \Delta x\, f'(x)$$

绝对误差的放大比例 $\approx f'(x)$

- $|f'(x)|>1?$
- $|f'(x)|<1?$

- 数值稳定性，既可能来自于f的选取，也取决于问题本身的性质（混沌）

一个定量分析的简单例子

- 给定函数f
- 输入： $x + \Delta x$

相对误差：$\frac{\Delta x}{x}$

- 计算：$f(x)$

$$f(x + \Delta x) - f(x) \approx \Delta x\, f'(x)$$

相对误差的放大比例 $\approx$ **条件数:=** $\frac{x f'(x)}{f(x)}$

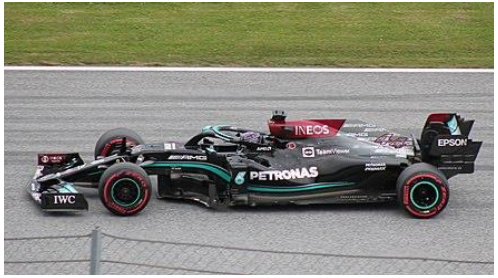
- 如果f不可导呢？或者$f'(x)$难以直接控制？

现代科学与工业界中的计算任务

一、插值与拟合

- 预测天体运动轨迹

- 工业设计

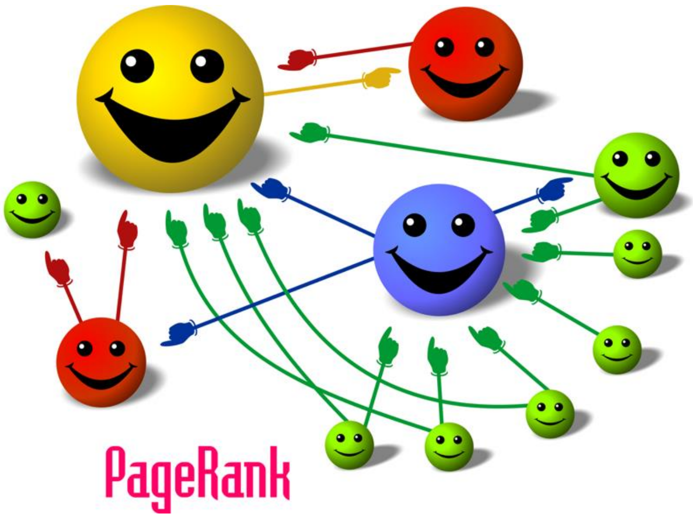


- 计算机动画

现代科学与工业界中的计算任务

二、特征值与随机游走

- （线性）动力系统的稳定性
  - 电力、网络调度策略的收敛性
- 线性代数问题的稳定性
- 现代算法设计
  - PageRank
  - 估计互联网的节点数
  - 局部聚类
  - Markov chain Monte Carlo

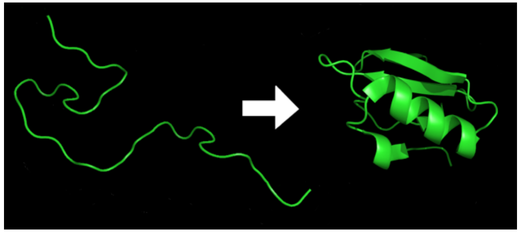


Source: Wikipedia CC BY-SA 2.5

现代科学与工业界中的计算任务

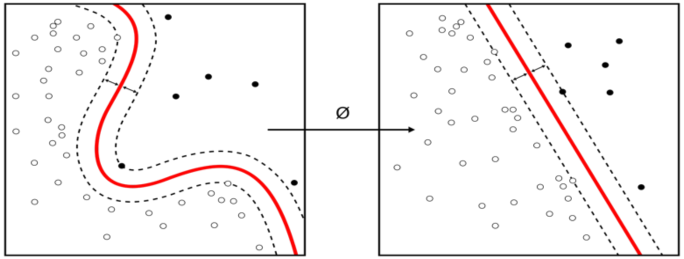
三、最优化与线性规划

- 蛋白质结构预测与折叠模型



Source: Wikipedia public domain

- 机器学习



Source: Wikipedia CC BY-SA 4.0

Beyond computational simulation

“计算的视角” Computational lens

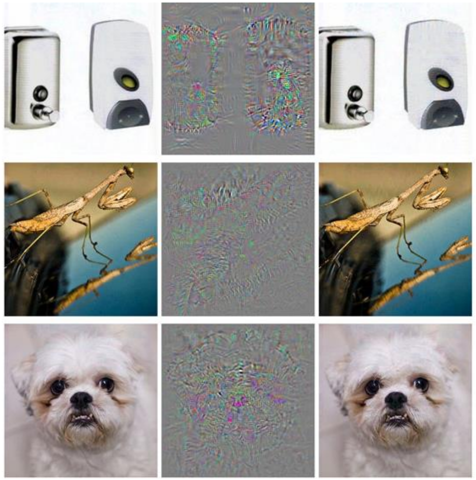
- Computational biology
- Algorithmic game theory
- Computational phase transition

Comparison with discrete math:

- A continuous perspective is needed even for the classic “combinatorial” max-flow problem
- Many problems about Boolean functions, can be reduced to a continuous analog thanks to the **Invariance Principle**.

更一般的计算稳定性挑战

- 游戏中的物理引擎
- 图像识别



右边的图片均被AlexNet识别为鸵鸟

求解方程的根

- 输入：连续函数f
- 求解：任意一个根r使得$f(r) = 0$
- 这里假设f是以oracle的形式给定(神喻模型)

- **思考**：为什么需要假设函数是连续的？

求解方程的根 - 二分法

- （介值定理）连续函数f满足$f(a)f(b) < 0$，则f在(a, b)中间有一个根r使得$f(r) = 0$

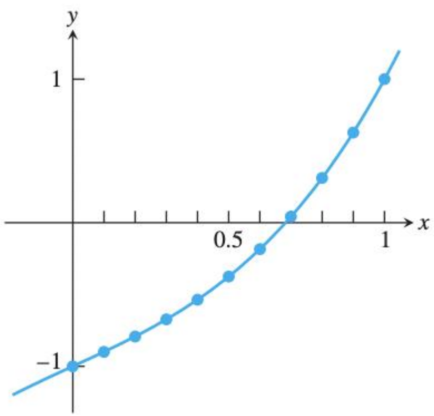


Figure 1.1 **A plot of** $f(x) = x^3 + x - 1$. The function has a root between 0.6 and 0.7.

求解方程的根 - 二分法

Bisection Method

Given initial interval $[a, b]$ such that $f(a)f(b) < 0$
while $(b - a)/2 > \text{TOL}$
    $c = (a + b)/2$
    if $f(c) = 0$, **stop**, **end**
    if $f(a)f(c) < 0$
        $b = c$
    else
        $a = c$
    end
end
The final interval $[a, b]$ contains a root.
The approximate root is $(a + b)/2$.

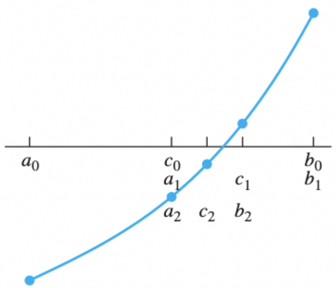


Figure 1.2 The Bisection Method. On the first step, the sign of $f(c_0)$ is checked. Since $f(c_0)f(b_0) < 0$, set $a_1 = c_0, b_1 = b_0$, and the interval is replaced by the right half $[a_1, b_1]$. On the second step, the subinterval is replaced by its left half $[a_2, b_2]$.

求解方程的根 - 二分法

- 收敛速度？

- 在n次二分之后，区间$[a_n, b_n]$长度为$\frac{b-a}{2^n}$
- 输出中点$x_c = (a_n + b_n)/2$，距离区间上任意一点的距离为$\frac{b-a}{2^{n+1}}$

- 记r为一个根，则$|x_c - r| < \frac{b-a}{2^{n+1}}$

求解方程的根 – 不动点迭代法

- **(不动点)** 若$g(r) = r$，则称r为g的一个不动点

- 不动点迭代法：
  - 猜测一个x_0
  - $x_{i+1} = g(x_i), i = 0,1,2, \dots$

- **不一定收**敛：e.g. $g(x) = x + 1$
- **但是**，对于连续函数g，**如果** x_i **收**敛，则一定收敛到不动点（证明见下一页）

求解方程的根 – 不动点迭代法

不动点迭代法:

- 猜测一个x_0
- $x_{i+1} = g(x_i), i = 0,1,2,\ldots$

对于连续函数g，**如果** x_i **收敛**，则一定收敛到一个不动点r

- $g(\lim_{i\to\infty} x_i) = \lim_{i\to\infty} g(x_i)$ by continuity
- $\lim_{i\to\infty} g(x_i) = \lim_{i\to\infty} x_{i+1}$ by definition of x_{i+1}
- Then, $g(\lim_{i\to\infty} x_i) = \lim_{i\to\infty} x_{i+1} = \lim_{i\to\infty} x_i$
- Thus, $r \coloneqq \lim_{i\to\infty} x_i$ is a fixed point of g

求解方程的根 – 不动点迭代法

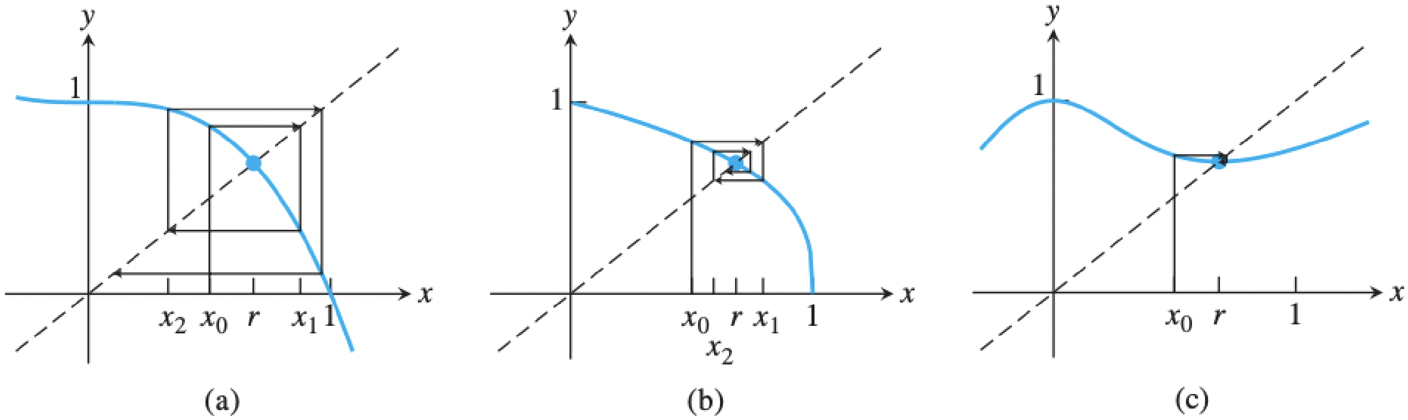


Figure 1.3 Geometric view of FPI. The fixed point is the intersection of $g(x)$ and the diagonal line. Three examples of $g(x)$ are shown together with the first few steps of FPI. (a) $g(x) = 1 - x^3$ (b) $g(x) = (1-x)^{1/3}$ (c) $g(x) = (1+2x^3)/(1+3x^2)$.

从x_0开始迭代

$$x_{i+1} = g(x_i), i = 0,1,2,\ldots$$
$$x_1 = g(x_0)$$
$$x_2 = g(x_1)$$

求解方程的根 – 不动点迭代法

- **不动点迭代法**：
  - 猜测一个x_0
  - $x_{i+1} = g(x_i), i = 0,1,2,\ldots$

- **不动点迭代满足线性收敛**：
  - 记e_i为第i**次迭代的**误差
  - **考**虑$S = \lim_{i\to\infty} \frac{e_{i+1}}{e_i}$
  - **如果**S<1，那么说**不**动点迭代**<u>线性收敛</u>**，S为收敛速度

求解方程的根 – 不动点迭代法

- 不动点迭代法：
  - 猜测一个x_0
  - $x_{i+1} = g(x_i), i = 0,1,2,\ldots$

- 定理：若g是连续可导，r满足$g(r) = r$，$S = |g'(r)| < 1$.
  则对于任意足够接近r的猜测，g的不动点迭代线性收敛到不动点r，收敛速度为S。
- 证明：
- $x_{i+1} - r = g(x_i) - g(r) = g'(\xi_i)(x_i - r)$
- 记$e_i = |x_i - r|$，则有$e_{i+1} = |g'(\xi_i)|e_i$

求解方程的根 – 不动点迭代法

- 定理：若g是连续可导，r满足$g(r)=r$，$S=|g'(r)|<1$.
  则对于任意足够接近r的猜测，g的不动点迭代线性收敛到不动点r，收敛速度为S。
- 证明：
- $x_{i+1}-r=g(x_i)-g(r)=g'(\xi_i)(x_i-r)$
- 记$e_i=|x_i-r|$，则有$e_{i+1}=|g'(\xi_i)|e_i$
- **因为** $S=|g'(r)|<1$,则在足够接近r**的一个**邻域里面，
  $$|g'(\xi_i)|<(S+1)/2$$
- **所以**误差至少会以$(S+1)/2$**的速度**缩减
- $\lim\limits_{i\to\infty}\frac{e_{i+1}}{e_i}=\lim\limits_{i\to\infty}|g'(\xi_i)|=|g'(r)|=S$

不动点迭代

- 稳定的不动点
  - 局部地收敛(local convergence)
  - 误差会越来越小
- 不稳定的不动点
  - 局部地发散
  - 误差会越来越大
  - 与“混沌”现象相关

根的敏感性

假设计算函数f本身有误差，记误差函数为$\epsilon g(x)$

- 对根的影响会有多大？
- 隐函数定理求导？
- $f(r+\Delta r)+\epsilon g(r+\Delta r)=0$

$$f(r)+\Delta r f'(r)+\epsilon g(r)+\epsilon(\Delta r)g'(r)+O\left((\Delta \mathrm{r})^2\right)=0$$

根的敏感性

- 假设计算函数f本身有误差，记误差函数为$\epsilon g(x)$, 对根的影响会有多大?
- $f(r+\Delta r)+\epsilon g(r+\Delta r)=0$

$$f(r)+\Delta r f'(r)+\epsilon g(r)+\epsilon(\Delta r)g'(r)+O\big((\Delta \mathrm{r})^2\big)=0$$

整理后$(\Delta r)\big(f'(r)+\epsilon g'(r)\big)\approx -f(r)-\epsilon g(r)=-\epsilon g(r)$

即$\Delta \mathrm{r}\approx \frac{-\epsilon g(r)}{f'(r)+\epsilon g'(r)}\approx -\epsilon\frac{g(r)}{f\prime(r)}$

用隐函数定理求导，可严格地得到一样的结果（见作业1）

牛顿法

给定f**可**导且导函数f'**也已知**，求x使得$f(x)=0$

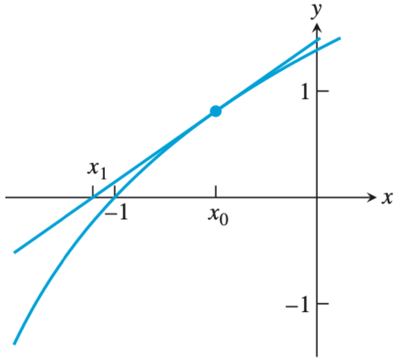


Figure 1.8 **One step of Newton's Method.** Starting with x_0, the tangent line to the curve $y=f(x)$ is drawn. The intersection point with the x-axis is x_1, the next approximation to the root.

牛顿方法：

x_0:初始估计

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)},k=0,1,2\ldots$$

牛顿法的二次收敛 (Quadratic convergence)

记$e_i = |x_i - r|$为i**次迭代的**误差

- **局部收**敛：在某个足够小的邻域里面，任意的初始值都会收敛
- **二次收**敛：如果$M = \lim_{i\to\infty} \frac{e_{i+1}}{e_i^2} < \infty$，则称该迭代方法二次收敛。

定理. 如果f二次可导且$f'(r) \neq 0$，其中r满足$f(r) = 0$，那么牛顿法在局部二次收敛到r，且

$$\lim_{i\to\infty} \frac{e_{i+1}}{e_i^2} = \left|\frac{f''(r)}{2f'(r)}\right|$$

牛顿法的二次收敛分析：不动点方程

定理. 如果f二次连续可导且$f'(r) \neq 0$，其中r满足$f(r)=0$，那么牛顿法在局部二次收敛到r，且

$$\lim_{i\to\infty} \frac{e_{i+1}}{e_i^2} = \left|\frac{f''(r)}{2f'(r)}\right|$$

证明：首先注意到牛顿法等价于以下不动点方程$x_{k+1} = g(x_k)$，其中$g(x) = x - \frac{f(x)}{f'(x)}$. g的不动点对应于f的根。

$$g'(x) = 1 - \frac{f'(x)^2 - f(x)f''(x)}{f'(x)^2} = \frac{f(x)f''(x)}{f'(x)^2}.$$

在g的不动点处，有$g'(r) = 0$. 因此牛顿法是局部收敛的。

牛顿法的二次收敛分析

定理. 如果f二次可导且$f'(r)\neq 0$，其中r满足$f(r)=0$，那么牛顿法在局部二次收敛到r，且

$$\lim_{i\to\infty}\frac{e_{i+1}}{e_i^2}=\left|\frac{f''(r)}{2f'(r)}\right|$$

证明：考虑第i**次迭代**，在r处进行Taylor**展开有**：

$$f(r)=f(x_i)+(r-x_i)f'(x_i)+\frac{(r-x_i)^2}{2}f''(\xi_i),\qquad \text{for some } \xi_i$$

因为r满足$f(r)=0$，化简得到

$$x_i-\frac{f(x_i)}{f'(x_i)}-r=\frac{(r-x_i)^2}{2}\frac{f''(\xi_i)}{f'(x_i)}$$

回顾牛顿法的迭代方程：

$$e_{i+1}=|x_{i+1}-r|=e_i^2\left|\frac{f''(\xi_i)}{2f'(x_i)}\right|$$

由局部收敛可知，只要足够的接近r，则有$x_i\to r$且$\xi_i\to r$。由f''和f'的连续性可知

$$\lim_{i\to\infty}\frac{e_{i+1}}{e_i^2}=\lim_{i\to\infty}\left|\frac{f''(\xi_i)}{2f'(x_i)}\right|=\left|\frac{f''(r)}{2f'(r)}\right|$$

课外阅读：牛顿法的变种

- 牛顿法只用到了一阶导数的信息
  - 注：二次收敛的分析也可以在只假设f'是C-Lipschitz的情况下进行
  - 如果使用更高阶的导数呢？例如，并不只是考虑切线，而是在局部做一个二次曲线的近似
- 牛顿法更容易推广到高维，可以解决多变量的求根问题/最优化问题
- 然而导数可能并不容易求解：割线法 (secant method)，quasi-newton method
  - Secant method: $x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}, k = 1,2,\ldots$
  - Convergence rate of Secant method: $\frac{1+\sqrt{5}}{2} \approx 1.618$
  - quasi-newton method非常常见的一个实现是Broyden-Fletcher-Goldfarb-Shanno (BFGS)算法
- 现实中，往往会结合二分法和牛顿/割线法（如Dekker/Brent方法）
  - 二分法可以保证全局收敛
  - 牛顿/割线法可以保证在足够接近解的时候，更快速地收敛到需要的精度

回顾与展望下节课

不动点有非常深刻的理论，例如通过Brouwer**不**动点定理，可以证明

- Sperner’s lemma: **公平地分担房租** (Rent split calculator)
- 纳什市场均衡的存在性
- …………

在后面的课程中，有不少的迭代算法的分析，本质上都可以看成是在研究某种不动点迭代

下节课：

- **插**值
- **插**值稳定性的应用：秘密分享、自纠错码