零点

二分法

$$x_{n+1} - x^* = \frac{x_0 - x^*}{2^{n+1}}$$

牛顿法

$$x_{n+1}=\phi(x_n)=x_n-\frac{f(x_n)}{f^{(1)}(x_n)}$$

收敛速度判断

由于

$$\begin{aligned}\phi(x)&=x-\frac{f(x)}{f^{(1)}(x)}\\ \phi^{(1)}(x)&=1-\frac{f^{(1)}(x)^2-f(x)f^{(2)}(x)}{f^{(1)}(x)^2}\\ &=\frac{f(x)f^{(2)}(x)}{f^{(1)}(x)^2}\end{aligned}$$

且在零点 x^* 处

$$f(x^*)=0$$

得

$$\phi^{(1)}(x^*)=0$$

根据泰勒展开，将$\phi(x_n)$在零点x^*展开

$$\phi(x_n)=\phi(x^*)+\phi^{(1)}(x^*)(x_n-x^*)+\frac{1}{2!}\phi^{(2)}(x^*)(x_n-x^*)^2+\dots$$

且

$$\begin{aligned}x_{n+1}&=\phi(x_n)\\ x^*&=\phi(x^*)\\ \phi^{(1)}(x^*)&=0\end{aligned}$$

得

$$\begin{aligned}x_{n+1}-x^*&=\phi(x_n)-\phi(x^*)\\ &\approx\phi^{(2)}(x^*)(x_n-x^*)^2\\ &\approx\phi^{(2)}(x^*)(\phi^{(2)}(x^*)(x_{n-1}-x^*)^2-x^*)^2\\ &=\dots\dots\\ &\approx(x_0-x^*)^{2^{n+1}}\end{aligned}$$

即

$$x_{n+1}-x^*\approx(x_0-x^*)^{2^{n+1}}$$

所以，牛顿法满足2阶收敛，收敛速度较快

重根问题

$f(x)$有m个重根，则$f(x)$可写作

$$f(x) = (x - x^*)^m$$

且

$$f^{(1)}(x^*) = f^{(2)}(x^*) = \ldots = f^{(m-1)}(x^*) = 0$$

此时，对于$\phi(x)$，有

$$\begin{aligned}\phi(x) &= x - \frac{f(x)}{f^{(1)}(x)} \\ &= x - \frac{x - x^*}{m} \\ \phi^{(1)}(x) &= 1 - \frac{1}{m}\end{aligned}$$

由于$\phi^{(1)}(x)$不为0，牛顿法退化为线性收敛(二分法)

改进

可改写递推式：

$$\begin{aligned}\phi(x) = x - m\frac{f(x)}{f^{(1)}(x)} = x^* \\ \phi^{(1)}(x) = 0\end{aligned}$$

方程组求解

高斯消去法

上三角矩阵U与下三角矩阵L

计算

$$\begin{aligned}Ux = b \\ x_k = b_k - \sum_{j=k+1}^{n} U_{k,j}x_j\end{aligned}$$

```
for (int k = n; 1 <= k; k--)
    for (int j = k+1; j <= n; j++)
        x[k] = b[k] - U[k][j] * x[j]
```

计算

$$\begin{aligned}Lx = b \\ x_k = b_k - \sum_{j=1}^{k-1} L_{k,j}x_j\end{aligned}$$

```
for (int k = 1; k <= n; k++)
    for (int j = 1; j <= k-1; j++)
        x[k] = b[k] - L[k][j] * x[j]
```

原理

对于

$$Ax = b$$

使用高斯变换，乘以下三角矩阵L

$$LAx = Lb$$
$$Ux = Lb$$
$$x = U^{-1}Lb$$

因此关键在于，将系数矩阵A转化为上三角矩阵U

$$LA = U$$

即，将矩阵A分解为下三角矩阵L和上三角矩阵之积U

$$A = LU$$

称为矩阵LU分解

$$a_{k,p} = a_{k,p} - \frac{a_{k,j}}{a_{j,j}} * a_{j,p}$$

```
gauss_lu(double* a, int n)--> void {
    for(j = 1; j <= n; ++j){
        for(k = j+1; k <= n; ++k){
            temp = a[k][j] = a[k][j]/a[j][j];
            for(p =j+1; p <= n; ++p)
                a[k][p] = a[k][p] − a[j][p] ∗ temp;
        }
    }
}
```

列选主元高斯消去法

交换行：选取一列中最大值，然后交换。避免很大的数字对结果产生影响

```
guass_lu(double* a, int* p, int n)--> void
{
    p[1,n] = {1, 2, · · · , n};
    for(j = 1; j <= n; ++j) {
        // 找到a[j][j]到a[n][j]的最大值a[t][j]
        swap(a[j], a[t]);
        swap(p[j], p[t]);
        for(k = j+1; k <= n; ++k) {
            temp = a[k][j] = a[k][j]/a[j][j];
            for(s = j+1; s <= n; ++s)
                a[k][s] = a[k][s] − a[j][s] ∗ temp;
        }
```

```
}
```

完全选主元高斯消去法

交换行和列

误差分析

具体内容见讲义

前向误差

由高斯法求解得到x的近似值

$$x^*$$

判断相对误差

$$\frac{\delta(x)}{x} = \frac{x^* - x}{x}$$

的上界

向后舍入误差分析

存在ΔA满足

$$(A + \Delta A)x^* = b$$

上三角矩阵方程组数组解的舍入误差分析

$$|\Delta b| = |Ux^* - b|$$

列选主元高斯消去法*LU*分解的舍入误差分析

$$|\Delta A| = |A - LU|$$

完全选主元高斯消去法*LU*分解的舍入误差分析

$$|\Delta A| = |A - LU|$$

列选主元高斯消去法解方程组舍入误差分析

$$|\Delta b| = |Ax^* - b|$$

范数

将向量和矩阵映射到实数范围，比较大小

向量范数

- 壹范数：绝对值相加

$$\|x\|_1 = \sum_{j=1}^{n} |x_j|$$

- 贰范数：平方值相加开根

$$\|x\|_2 = \sqrt{\sum_{j=1}^{n} |x_j|^2}$$

- *p*范数：p次方相加开p次根

$$\|x\|_p = \sqrt[p]{\sum_{j=1}^{n} |x_j|^p} = (\sum_{j=1}^{n} |x_j|^p)^{\frac{1}{p}}$$

- 无穷范数：最大值

$$\|x\|_\infty = max_{1\leq j\leq n}\{|x_j|\}$$

矩阵范数

- 行范数(无穷范数)

$$\|A\|_\infty = max_{1\leq j\leq n} \sum_{k=1}^{n} |a_{jk}|$$

- 列范数(壹范数)

$$\|A\|_1 = max_{1\leq k\leq n} \sum_{j=1}^{n} |a_{jk}|$$

- 贰范数(欧几里得范数、谱范数)

$$\|A\|_2 = \sqrt{\lambda_{max}(A^T A)} = \sqrt{\rho(A^T A)}$$

- *p*范数

$$\|A\|_p = max_{\|x\|_p=1} \|Ax\|_p$$

- *F*范数

$$\|A\|_F = \sqrt{\sum_{j,k=1}^{n} a_{jk}^2} = \sqrt{tr(A^T A)} = \sqrt{\sum_{j=1}^{n} \lambda_j(A^T A)}$$

迭代法

计算

$$Ax = b$$

使用迭代法

收敛判断

对于

$$x = Gx + b$$
$$\begin{cases} x^0 = 随机选取 \\ x^{k+1} = Gx^k + b \end{cases}$$

上面迭代格式的充要条件是$\rho(G) < 1$

Jacobi迭代

$$A = L + U + D$$
$$x = -D^{-1}(L+U)x + D^{-1}b$$

计算格式

$$x_0 = \text{初始向量}$$
$$x_{k+1} = D^{-1}(b-(L+U)x_k), k = 0, 1, 2...$$

如果A主对角线元行占优(或列占优)，则上面迭代格式收敛

Gauss-Seidel迭代

$$A = L + U + D$$
$$(L+D)x = -Ux + b$$

计算格式

$$x_0 = \text{初始向量}$$
$$x_{k+1} = D^{-1}(b - Ux_k - Lx_{k+1}), k = 0, 1, 2...$$

如果A

1. 主对角线元占优；或
2. 对称正定

则上面迭代格式收敛

代码

```
见讲义
```

逐次松弛迭代

使用*Gauss-Seidel*迭代的求解方向，并使用过松弛以加快收敛速度

令*w*是一个实数，将x_{k+1}定义为*w*乘上*Gauss-Seidel*公式和$1-w$乘上当前估计x_k的平均。

*w*称为松弛参数，$w > 1$时称为过松弛

$$Dx_{k+1} = (1-w)Dx_k + w(b - Lx_{k+1} - Ux_k)$$

计算格式

$$x_0 = \text{初始向量}$$
$$x_{k+1} = (wL+D)^{-1}[(1-w)Dx_k - wUx_k] + w(D+wL)^{-1}, b, k = 0, 1, 2...$$

如果*A*

1. 主对角线占优并且$w\epsilon(0,1]$；或者

2. 对称正定并且$w\epsilon(0,2)$

则上面迭代格式收敛

代码

```
见讲义
```

用于对称正定矩阵的方法

对称正定矩阵

对称矩阵只有一半数量的独立元素

$$A^T=A$$

是否能以一半的计算代价，并且仅仅使用一半的内存来求解

楚列斯基分解

如果A是对称正定矩阵，则存在上三角矩阵R满足

$$A=R^TR$$

共轭梯度方法

共轭

定义A内积

$$(v,w)_A=v^TAw$$

当$(v,w)_A=0$时，向量v和w为A共轭

计算格式

$$\begin{gathered}
x_0=\text{初始估计}\\
p_0=r_0=b-Ax_0\\
for\ k=0,1,2,...\\
\alpha_k=\frac{r_k^Tr_k}{p_k^TAp_k}\\
x_{k+1}=x_k+\alpha_kp_k\\
r_{k+1}=r_k-\alpha_kAp_k\\
\beta_k=\frac{r_{k+1}^Tr_{k+1}}{r_k^Tr_k}\\
p_{k+1}=r_{k+1}+\beta_kp_k
\end{gathered}$$

向量x_k是第k步时的近似解. 向量r_k表示近似解x_k的余项

变量p_k表示用于更新x_k得到改进的x_{k+1}时所使用的新的搜索方向

对于α_k和β_k的选择

1. 选择α_k使得新的余项r_{k+1}和方向p_k正交，保证下一余项向量和前面所有的余项向量都正交，即$(r^{k+1}, r^k) = 0$
   $r^0, r^1, \ldots, r^{n-1}$是正交的
2. 选择β_k，保证p_{k+1}和$p_k A$共轭，即$(p_{k+1}, p_k)_A = 0$
   $p^0, p^1, \ldots, p^{n-1}$是$A$共轭的

代码

```
见讲义
```

插值

考虑函数

$$y = f(x)$$

在上面取点

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

我们希望通过这些有限的点构造多项式，来模拟函数$f(x)$，即取值的逆过程，我们称为**插值**

拉格朗日插值

令

$$L_k(x) = \frac{(x - x_1)\ldots(x - x_{k-1})(x - x_{k+1})\ldots(x - x_n)}{(x_k - x_1)\ldots(x_k - x_{k-1})(x_k - x_{k+1})\ldots(x_k - x_n)}$$
$$L_k(x) = \begin{cases} 0 & x \neq x_k \\ 1 & x = x_k \end{cases}$$

构造多项式

$$P_{n-1}(x) = y_1 L_1(x) + \cdots + y_n L_n(x)$$

满足

$$P_{n-1}(x_k) = y_k$$

牛顿差商

令

$$
f[x_k] = f(x_k) \\
f[x_1,\dots,x_k] = \frac{f[x_2,\dots,x_k] - f[x_1,\dots,x_{k-1}]}{x_k - x_1}
$$

构造多项式

$$P(x) = f[x_1] + f[x_1 x_2](x - x_1) + f[x_1 x_2 x_3](x - x_1)(x - x_2) + \dots + f[x_1 \dots x_n](x - x_1)\dots(x - x_{n-1})$$

插值误差

假设$P(x)$是$n-1$或者更低阶的插值多项式，拟合n个点$(x_1,y_1),\dots,(x_n,y_n)$，则误差满足

$$
f(x) - P(x) = (x - x_1)(x - x_2)\dots(x - x_n)\frac{f^{(n)}(c)}{n!} \\
min(x_1,\dots,x_n) \le c \le max(x_1,\dots,x_n)
$$

龙格现象

极端的“多项式扭动”，插值次数越高，插值结果越偏离原函数的现象

切比雪夫插值

考虑误差公式中的分子

$$(x - x_1)(x - x_2)\dots(x - x_n)$$

找到特定的$x_1,x_2,\dots,x_n$，使得该式足够小

切比雪夫多项式

定义n阶切比雪夫多项式

$$T(n) = cos(n \arccos x)$$

得到递推式

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

T_n主导系数为2^{n-1}，满足

$$deg(T_n) = n$$

即x的最高次数为n次，T_n有n个根，因此可写为

$$T_n(x) = 2^{n-1}(x - x_1)(x - x_2)\dots(x - x_n)$$

区间的变化

[-1,1]区间

$$
x_i = cos\frac{(2i-1)\pi}{2n}, i = 1, \ldots, n
$$
$$
(x-x_1)(x-x_2)\ldots(x-x_n) = \frac{T_n(x)}{2^{n-1}}
$$

[a,b]区间

$$
x_i = \frac{b+a}{2} + \frac{b-a}{2}cos\frac{(2i-1)\pi}{2n}, i = 1, \ldots, n
$$
$$
|(x-x_1)(x-x_2)\ldots(x-x_n)| \le \frac{(\frac{b-a}{2})^n}{2^{n-1}}
$$

样条插值

在多项式插值中，多项式给出的单一公式满足所有数据点。而样条使用多个公式，其中每个都是低阶多项式，来通过所有数据点

我们定义*n*次样条，*n*为*x*的最高阶数

$$
S_k(x) = y_k + c_1(x-x_k)^1 + c_2(x-x_k)^2 + \cdots + c_n(x-x_k)^n
$$

三次样条

给定*n*个点

$$
(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)
$$

通过这些点的三次样条$S(x)$是一组三次多项式：

$$
\begin{gathered}
S_1(x) = y_1 + b_1(x-x_1) + c_1(x-x_1)^2 + d_1(x-x_1)^3 \\
S_2(x) = y_2 + b_2(x-x_2) + c_2(x-x_2)^2 + d_2(x-x_2)^3 \\
\vdots \\
S_{n-1}(x) = y_{n-1} + b_{n-1}(x-x_{n-1}) + c_{n-1}(x-x_{n-1})^2 + d_{n-1}(x-x_{n-1})^3
\end{gathered}
$$

满足：

- 性质一：通过数值点

$$
S_i(x_i) = y_i, S_i(x_{i+1}) = y_{i+1}, \text{其中} i = 1, \ldots, n-1
$$

- 性质二：相邻的样条段斜率相同

$$
S_i^{(1)}(x_i) = S_{i-1}^{(1)}(x_i), \text{其中} i = 2, \ldots, n-1
$$

- 性质三：相邻的样条段曲率相同

$$
S_i^{(2)}(x_i) = S_{i-1}^{(2)}(x_i), \text{其中} i = 2, \ldots, n-1
$$

求解

对于通过*n*个点的三次样条方程组，一共有$3(n-1) = 3n-3$个未知数

性质一包含$n-1$个方程，性质二和性质三各包含$n-2$个方程，一共有$n-1+2(n-2)=3n-5$个方程

所以一共有无穷个解。可以添加额外的方程来使解唯一化

自然样条

满足

$$S_1^{(2)}(x)=S_n^{(2)}(x)=0$$

这两个附加条件的三次样条被称为自然三次样条，这种条件称为自然样条的端点条件

通过更改端点条件可以得到不同的样条

贝塞尔曲线

贝塞尔样条允许用户控制节点处斜率，但是不再保证在节点导数的平滑性

最小二乘

考虑方程组

$$Ax=b$$

当解不存在时，可以试着找到近似解，即最小二乘近似

最小二乘解

法线方程

$$Ax=b$$

方程组可写为

$$x_1v_1+x_2v_2+...+x_nv_n=b$$

可以把b看作是A的列向量v_i的线性组合，对应的组合系数是$x_1,...,x_n$

即向量$v_1,v_2,...,v_n$生成的空间中找到组合系数$x_1,...,x_n$，使b在这个空间上

当解不存在时，在向量空间A={$v_1,v_2,...,v_n$}中存在与b最接近的点：
即存在$\overline{x}$，满足

$$b-A\overline{x}\perp_{\text{平面}}\{Ax|x\epsilon R^n\}$$

把垂直性表示为矩阵的乘法，对于R^n上所有的x，

$$(Ax)^T(b-A\overline{x})=0$$

可写为

$$x^T A^T (b - A\overline{x}) = 0$$

这意味着n维向量$A^T(b - A\overline{x})$和R^n中其他的n维向量垂直，这表明

$$A^T (b - A\overline{x}) = 0$$

即

$$A^T A\overline{x} = A^T b$$

称为法线方程，它的解$\overline{x}$是方程组$Ax = b$的最小二乘解

余项

最小二乘解$\overline{x}$的余项

$$r = b - A\overline{x}$$

大小度量：

- 向量的欧氏长度(2范数)

$$\| r \|_2 = \sqrt{r_1^2 + \cdots + r_m^2}$$

- 平方误差

$$SE = r_1^2 + \cdots + r_m^2$$

- 平均平方根误差

$$RMSE = \sqrt{\frac{SE}{m}} = \sqrt{\frac{r_1^2 + \cdots + r_m^2}{m}}$$

QR分解

Gram-Schmidt正交

对一组向量正交化：给定一组输入的m维向量，找出正交坐标系统，获得由这些向量张成的空间

对于A向量和B向量，找到正交单位向量：

$$\begin{aligned} q_1 &= A \\ q_1 &= \frac{q_1}{\|q_1\|_1} \end{aligned}$$

接着找到B向量在A向量上的投影，然后用B向量减去该投影即可得到与A向量垂直的向量：

$$q_2 = B - q_1(q_1^T B)$$

然后转化为单位向量

$$q_2 = \frac{q_2}{\|q_2\|_2}$$

推广到多维向量，令$A_1, \ldots, A_n$是R^m中的线性无关向量：

$$y_j = A_j - q_1(q_1^T A_j) - q_2(q_2^T A_j) - \cdots - q_{j-1}(q_{j-1}^T A_j)$$
$$q_j = \frac{y_j}{\|y_j\|_2}$$

对于上面的结果，引入新的符号：

$$r_{jj} = \| y_j \|_2$$
$$r_{ij} = q_i^T A_j$$

则

$$A_1 = r_{11} q_1$$
$$A_2 = r_{12} q_1 + r_{22} q_2$$
$$\cdots$$
$$A_j = r_{1j} q_1 + \cdots + r_{j-1,j} q_{j-1} + r_{jj} q_j$$

即

$$A_{m\times n} = Q_{m\times n} R_{n\times n}$$

称为消减QR分解

$$A_{m\times n} = Q_{m\times m} R_{m\times n}$$

称为完全QR分解

改进：用y代替A_j

$$y = A_j - q_1(q_1^T y) - q_2(q_2^T y) - \cdots - q_{j-1}(q_{j-1}^T y)$$

Household变换

*Household*反射子是正交矩阵，通过$m-1$维平面反射m维向量。

即给定一个向量x，重新找出一个相同长度的向量w，计算*Household*反射得出矩阵H满足

$$Hx = w$$

定义向量$v = w - x$，考虑投影矩阵

$$P = \frac{vv^T}{v^T v}$$

对于任何向量u，Pu是u在v上的投影

令$H = I - 2P$，则

$$Hx = x - 2Px = w - v - \frac{2vv^T}{v^T v} x = w - \frac{vv^T(w+x)}{v^T v} = w$$

矩阵H被称为**Household反射子**

实现QR分解：

$$H_3H_2H_1A = R$$
$$A = H_1H_2H_3R = QR$$

最小二乘实现

实现最小二乘:
求解A的完全QR分解

$$A_{mn} = Q_{mm}R_{mn}$$

代入$Ax = b$，得

$$R_{mn}x_{n1} = Q_{mm}^T b_{m1}$$

即

$$\begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \\ 0 & \dots & \dots & 0 \\ \vdots & & & \vdots \\ 0 & \dots & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = (q_1|\dots|q_m) \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

化简，得:

$$\begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \\ 0 & \dots & \dots & 0 \\ \vdots & & & \vdots \\ 0 & \dots & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_m \end{bmatrix}$$

分为两部分

$$\begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} 0 & \dots & \dots & 0 \\ \vdots & & & \vdots \\ 0 & \dots & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_{n+1} \\ \vdots \\ d_m \end{bmatrix} \tag{2}$$

简写为

$$R_{nn}\overline{x}_{n1} = d_{n1} \tag{1}$$

解$\overline{x}$即为最小二乘解

误差向量e满足

$$e = Ax - b = QRx - b = Rx - Q^T b = - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ d_{n+1} \\ \vdots \\ d_m \end{bmatrix} \quad (2)$$

最小二乘误差大小为

$$\| e \|_2^2 = d_{n+1}^2 + \cdots + d_m^2$$