第三章　迭代法

引言：

针对非线性方程的求解问题

- 代数方程、超越方程
- 方程 $f(x)=0$ 的解或根
- 单根、复根

本章：求非线性方程及线性方程组数值解的迭代算法，二分法、迭代法、牛顿迭代法，雅克比迭代法、高斯-赛德尔迭代法和SOR迭代法。

第三章　迭代法

§ 3.1 二分法

- 根的估计
- 二分法

非线性方程的根

求 $f(x)=0$ 的根

- 代数： *5x+2=7*
  超越： **超越函数，如** $\sin x$，e^x，$\ln x$ **等**，$\sin x+x=0$
- 实根与复根
- 根的重数

如果对于数 α 有 $f(\alpha)=0, f'(\alpha)\neq 0$，则称 α 为 $f(x)$ 的单根。

如果有 $k>1, f(x^*)=f'(x^*)=\ldots=f^{(k-1)}(x^*)=0, f^{(k)}(x^*)\neq 0$，则 x^* 为 k 重根.

- 有根区间：$[a,b]$ 上存在 $f(x)=0$ 的一个根

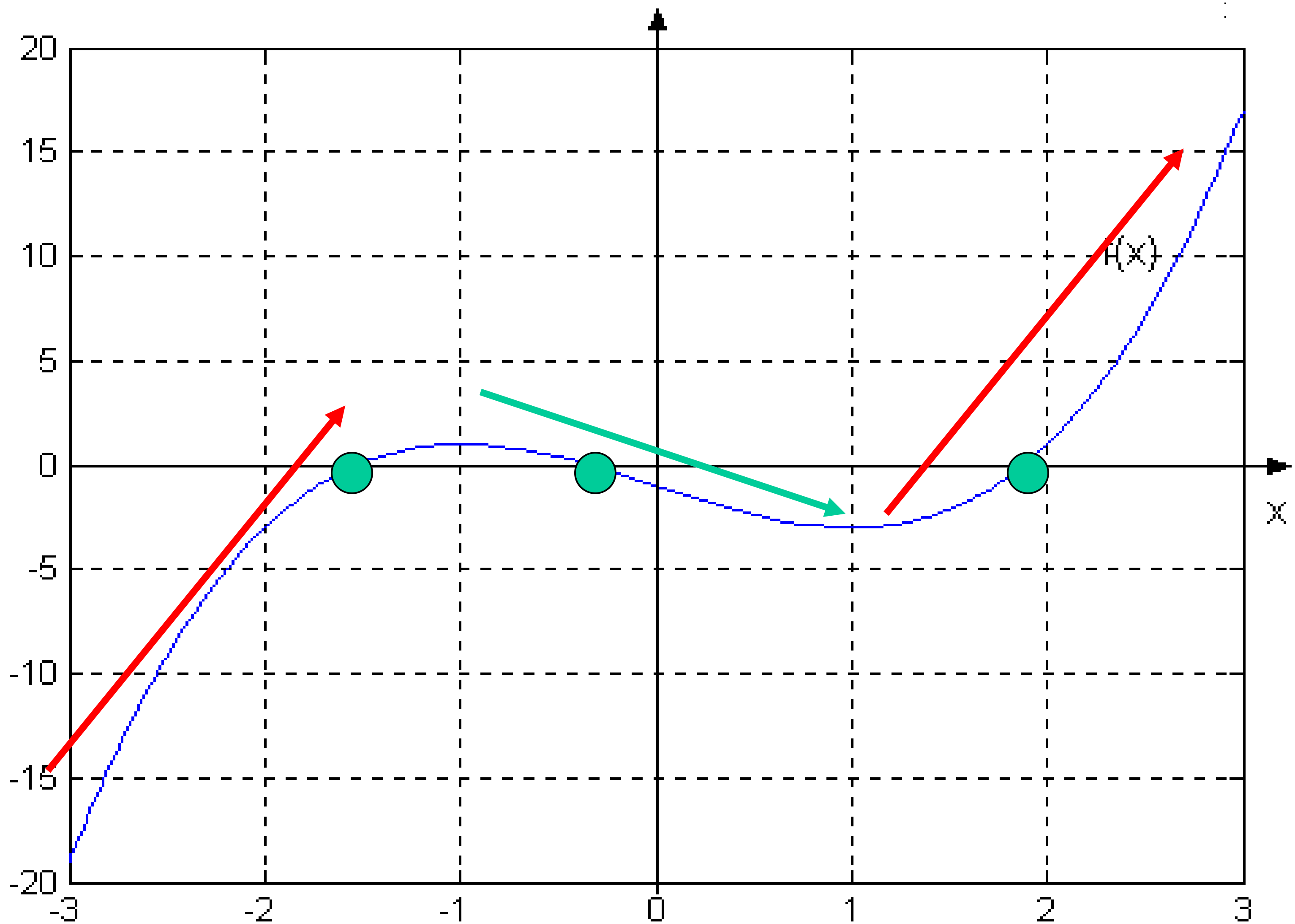
研究内容： 在有根的前提下求出方程的近似根。

1. 根的估计

- **引理3.1（连续函数的介值定理）** 设$f(x)$在有限闭区间$[a,b]$上连续，且$f(a)\,f(b)<0$，则存在$x^*\in(a,b)$使$f(x^*)=0$。
- 例3.1 证明$x^3-3x-1=0$有且仅有3个实根，并确定根的大致位置使误差不超过$\varepsilon=0.5$。

解：

- **单调性分析解的位置，利用一阶导数**
- **选步长$h=2\varepsilon$，扫描节点函数值，异号区间内有根**
- **取区间中点作为估计值，误差不超过ε**



20
15
10
5
0
-5
-10
-15
-20
-3
-2
-1
0
1
2
3
f(x)
x

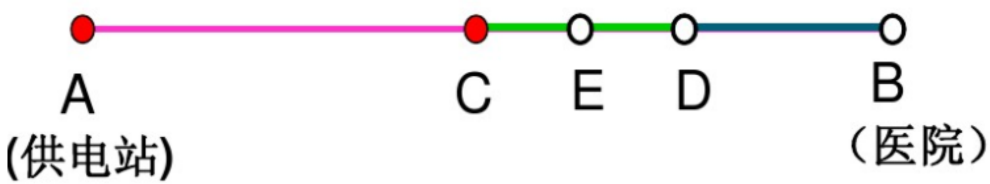
抢修电路、水管、气管

在某个雷雨交加的夜晚，医院的医生正在抢救病人，忽然电停了。据了解，原因是供电站到医院的某处线路出现了故障，维修工如何迅速查找出故障所在？（线路长10km，每50m一根电线杆)

提醒：若沿着线路一段一段查找，困难多，费时长，10km大约有200根电线杆子要查。

抢修电路、水管、气管

探索问题，提取原理，找到方法。



取中点，这样可以有效缩减时间。

思考：每查一次，就可以把待查的线路长度缩减一半，要把故障可能发生的范围缩小到50m~100m左右，即一两根电杆附近，大概要查多少次?

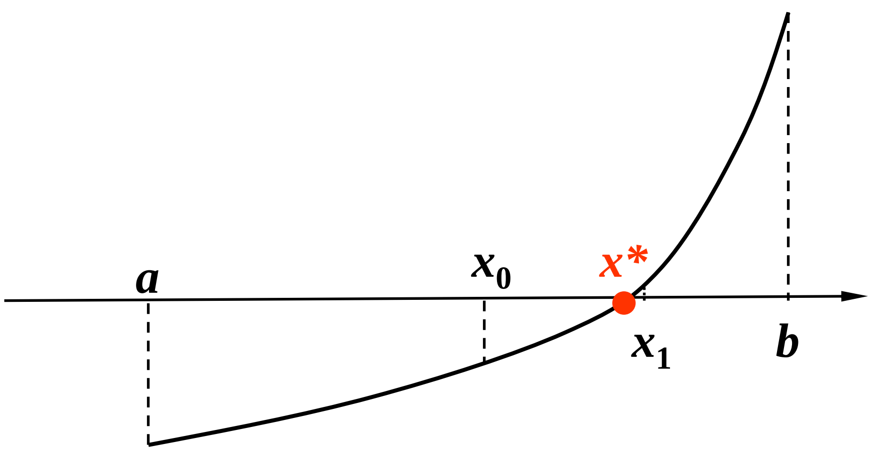
2. 二分法

□ 基本前提:

设 f 在$[a, b]$上连续，f 在 $[a, b]$存在唯一解，且 $f(a)\ f(b) < 0$。

□ 基本方法:

通过二等分不断缩小有根区间的长度，直到满足精度为止。



何时终止?

$$|x_{k+1} - x_k| < \varepsilon \ \text{或} \ |f(x_k)| < \varepsilon$$

不能保证 x 的精度

二分法的基本思想

设$f(x)$在[a,b]上连续，$f(x)=0$在[a,b]上存在唯一解，且$f(a)f(b)<0$。记$a_0=a$, $b_0=b$, $x_0=(a+b)/2$

- 第一步，计算$f(a_0)f(x_0)<0$？

  Y: $a_1=a_0$, $b_1=x_0$; N: $a_1=x_0$, $b_1=b_0$。

  两种情形均有$x^*\in[a_1, b_1]$, 记$x_1=(a_1+b_1)/2$

- 第k步，计算$f(a_{k-1})f(x_{k-1})<0$？

  Y: $a_k=a_{k-1}$, $b_k=x_{k-1}$; N: $a_k=x_{k-1}$, $b_k=b_{k-1}$。

  两种情形均有$x^*\in[a_k, b_k]$,记$\boldsymbol{x_k=(a_k+b_k)/2}$

- 过程直至$\boldsymbol{b_k-a_k\leq 2^*\varepsilon}$

误差分析

记 $a_0 = a, b_0 = b$, 第 k 步的有根区间为 $[a_k, b_k]$

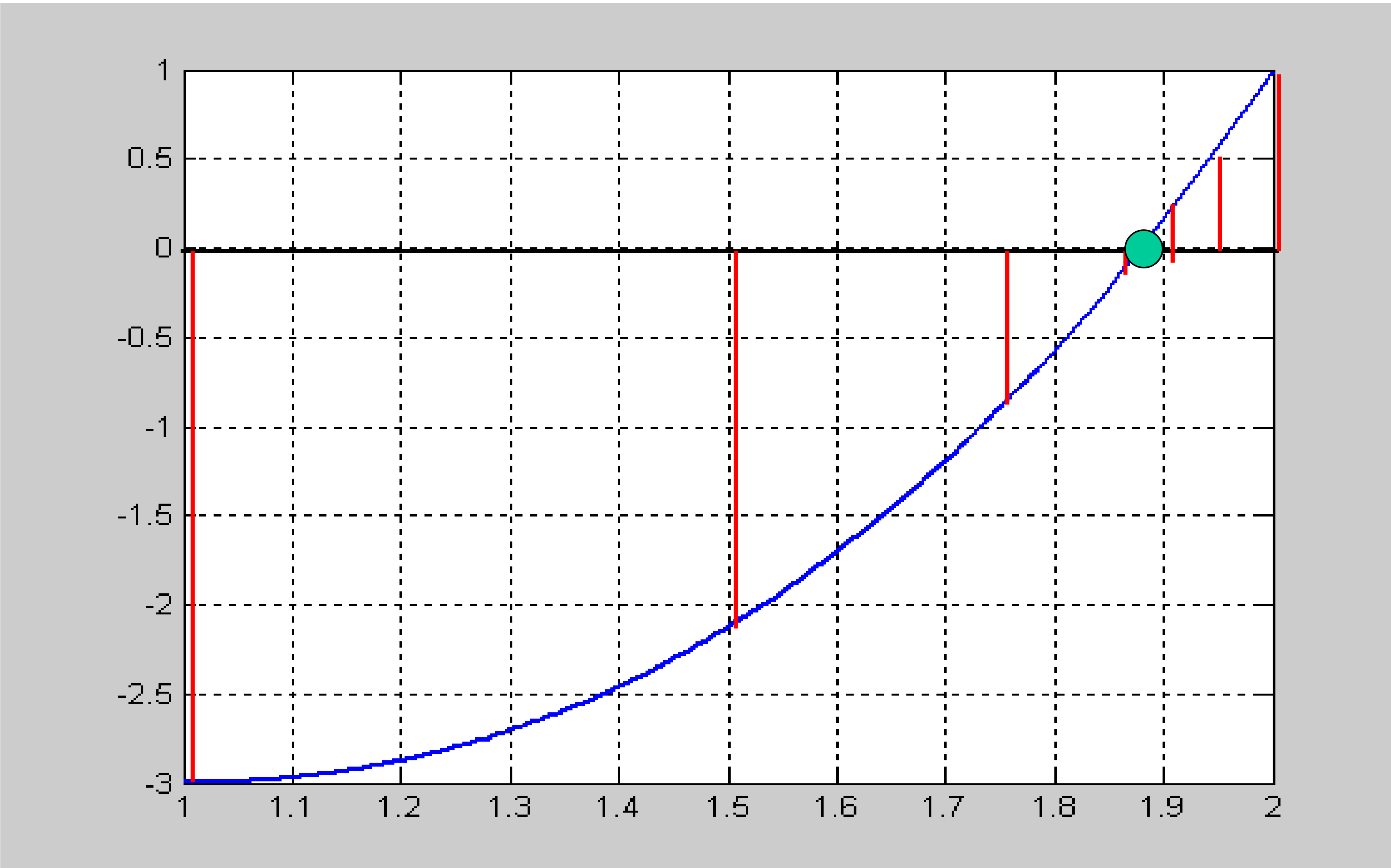
$$\left|x_k - x^*\right| = \left|\frac{b_k + a_k}{2} - x^*\right| \le \frac{b_k - a_k}{2} = \frac{b_{k-1} - a_{k-1}}{2^2} = \cdots = \frac{b_0 - a_0}{2^{k+1}} = \frac{b-a}{2^{k+1}}$$

对于给定的精度 ε,可估计二分法所需的步数 k：

$$\frac{b-a}{2^{k+1}} \le \varepsilon \Rightarrow \quad k \ge \log_2 \frac{b-a}{\varepsilon} - 1, \quad \text{取 } k = \left\lceil \log_2 \frac{b-a}{\varepsilon} \right\rceil - 1$$

例3.2 求$x^3–3x–1 = 0$在 [1,2]内的根

两位有效数字 $\varepsilon=0.5*10^{-1}$， $k\geq$ (ln 20/ ln 2)-1, 取$k=4$

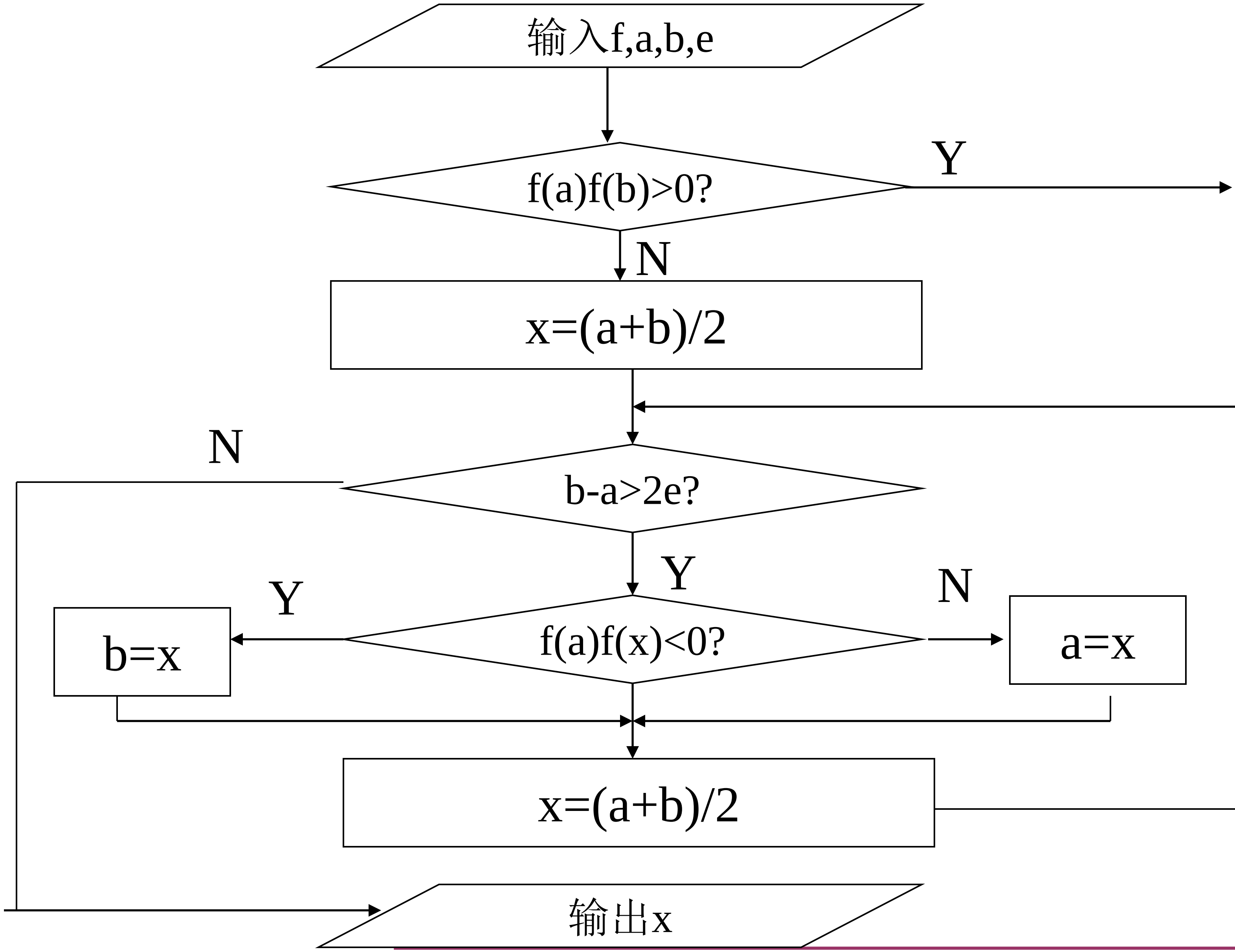


k	a_k	b_k	x_k	b_k-a_k	$f(a_k)\ f(x_k)$
0	1	2	1.5	1	+
1	1.5	2	1.75	0.5	+
2	1.75	2	1.875	0.25	+
3	1.875	2	1.9375	0.125	—
4	1.875	1.9375	1.90625	0.0675	

验证：

- ✓ 与3.1方法相比：提高了计算效率 6<11次
- ✓ 算法简单，收敛有保障
- ✓ 对两端点的选取苛刻，高精度问题收敛速度慢

3. 二分法的算法和程序



算法

算法 3.1 (二分法)

给定有根区间 $[a, b]$ **(** $f(a)\cdot f(b) < 0$**) 和 精度要求** ε

1. 令 $x = (a+b)/2$

2. 如果 $b - a <= 2\varepsilon$**, 停止计算，输出** x **,否则执行第3步**

3. 如果 $f(a)\,f(x) < 0$ **, 则令** $b = x$**，否则令** $a = x$**, 返回第1步**

P48. Matlab源程序：nabisect.m

- ✓ **简单易用**
- ✓ **对** $f(x)$ **要求不高，只要连续即可**

- ✓ **收敛速度慢**
- ✓ **无法求复根及偶重根**

根据数学中“不动点原理”设计出的迭代法，可以大大提高收敛速度，是求解各类非线性方程的主要算法。

§ 3.2 迭代法原理

- 迭代法的思想
- 不动点原理
- 局部收敛性
- 收敛性的阶

第二节 迭代法原理

基本思想

$f(x)=0 \xleftrightarrow{\text{等价变换}} x=g(x)$ 称为迭代函数

$f(x)$ 的零点 x^* $\Longleftrightarrow$ $g(x)$ 的不动点 x^*

$\boxed{x_k=g(x_{k-1})}$ 　 $\boxed{\text{不动点迭代}}$

具体做法：

从一个给定的初值 x_0 出发，计算 $x_1=g(x_0)$，$x_2=g(x_1)$, …得 $\{x_k\}_{k=0}^{\infty}$ 序列。若 $x_k \to x^*$ 且g(x) 连续，则 $x^*=g(x^*)$，即 x^* 是 g 的不动点，也就是 $f(x)$ 的零点。

成功条件：

1.保证变形的合理性：x_0附近同解；

2. 保证序列 $x_k \to x^*$，同时保证效率（速度）

例 $x^3-x-1=0$, $[1,2]$, 取 $x_0=1.5$

- 迭代公式1: $x_k = x_{k-1}^3 - 1$
- 迭代公式2: $x_k = (x_{k-1}+1)^{\frac{1}{3}}$

- 计算结果:

公式1:

k	x_k
0	1.5
1	2.375
2	12.4
3	1904

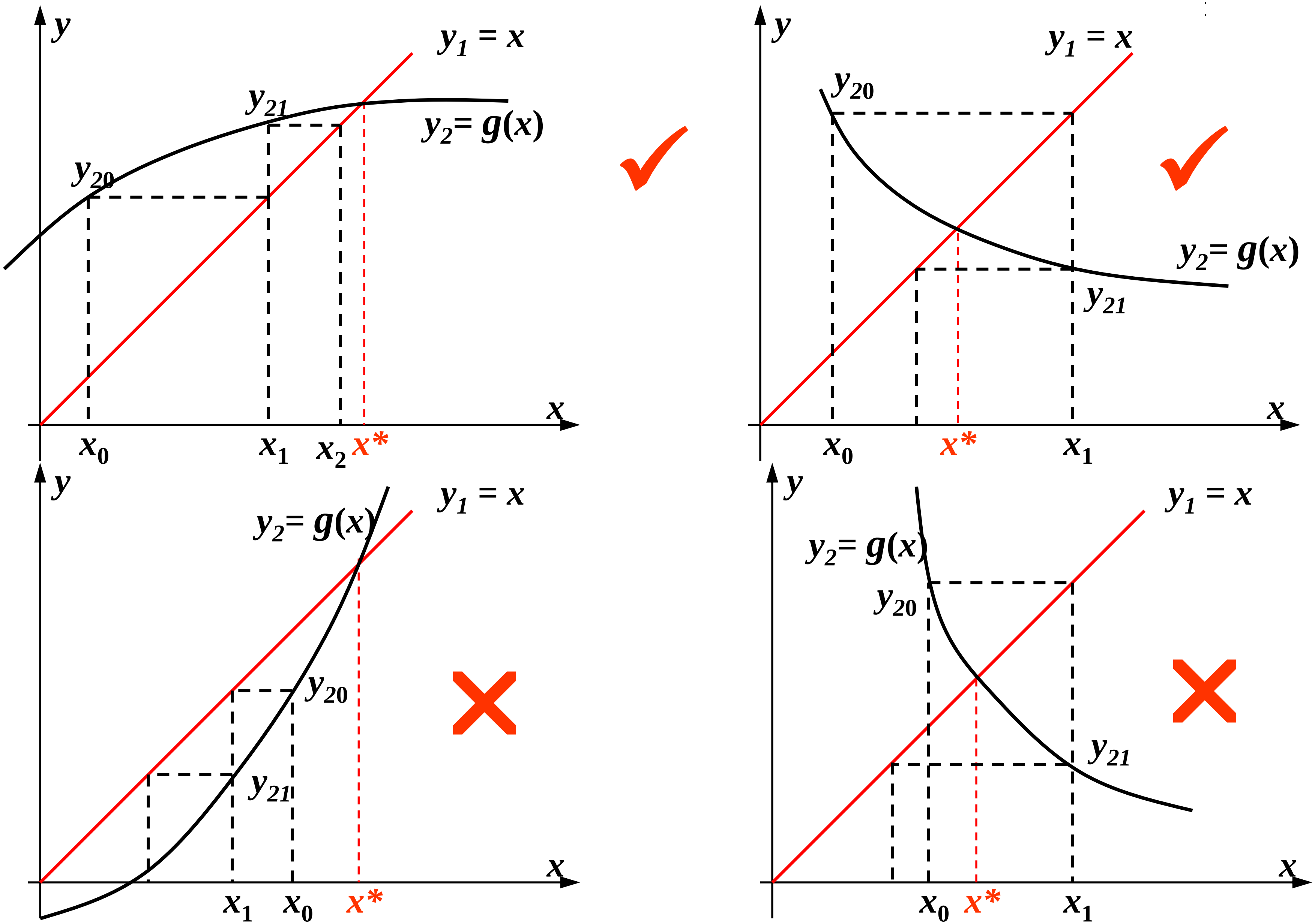
公式2:

k	x_k
0	1.5
1	1.35721
2	1.33086
3	1.32588
4	1.32494
5	1.32476
6	1.32473
7	1.32472

精确解 $x^* =$ 1.3247179...

怎么判断迭代公式收敛或发散呢?

□ 几何含义：求曲线 $y = g(x)$ 与直线 $y = x$ 的交点



收敛性分析—不动点原理

定理 3.1 设 映射$g(x)$在有限区间$[a, b]$上有连续的一阶导数，且满足

(1)封闭性：即对任意 $x\in[a, b]$， $g(x)\in[a, b]$；

(2) 压缩性：存在 $L\in(0,1)$, 对 $\forall x\in[a, b]$，$|g'(x)|\le L$成立,

则：

(a) $g(x)$在[a, b]上存在唯一的不动点x^*，且对任意 $x_0\in[a, b]$, $x_k = g(x_{k-1})$收敛于x^* 。

(b) 有如下的误差估计:

后验估计：

$$|x^*-x_k|\le\frac{L}{1-L}|x_k-x_{k-1}|$$

可用$|x_k-x_{k-1}|$来控制收敛精度

先验估计：

$$|x^*-x_k|\le\frac{L^k}{1-L}|x_1-x_0|$$

L 越小收敛越快

例3.3 用定理3.1讨论 $x^3 - x - 1 = 0$ 在[1, 2]内根的迭代格式的合理性和收敛性

（1）$x_k = x_{k-1}^3 - 1$　　（2）$x_k = (x_{k-1} + 1)^{\frac{1}{3}}$

解：首先：判断格式合理性

然后：验证3.1的两个条件是否满足，若不满足则发散；若满足，<u>则收敛</u>。

进一步，利用估计式得到误差估计

- **利用先验估计式，可计算需要迭代的次数；**
- **后验估计式，可用于在计算过程中判断数值解的精度**

例 $x^3-x-1=0$, [1,2], 取 $x_0=1.5$

- 迭代公式1：$x_k = x_{k-1}^3 - 1$
- 迭代公式2：$x_k = (x_{k-1}+1)^{\frac{1}{3}}$

- 计算结果：

公式1：

k	x_k
0	1.5
1	2.375
2	12.4
3	1904

公式2：

精确解 $x^* =$ 1.3247179...

k	x_k
0	1.5
1	1.35721
2	1.33086
3	1.32588
4	1.32494
5	1.32476
6	1.32473
7	1.32472

✓ 比二分行法快

✓ 但需要估计L的大小，且两个条件相互牵制，使用不便

局部收敛性

- 迭代过程不收敛：1. 迭代格式不当；2. 初值选取不当

（定理3.1中的条件 $|g'(x)| \le L < 1$ 可以适当放宽）

定理3.2
设g(x)在x=g(x)的根x^*附近有连续的一阶导数，且$|g'(x^*)|<1$，则存在充分靠近 x^*的初值x_0，使$x_k=g(x_{k-1})$收敛于x^*。

- 这种在 x^* 的邻域内具有的收敛性称为**局部收敛性**。

例题 3.4 用局部收敛定理3.2 讨论例3.3迭代格式

讨论 $x^3 - x - 1 = 0$ 在[1, 2]内根的迭代格式的收敛性

（1）$x_k = x_{k-1}^3 - 1$

（2）$x_k = (x_{k-1} + 1)^{\frac{1}{3}}$

局部收敛性

解（1） $x^* \in (1,2), |g'(x)| > 1$ 无法保证 $x_k = x_{k-1}^3 - 1$ 收敛性

（2） $x^* \in (1,2)$

$$|g'(x^*)| = \frac{1}{3} \times \frac{1}{(x^*+1)^{\frac{2}{3}}} < \frac{1}{3 \times 2^{\frac{2}{3}}} < 1$$

根据局部收敛性定理，当选取x_0充分靠近x^*，使 $x_k = (x_{k-1}+1)^{\frac{1}{3}}$ 收敛于x^*

1. 具有局部收敛性的迭代计算上不一定收敛，还要看初值是否取的恰当；

2. 而不具有局部收敛性的迭代对任何初值都不可能收敛；

3. 具有全局收敛性的迭代一定具有局部收敛性。

例　用不同方法求 $x^2-3=0$的根 $\sqrt{3}$, 取 $x_0=2$. 讨论合理性和收敛性

- 迭代公式1：$x_{k+1}=x_k^2+x_k-3$　　[1, 2]上迭代收敛性?
- 迭代公式2：$x_{k+1}=3/x_k$
- 迭代公式3：$x_{k+1}=x_k-(x_k^2-3)/4$
- 迭代公式4：$x_{k+1}=(x_k+3/x_k)/2$
- 计算结果：

k	x_k	方法1	方法2	方法3	方法4
0	x_0	2	2	2	2
1	x_1	3	1.5	1.75	1.75
2	x_2	9	2	1.734375	1.732143
3	x_3	87	1.5	1.732361	1.732051
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

精确值：

$\sqrt{3}=1.7320508\ldots$

怎么判断收敛的迭代公式的速度快慢呢？

收敛性的阶

定义3.1

设迭代 $x_{k+1} = g(x_k)$ 收敛到 $g(x)$ 的不动点 x^*。

记绝对误差 $e_k = x^* - x_k$，若存在p，使 $\lim\limits_{k\to\infty}\left|\dfrac{e_{k+1}}{e_k{}^p}\right| = C \neq 0$ (C为常数)

则称该迭代为 p 阶收敛。

(1) 当 $p = 1$ 时称为线性收敛;

(2) 当 $p = 2$ 时称为二次收敛，或平方收敛;

(3) 当 $p > 1$ 时称为超线性收敛。

- 不动点迭代中，若迭代数列 $\{x_k\}$ 收敛，且 $g'(x^*) \neq 0$，则

$$e_{k+1} = x_{k+1} - x^* = g(x_k) - g(x^*) = g'(\xi)e_k$$

取极限得 $\lim\limits_{k\to\infty}\left|\dfrac{e_{k+1}}{e_k}\right| = g'(x^*) \neq 0 \Longrightarrow$ 线性收敛.

定理3.3

设$g(x)$在$x=g(x)$的根x^*附近有连续的二阶导数，且$x_k=g(x_{k-1})$收敛于x^*, 则有：

（1）如果$g'(x^*) != 0$，则$\{x_k\}$线性收敛；

（2）如果$g'(x^*) = 0$，而$g''(x^*) !=0$, 则$\{x_k\}$平方收敛。

注意：

✓ P越大，收敛速度越快；

✓ 一个迭代格式的阶是局部概念，全局是否收敛以及收敛速度还与初值的选取有关。

例题

讨论 $x^3 - x - 1 = 0$ 在[1, 2]内根的迭代格式的收敛性和收敛速度

（1）$x_k = x_{k-1}^3 - 1$

（2）$x_k = (x_{k-1} + 1)^{\frac{1}{3}}$

解：（1） $g(x) = x^3 - 1,\ \forall x \in [1, 2],\ |g'(x)| = |3x^2| \geq 3$ 发散

（2）

$$\left.\begin{aligned} &\forall x \in [1, 2],\ |g'(x)| = \frac{1}{3} \times \frac{1}{(x+1)^{\frac{2}{3}}} \leq \frac{1}{3} \\ &g : [1,2] \to [1,2] \end{aligned}\right\}$$ 收敛

收敛速度 $x \in (1, 2)$,

$$g'(x) = \frac{1}{3} \times \frac{1}{(x+1)^{\frac{2}{3}}} \Rightarrow g'(x) > \frac{1}{3 \times 3^{\frac{2}{3}}} > 0$$ 线性收敛

3.3 Newton 迭代

基本思想：将非线性方程线性化,切线法

设$f(x)$在其零点x^*附近连续可微，x_0是$f(x)=0$的近似解。

在x_0附近用$f(x)$的一阶泰勒多项式近似$f(x)$，有

$$f(x) \approx p_1(x) = f(x_0) + f'(x_0)(x - x_0)$$

当$f'(x_0) \neq 0$时，可以取线性方程$p_1(x) = 0$的根

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

作为x^*的第1次近似值。

同理，当$f'(x_1) \neq 0$时，有$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$。

依次类推，当$f'(x_k) \neq 0$时，

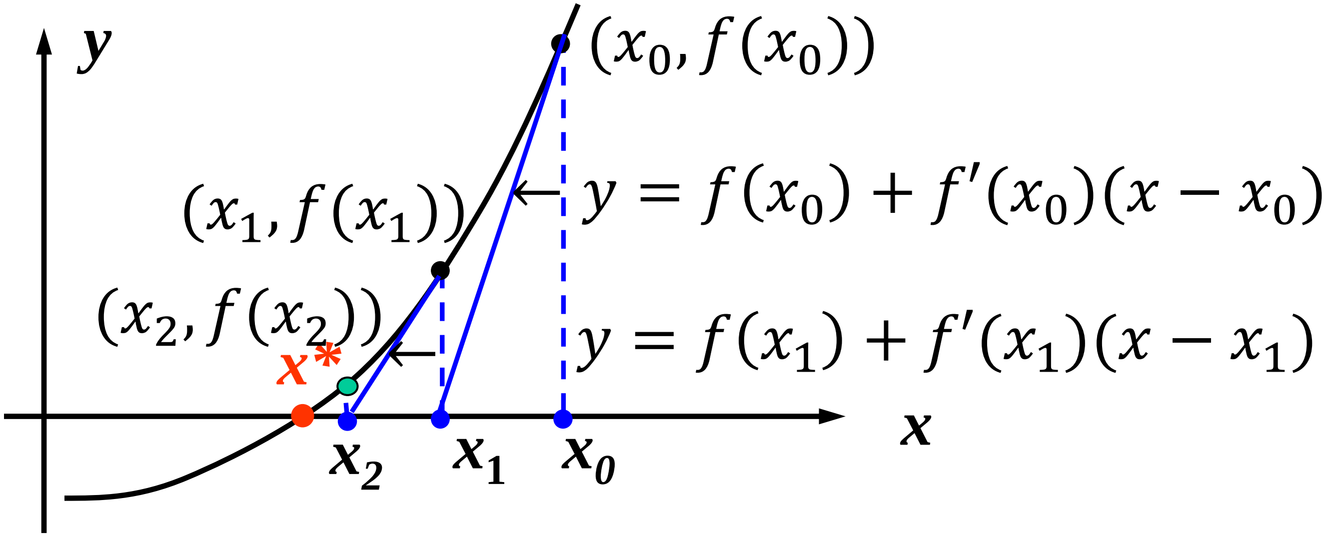
有$\boldsymbol{x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, k = 0, 1, 2, \ldots,}$（牛顿迭代公式）

作为x^*的第k+1次近似值。

迭代函数：$\varphi(x) = x - \frac{f(x)}{f'(x)}$

条件：设$f(x)$在x^*附近连续可微，$f'(x) \neq 0.$

Newton 迭代法的几何意义



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

Newton 迭代法的收敛性

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

$$\Longrightarrow g'(x^*) = \frac{f''(x^*)f(x^*)}{[f'(x^*)]^2} = 0$$

由定理3.2（局部收敛性）和定理3.3（收敛的阶），知：

牛顿迭代法局部收敛，至少具有二阶收敛速度

计算结束的条件一般为 $\boxed{|x_k - x_{k-1}| \le \varepsilon}$

例3.5 取$x_0 = 1.5$，用牛顿迭代法求解$x^3 - x - 1 = 0$, 使计算结果有4位有效数字（$\varepsilon = 0.0005$）。

解　迭代格式：$x_{k+1} = x_k - \frac{x_k^3 - x_k - 1}{3x_k^2 - 1}$

迭代次数	x_{k}	$\lvert x_k - x_{k-1} \rvert$
0	1.5	
1	1. 347826⋯	0.152174…
2	1.325200…	0.022826…
3	1.324718…	0.00048…

近似解为1.3247。

牛顿迭代法的优缺点

优点：收敛快

缺点：

每一步迭代要计算$f(x_k)$、$f'(x_k)$, 计算量大, 且有时$f'(x_k)$计算较困难；

初始近似x_0只在根x^*附近才能保证收敛，如x_0给的不合适可能不收敛。

迭代-加速技术

- 利用迭代函数$g(x)$的导数值，可以改进原迭代格式以加速迭代过程的收敛速度，甚至将发散的迭代格式加工成收敛的。

假设迭代 $x_k= g(x_{k-1})$ 收敛于方程 $x= g(x)$ 的根x^*,

记：$\overline{x_k} = g(x_{k-1})$，在$x^*$附近$g'(x) \approx D \neq 1$,

则：$x^* - \overline{x_k} = g(x^*) - g(x_{k-1}) = g'(\xi)(x^* - x_{k-1})$

$$\approx D(x^* - x_{k-1})$$

得：$x^* \approx \frac{1}{1-D}(\overline{x_k} - Dx_{k-1}) = \frac{1}{1-D}(g(x_{k-1})\text{-D } x_{k-1})$

由此预期：$x_k = \dfrac{1}{1-D}[g(x_{k-1}) - Dx_{k-1}]$

此为收敛速度更快的迭代格式，迭代加速格式。

迭代-加速技术

加速迭代函数：

$$\bar{g}(x)=\frac{\mathbf{1}}{\mathbf{1}-\boldsymbol{D}}[g(x)-Dx]$$

又因：

$$\bar{g}'(x)=\frac{\mathbf{1}}{\mathbf{1}-\boldsymbol{D}}[g'(x)-D]\approx 0$$

所以：在适当选取**D**后，可使在***x****附近, $|\bar{g}'(x)|\le|g'(x)|$

因此：基于定理3.1，加速迭代格式，收敛速度更快（**L**越小，速度越快）

迭代-加速技术

例3. 6 由迭代-加速格式改进解 $x^3-x-1=0$ 的迭代格式初值为1.5。 (1) $x_k=(x_{k-1}+1)^{\frac{1}{3}}$

解：（1）$g(x)=(x+1)^{\frac{1}{3}}, g'(x)\approx 0.2$

迭代-加速格式：$x_k=\frac{1}{1-0.2}[g(x_{k-1})-0.2x_{k-1}]$

$$=1.25[(x_{k-1}+1)^{\frac{1}{3}}-0.2x_{k-1}] \quad \text{(a)}$$

对应$\overline{g}(x)$=1.25 $[(x+1)^{\frac{1}{3}}-0.2x]$,在[1,2]上$|\overline{g}'(x)|\le 0.05$

而$|g'(x)|\ge 0.16$。故式（a）比（1）收敛速度快。

x_1 =1.3215； x_2 =1.3248； x_3 =1.3247。

第三章 迭 代 法

3.4 解线性方程组的迭代法

求解线性方程组的迭代法

- ❑ 直接法的缺点：
  - ✓**运算量大，不适合大规模的线性方程组求解;**
  - ✓**对于一些大型稀疏矩阵，浪费内存严重。**
- ❑ 迭代法：**从一个初始向量出发，按照一定的迭代格式，构造出一个趋向于真解的无穷序列。**
  - ✓**只需存储系数矩阵中的非零元素;**
  - ✓**运算量不超过** $O(kn^2)$，其中 k 为迭代步数.

迭代解法是目前求解大规模线性方程组的主要方法。

研究内容:

(1) 迭代格式的建立;

(2) 收敛性判断;

(3) 误差估计和收敛速度.

解线性方程组迭代法的基本思想

❑ 迭代格式的建立

$$\boldsymbol{Ax} = \boldsymbol{b} \Longleftrightarrow \boldsymbol{Mx} = \boldsymbol{Nx} + \boldsymbol{b}$$

$\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$

$$\Updownarrow$$

$$\boldsymbol{x} = \boldsymbol{M}^{-1}\boldsymbol{Nx} + \boldsymbol{M}^{-1}\boldsymbol{b}$$

给定一个初始向量 $x^{(0)}$，可得迭代格式：

$$\boxed{x^{(k+1)} = Gx^{(k)} + f} \qquad k = 0, 1, 2, \ldots$$

其中 G 称为迭代矩阵。

若产生的迭代序列 $\{x^{(k)}\}$ 收敛到一个确定的向量 x^*，则 x^* 就是原方程组的解。

1. Jacobi 迭代

令 $A = D + L + U$， 其中 $D = \mathrm{diag}(a_{11}, a_{22}, \ldots, a_{nn})$,

$$L = \begin{bmatrix} 0 & & & \\ a_{21} & 0 & & \\ \vdots & \ddots & \ddots & \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{bmatrix}, \qquad U = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & 0 & \ddots & \vdots \\ & & \ddots & a_{n-1,n} \\ & & & 0 \end{bmatrix}$$

如果D可逆，则可得雅可比 (Jacobi) 迭代格式:

$$\boxed{x^{(k+1)} = -D^{-1}(L+U)x^{(k)} + D^{-1}b} \qquad k = 0, 1, 2, \ldots$$

$G = -D^{-1}(L+U) = I - D^{-1}A$ **称为雅可比 (Jacobi) 迭代矩阵**

Jacobi 迭代的分量形式：

$$
\begin{cases}
x_1^{(k+1)} = \left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \ldots - a_{1n}x_n^{(k)}\right)\Big/ a_{11} \\
x_2^{(k+1)} = \left(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} - \ldots - a_{2n}x_n^{(k)}\right)\Big/ a_{22} \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
x_j^{(k+1)} = \left(b_j - a_{j1}x_1^{(k)} \ldots - a_{j,j-1}x_{j-1}^{(k)} - a_{j,j+1}x_{j+1}^{(k)} \ldots - a_{jn}x_n^{(k)}\right)\Big/ a_{jj} \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
x_n^{(k+1)} = \left(b_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \ldots - a_{n,n-1}x_{n-1}^{(k)}\right)\Big/ a_{nn}
\end{cases}
$$

或：

$$
x_i^{(k)} = \left[b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^{(k-1)}\right] / a_{ii},\ i = 1, 2, \ldots n,
$$

迭代结束条件一般用：

$\left\| x^{(k)} - x^{(k-1)} \right\| \leq \varepsilon$， ε为精度要求，$\|.\|$为某种向量范数（常用∞-范数）

$$x_i^{(k)} = \left[b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j^{(k-1)} \right] / a_{ii},\ i = 1, 2, \ldots n,$$

例题3.7 用雅克比迭代解线性方程组:

$$\begin{cases} 10x_1 - x_2 - 2x_3 = 7.2 \\ -x_1 + 10x_2 - 2x_3 = 8.3 \\ -x_1 - x_2 + 5x_3 = 4.2 \end{cases}$$

初值取$x_1^{(0)}=x_2^{(0)}=x_3^{(0)}=1$, 精度要求$\varepsilon=10^{-3}$.

Jacobi 迭代的分量形式：

$$\begin{cases} x_1^{(k+1)} = \left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \dots - a_{1n}x_n^{(k)}\right)\Big/ a_{11} \\ x_2^{(k+1)} = \left(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)}\right)\Big/ a_{22} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_j^{(k+1)} = \left(b_j - a_{j1}x_1^{(k)} \dots - a_{j,j-1}x_{j-1}^{(k)} - a_{j,j+1}x_{j+1}^{(k)} \dots - a_{jn}x_n^{(k)}\right)\Big/ a_{jj} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n^{(k+1)} = \left(b_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \dots - a_{n,n-1}x_{n-1}^{(k)}\right)\Big/ a_{nn} \end{cases}$$

在计算 $x_i^{(k+1)}$ 时，如果用 $x_1^{(k+1)},\cdots,x_{i-1}^{(k+1)}$ 代替 $x_1^{(k)},\cdots,x_{i-1}^{(k)}$，则可能会得到更好的收敛效果。此时的迭代公式为

$$\begin{cases} x_1^{(k+1)} = \left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \dots - a_{1n}x_n^{(k)}\right)\Big/ a_{11} \\ x_2^{(k+1)} = \left(b_2 \boxed{- a_{21}x_1^{(k+1)}} - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)}\right)\Big/ a_{22} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_j^{(k+1)} = \left(b_j \boxed{- a_{j1}x_1^{(k+1)} \dots - a_{j,j-1}x_{j-1}^{(k+1)}} - a_{j,j+1}x_{j+1}^{(k)} \dots - a_{jn}x_n^{(k)}\right)\Big/ a_{jj} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ x_n^{(k+1)} = \left(b_n \boxed{- a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - \dots - a_{n,n-1}x_{n-1}^{(k+1)}}\right)\Big/ a_{nn} \end{cases}$$

2. Gauss-Seidel 迭代

写成矩阵形式:

$$\begin{pmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{pmatrix} = \begin{pmatrix} 1/a_{11} & & & \\ & 1/a_{22} & & \\ & & \ddots & \\ & & & 1/a_{nn} \end{pmatrix} \left[\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} - \begin{pmatrix} 0 & & & \\ a_{21} & 0 & & \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & \cdots & 0 \end{pmatrix} \begin{pmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{pmatrix} - \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ & 0 & \cdots & a_{2n} \\ & & \ddots & \vdots \\ & & & 0 \end{pmatrix} \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{pmatrix} \right]$$

即 $x^{(k+1)} = D^{-1}\left(b - Lx^{(k+1)} - Ux^{(k)}\right)$

解得 $\boxed{x^{(k+1)} = -(L+D)^{-1}Ux^{(k)} + (L+D)^{-1}b}$ $k = 0, 1, 2, \ldots$

此迭代格式称为高斯-塞德尔 (Gauss-Seidel) 迭代

$G = -(L+D)^{-1}U$ **称为 G-S 迭代矩阵**

$$x_i^{(k)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)} \right], i = 1, 2, \ldots n,$$

3. 迭代的收敛性

■定理3.4 设迭代矩阵G的某种范数$\|G\|<1$，则 $x=Gx+f$ 存在唯一解，且对任意初值，迭代序列

$$x^{(k)}=Gx^{(k-1)}+f$$

收敛于x^*，进一步有误差估计式

$$\left\|x^{(k)}-x^*\right\|\le\frac{\|G\|}{1-\|G\|}\left\|x^{(k)}-x^{(k-1)}\right\|\le\frac{\|G\|^k}{1-\|G\|}\left\|x^{(1)}-x^{(0)}\right\|$$

后验估计 先验估计

证明思路：(1)解的存在唯一性; (2)解的收敛性; (3)误差估计式。

直接从Ax=b判断

- **推论3.1** 若A按行严格对角占优($|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|$), 则解Ax=b的Jacobi迭代和Gauss-Seidel迭代均收敛。

- 谱半径ρ(G)：G的特征值的模的最大值

- 引理 3 . 2 设 G 是方阵，则$G^k \to 0 \Leftrightarrow \rho(G)<1$.

- 定理3. 5 迭代$x^{(k)}= Gx^{(k-1)} + f$ 对任意初值收敛 $\Leftrightarrow \rho(G)<1$.

三种方法比较

- **方法一 (推论3.1): 从系数矩阵A判断, A严格对角占优, 则Jacobi迭代和Gauss-Seidel迭代收敛, 充分条件, 最方便**

- **方法二(定理3.4): 从迭代矩阵G判断, 有一种范数 $\|G\|<1$, 充分条件**

- **方法三(定理3.5): 从迭代矩阵G判断,谱半径ρ(G) <1, 充分必要条件, 最宽，缺点：求特征值，计算困难**

例3.8 判断Gauss-Seidel迭代求解 $A_i x=b$ 的收敛性。

4. 加速迭代--SOR 迭代

在 GS 迭代中

$$x_i^{(k)}=\left(b_i-a_{i1}x_1^{(k)}-\cdots-a_{i,i-1}x_{i-1}^{(k)}-a_{i,i+1}x_{i+1}^{(k-1)}-\cdots-a_{i,n}x_n^{(k-1)}\right)\Big/a_{ii}$$

$$=\left(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(k-1)}\right)\Big/a_{ii}$$

为了得到更好的收敛效果，可选参数ω作$x_i^{(k-1)}$与$x_i^{(k)}$两步的加权平均，于是就得到逐次超松弛迭代法，简称 SOR迭代，其中 ω 称为松弛因子。收敛的必要条件0<ω<2。 此时

$$x_i^{(k)}=(1-\omega)\,x_i^{(k-1)}+\frac{\omega}{a_{ii}}\left[b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(k-1)}\right]$$

低松弛法: 0<ω<1 ； ω=1: Gauss-Seidel迭代；超松弛法: 1<ω<2

Jacobi、GS 和 SOR 算法

- **Jacobi 算法**

$$\boxed{x^{(k)} = -D^{-1}(L+U)x^{(k-1)} + D^{-1}b}$$ 向量

$$\boxed{x_i^{(k)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k-1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}\right)\Big/ a_{ii}}$$ 分量

- **GS 算法**

$$\boxed{x^{(k)} = -(L+D)^{-1}Ux^{(k-1)} + (L+D)^{-1}b}$$ 向量

$$\boxed{x_i^{(k)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}\right)\Big/ a_{ii}}$$ 分量

- **SOR 算法**

$$\boxed{x^{(k)} = (D+\omega L)^{-1}\left[(1-\omega)D - \omega U\right]x^{(k-1)} + \omega(D+\omega L)^{-1}b}$$

$$\boxed{x_i^{(k)} = (1-\omega)x_i^{(k-1)} + \omega\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}\right)\Big/ a_{ii}}$$

举例

- **例：解线性方程组** $\begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \\ -5 \end{bmatrix}$

$$x^* = \begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix}$$

取初始向量 $x^{(0)} = (0, 0, 0)$，迭代过程中保留小数点后4位。

解： Jacobi 迭代格式

$$\begin{cases} x_1^{(k)} = \left(1 + x_2^{(k-1)}\right)/2 \\ x_2^{(k)} = \left(8 + x_1^{(k-1)} + x_3^{(k-1)}\right)/3 \\ x_3^{(k)} = \left(-5 + x_2^{(k-1)}\right)/2 \end{cases}$$

令 $x = (x_1, x_2, x_3)^T$ 则迭代得：

$$x^{(1)} = (0.5000, 2.6667, -2.5000)^T$$

$$\vdots$$

$$x^{(21)} = (2.0000, 3.0000, -1.0000)^T$$

举例（续）

GS 迭代格式

$$\begin{cases} x_1^{(k)} = \left(1 + x_2^{(k-1)}\right)\Big/2 \\ x_2^{(k)} = \left(8 + x_1^{(k)} + x_3^{(k-1)}\right)\Big/3 \\ x_3^{(k)} = \left(-5 + x_2^{(k)}\right)\Big/2 \end{cases}$$

得 $x^{(1)} = (0.5000, 2.8333, -1.0833)^T$

$$\vdots$$

$x^{(9)} = (2.0000, 3.0000, -1.0000)^T$

举例（续）

SOR 迭代格式

$$\begin{cases} x_1^{(k)} = (1-\omega)x_1^{(k-1)} + \omega\left(1 + x_2^{(k-1)}\right)/2 \\ x_2^{(k)} = (1-\omega)x_2^{(k-1)} + \omega\left(8 + x_1^{(k)} + x_3^{(k-1)}\right)/3 \\ x_3^{(k)} = (1-\omega)x_3^{(k-1)} + \omega\left(-5 + x_2^{(k)}\right)/2 \end{cases}$$

取 $\omega = 1.1$，得

$$x^{(1)} = (\ 0.5500,\ 3.1350,\ -1.0257\ )^T$$

$$\vdots$$

$$x^{(7)} = (\ 2.0000,\ 3.0000,\ -1.0000\ )^T$$

如何确定SOR迭代中的最优松弛因子是一件很困难的事。

线性方程组的迭代法与直接法（第二章）

- **优势**
  - **对大型稀疏方程组，可利用分量形式/稀疏矩阵的方式存储，节省空间**
- **存在问题：**
  - **可能不收敛**

P61-63，算法及程序

线性方程组的迭代特点
---与非线性方程相比

- 复杂之处
  - 多维问题：计算量大、理论分析需要矩阵和向量理论
- 简单之处
  - 线性问题，理论性质好：
    - ✓ 存在收敛性的充分必要条件
    - ✓ 迭代收敛性仅与方程组稀疏矩阵有关，与右端无关
    - ✓ 迭代收敛性不依赖于初始量的选取

非线性方程组的求解----Matlab

本章作业

习题：2,3,7,10