Chapter 3 函数逼近

张亚楠[*]

March 13, 2024

问题提出

1 已知不共线的$n(n \gg 2)$个点$\{f(x_j)\}_{j=0}^{n}$, 如何确定一条可信的直线?

2 已知$f(x)$ 属于某一个函数空间A，例如$f(x) \in \boldsymbol{C}[a,b]$，出于某种考虑，希望找到简单函数$p(x)$ 属于A的子空间B (例如B 是所有次数不超过n次多项式)；要求 $p(x)$与 $f(x)$ 在某种意义下误差最小?

Q-1 与插值有何区别?

Q-2 如何定义误差最小? 函数间距离（$\|\cdot\|$) 最小！

Q-3 定义距离之后，问题变为：给定集合A中的一个点f，在A的子集B中找一个点p, s.t. $\|f-p\|$ 取最小。 若存在，如何给出?

1 线性空间

常见函数空间:

1. 连续函数空间 $\mathrm{C}_{[a,b]}$
2. 多项式空间
3. n维向量空间 $\mathbb{R}^n$.

名词回顾：线性赋范空间、内积空间；范数、内积； 三角不等式；Cauchy-Schwarz不等式。

距离(范数)和内积在逼近论里有何用处? 距离表示远近，内积定义角度 ！！

[*]ynzhang@suda.edu.cn 苏州大学数学科学学院

1.1 线性赋范空间

定义 1 设S是一个线性空间，$x \in S$, 若存在非负实数$\|\cdot\|$满足

1. 正定性：$\|x\| \geq 0$, 且 $\|x\| = 0 \Leftrightarrow x = 0$.
2. 齐次性：$\|\alpha \cdot x\| = |\alpha| \cdot \|x\|, \alpha \in \mathbb{C}$
3. 三角不等式: $\|x + y\| \leq \|x\| + \|y\|$.

此时称 $\|\cdot\|$ 是S上的一个范数(模、长度)； 空间S 和范数$\|\cdot\|$ 一起称为“线性赋范空间”。

连续函数空间$C[a,b]$常用范数

1) 连续(无穷)范数：$\|f\|_c = \|f\|_\infty := \max\limits_{a \leq x \leq b} |f(x)|$

2) L_p范数：$\|f\|_p = \sqrt[p]{\int_a^b |f(x)|^p dx}$

有限维空间$\boldsymbol{R}^n$常用范数

$$\|x\|_p = \sqrt[p]{\sum_{k=1}^{n} |x_k|^p},\ 1 \leq p \leq \infty$$

$p = 2$时即是最为常用的欧几里德范数或简称欧氏范数.

1.2 内积空间

定义 2 设X是数域F(例如实数或者复数)上的线性空间，对$\forall x, y \in X$, 定义一个二元运算$(\cdot,\cdot)$, 若其满足:

- 共轭对称性：$(x,y) = \overline{(y,x)}$.
- 关于第二个分量是线性的[1]. 对$a, b \in \mathbb{C}$,
$$(x, ay_1 + by_2) = a(x, y_1) + b(x, y_2)$$
- 正定性
$$(x,x) \geq 0$$
等号成立当且仅当$x = 0$.

此时称$(\cdot,\cdot)$为内积 inner product. 定义了内积的线性空间称为内积空间.

例 1 内积空间举例: $L^2(\Omega)$, $\mathbb{R}^n$, $\mathbb{C}^n$

$$(f,g) = \int_\Omega f(x)\overline{g(x)}dx, \quad (x,y) = x^{\mathrm{H}} \cdot y;$$

这里横线表示取共轭；上标H表示共轭转置。 本课程不做特殊说明的话，都考虑实内积.

[1]不同的教材对哪个变元规定线性，另一个变元规定共轭线性是不统一的！若按照这里的规定， 则$\langle x,y \rangle = x^{\mathrm{H}}y$, 此时形式是自然的。反过来， 规定第一个变元为线性，第二个变元为共轭线性，此时复内积定义为$\langle x,y \rangle = x^{\mathrm{T}}\bar{y}$, 按照严质彬教授的说法：这是丑陋的，不能被接受的！

例 2 (Cauchy-Schwarz不等式) 任意的内积满足如下不等式

$$|(x,y)| \le \sqrt{(x,x)}\sqrt{(y,y)}$$

证明:(实内积,简单情况);对任意实数α,

$$(x+\alpha y, x+\alpha y) = (x,x) + 2\alpha(x,y) + \alpha^2(y,y) \ge 0$$

取$\alpha = -(x,y)/(y,y)$ 检验即可。

若X以及其定义的内积和范数,既是内积空间又是Banach空间(完备的距离空间),则称之为Hilbert空间。 Hilbert空间比起线性赋范空间有何优势呢? Hilbert空间即可定义距离(长度)

$$\|x\| = \sqrt{(x,x)}$$

又可定义角度

$$\cos\theta = \frac{(x,y)}{\|x\|\cdot\|y\|}$$

有了距离,可以定义最佳逼近(距离最小);有了角度,可以定义垂直(正交), 直观上也可以更好利用几何知识来理解投影距离最小。“正交”这一名词在数值计算中非常重要, 例如正交矩阵,正交多项式。从数值计算上看,可简单理解为“正交” 就是好的、稳定的!!

定义 3 带权内积的权函数:设非负函数$\rho(x)$满足:

(1) $\int_a^b x^k \rho(x)dx$ 存在且为有限值.

(2) 对非负连续函数 $g(x)$, 如果 $\int_a^b g(x)\rho(x)dx = 0$, 则 $g(x) \equiv 0$

连续函数的带权 L^2 内积及其范数(若不引起歧义,可略去下标)

$$(f,g)_\rho = \int_a^b \rho(x)f(x)g(x)dx, \quad \|f\| = \sqrt{(f,f)_\rho}$$

有限维空间的带权内积和范数: 给定$w_i > 0, x, y \in \mathbb{R}^n$, 定义

$$(x,y)_\omega = \sum_{i=1}^n w_i x_i y_i, \quad \|x\|_\omega = \sqrt{(x,x)_\omega}$$

1.3 最佳逼近的定义

记$P_n[a,b]$, 或者$H_n[a,b]$是$[a,b]$区间上所有次数不超过n的多项式组成的线性空间. 由于多项式是连续函数, 前文定义的内积,范数,带权内积以及范数均适用.

定义 4 给定$f \in C[a,b]$, 若$p^\star(x) \in H_n$, 使得

$$\|f - p^\star\| = \min_{p\in H_n} \|f - p\|$$

则称$p^\star(x)$是$f(x)$的最佳逼近多项式. 其中

1. 上式中的范数取无穷范数时，称为最佳一致逼近；

2. 取2范数时，称为最佳平方逼近.

若$f(x)$给定的是离散点值 $\{f(x_j)\}_{j=0}^m$, 则满足

$$\|f-p^\star\|^2 = \min_{p\in H_n(x)} \sum_{j=0}^m |f(x_j)-p(x_j)|^2$$

或者

$$\|f-p^\star\|^2 = \min_{p\in H_n(x)} \sum_{j=0}^m w_j|f(x_j)-p(x_j)|^2$$

的$p^\star$称为f的离散的（带权）最佳平方逼近，也称为最小二乘拟合.

Theorem 1 (Weierstrass) 设$f(x)\in C_{[a,b]}$, 则对 $\forall\varepsilon>0$, 存在代数多项式$p(x)$， s.t.,

$$\|f-p\|_c := \max_{x\in[a,b]} |f(x)-p(x)| < \varepsilon$$

标注 1 定理表明：利用多项式逼近连续函数式可行的. 从计算和理论分析的角度看， 最佳平方逼近多项式容易得到，而最佳一致逼近的理论和算法较为困难. 我们不加证明的给出 最佳一致逼近多项式的存在唯一性定理[2]—— Chebyshev定理.

设

$$p^* = \arg\min_{\forall p\in H^n} \|f-p\|_\infty$$

为f 的最佳一致逼近多项式，若x^*满足

$$|f(x^*)-p^*(x^*)| = \min_{\forall p\in H^n} \|f-p\|_\infty := \Delta(p^*),$$

则称x^* 是一个交错点。

Theorem 2 (Chebyshev定理, 最佳一致逼近交错定理) 两个结论：

1. $f(x)$ 在H^n中的最佳一致逼近多项式存在唯一

2. p^*是$f(x)$的最佳一致逼近多项式当且仅当$p^*(x)-f(x)$在至少$(n+2)$个点上取到正负 交错的偏差点.

例 3 假设X是内积空间, $\{v_j|j=1,2,...,n\}\subset X$, 记

$$G(v_1,v_2,...,v_n) = \Big(G_{jk}\Big) = \Big((v_j,v_k)\Big)$$

为格拉姆(Gramian) 矩阵, 则 $\det(G)\neq 0 \Leftrightarrow \{v_j\}$ 线性无关

[2]证明可见王仁宏《数值逼近》; Trefethen, Approximation Theory and Approximation Practice

提示：$\det(G)\neq 0$等价于齐次方程

$$\sum_{j=1}^{n}(u_j,u_k)\alpha_j=0,\ 1\leq k\leq n$$

只有零解.

$$\sum_{j=1}^{n}(u_j,u_k)\alpha_j=\sum_{j=1}^{n}(\alpha_j u_j,u_k)=\Big(\sum_{j=1}^{n}\alpha_j u_j,u_k\Big)=0,\ 1\leq k\leq n$$

$$\Big(\sum_{j=1}^{n}\alpha_j u_j,\sum_{k=1}^{n}\alpha_k u_k\Big)=0\Leftrightarrow\sum_{k=1}^{n}\alpha_k u_k=0$$

读者可按照充分必要性两方面给出证明过程，留作练习.

例 4 (多项式空间) $X=H_n$ 是n次多项式空间， $v_j=x^{j-1}$, $j=1,...,n+1$ 可作为空间X的一组基. 此时, G即是著名的Hilbert矩阵，Hilbert矩阵是坏条件数的，数值不稳定.

2 正交多项式

本节给出正交多项式的定义和一般性质.

定义 5 (正交函数和正交函数列) 若$f(x)$, $g(x)\in C[a,b]$,$\rho(x)$是$[a,b]$上的权函数且满足

$$(f,g)=\int_a^b\rho(x)f(x)g(x)dx=0$$

则称f,g在$[a,b]$上带权ρ 正交. 若有函数族ϕ_0, $\phi_1,...,\phi_n,...$满足

$$(\phi_j,\phi_k)=\int_a^b\rho(x)\phi_j\phi_k dx=\begin{cases}0, & j\neq k\\ A_k>0, & j=k\end{cases}$$

则称$\phi_k(x)$是[a,b] 上带权$\rho(x)$ 的正交函数族；若$A_k\equiv 1$, 可称为标准正交函数族

例 5 在$[-\pi,\pi]$上的三角函数族：

$$1,\cos x,\sin x,\cos 2x,\sin 2x,...$$

满足

$$\big(\sin kx,\sin kx\big)=(\cos kx,\cos kx)=\int_{-\pi}^{\pi}\cos^2 kxdx=\pi$$
$$(\sin kx,\cos kx)=\int_{-\pi}^{\pi}\sin kx\cdot\cos kxdx=0$$
$$(\sin kx,\sin jx)=\frac{1}{2}\int_{-\pi}^{\pi}[\cos(j+k)x-\cos(k-j)x]dx=0$$

定义 6 设$\phi_n(x)$是$[a,b]$上的首项系数a_n非零的n次多项式，$\rho(x)$是权函数. 如果多项式序列$\{\phi_n(x)\}_0^\infty$满足正交函数族定义，则称多项式序列$\{\phi_n(x)\}_0^\infty$在$[a,b]$上带权ρ正交；称$\phi_n(x)$为$[a,b]$上带权ρ的n次正交多项式.

例 6 给定区间[a,b]上的一族线性无关的幂函数$1, x, x^2, ..., x^n, ...$,可通过 Gram-Schmidt 正交化方法构造正交多项式序列, 表达式如下:

$$\phi_0(x) = 1$$
$$\phi_n(x) = x^n - \sum_{j=0}^{n-1} \frac{(x^n,\ \phi_j)}{(\phi_j,\ \phi_j)} \phi_j(x)$$

提示: 可逐一验证正交性，归纳法证明. ■

Theorem 3 非零的正交序列 $\{\phi_j\}_{j=0}^n$线性无关。

提示: 事实上, 若 $\sum_{j=0}^{n} c_j\phi_j(x) = 0$ 对上式作用内积ϕ_k

$$\sum_{j=0}^{n} c_j \Big(\phi_j(x),\ \phi_k(x)\Big) = 0, \quad k = 0, 1, 2..., n$$

得到$c_k = 0, \quad k = 0, 1, 2, ..., n$; 线性无关得证。 ■

(I) 非零正交多项式$\{\phi_j(x)\}$必线性无关; 可选为基函数！

(II) 对任何次数不超过 n 的多项式$p(x) \in H_n$均可表示为$\phi_j(x)$的线性组合

$$p(x) = \sum_{j=0}^{n} c_j\phi_j(x)$$

(III) $\phi_n(x)$ 与任何次数不超过$(n-1)$的多项式正交

Theorem 4 n次正交多项式$\phi_n(x)$ 在(a, b) 上有 n 个不同的零点

Proof: 假定$\phi_n(x)$在$[a, b]$上没有零点，或者所有零点全是偶数重的，则$\phi_n(x)$ 符号不变. 这与

$$(\phi_n, \phi_0) = \int_a^b \phi_n(x)\rho(x)dx = 0$$

矛盾。因此零点不可能全是 偶数重的，有奇有偶。挑出奇数重零点$x_j(j = 1, 2, ..., l)$, 设

$$a < x_1 < x_2 < \cdots < x_l < b$$

则$\phi_n(x)$ 在 x_j处变号，令

$$q(x) = (x - x_1)(x - x_2)...(x - x_l)$$

则得$\phi_n(x)q(x) \geq 0$. 进而

$$(\phi_n,\ q) = \int_a^b \rho(x)\phi_n(x)q(x)dx \neq 0$$

若$l < n$, 与正交性矛盾；则$l = n$, 得证. ■

2.1 正交多项式求根（选讲）

n 次正交多项式有n个实根，可以利用矩阵特征值的计算方法 求出正交多项式的根[3] 。

Theorem 5 (三项循环式 three term recurrence relationship) 任意正交多项式序列均满足

$$\begin{cases} p_j(x) = (a_j x - b_j)\, p_{j-1}(x) - c_j p_{j-2}(x), & j = 1, 2, \cdots, N; \\ p_{-1}(x) \equiv 0, \quad p_0(x) \equiv 1, \end{cases} \tag{1}$$

研究生留作作业. 提示: 由表达式 $xp_{j-1} = \sum_{k=0}^{j} \alpha_k p_k$, 再利用正交性推导。

改写循环式为

$$xp_{j-1} = \frac{c_j}{a_j} p_{j-2} + \frac{b_j}{a_j} p_{j-1} + \frac{1}{a_j} p_j, \quad j = 1, 2, ..., n$$

进而得到矩阵形式

$$x \cdot \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \end{bmatrix} = \begin{bmatrix} b_1/a_1 & 1/a_1 & & \\ c_2/a_2 & b_2/a_2 & 1/a_2 & \\ & \ddots & \ddots & \\ & & c_n/a_n & b_n/a_n \end{bmatrix} \cdot \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \end{bmatrix} + \frac{p_n(x)}{a_n},$$

若$p_n(x^*) = 0$, 则

$$x^* \cdot \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \end{bmatrix}_{x=x^*} = \begin{bmatrix} b_1/a_1 & 1/a_1 & & \\ c_2/a_2 & b_2/a_2 & 1/a_2 & \\ & \ddots & \ddots & \\ & & c_n/a_n & b_n/a_n \end{bmatrix} \cdot \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \end{bmatrix}_{x=x^*} = T * q,$$

上式说明x^*是矩阵T 的一个特征值，特征向量为$q = [p_0(x^*), ..., p_{n-1}(x^*)]^{\mathrm{T}}$.

相应的， $p_n(x)$的n个根也是求解三对角矩阵A的特征值！ QR算法容易计算该特征值。 为了进一步提高计算效率，可以将矩阵做相似对角化，转化为计算对阵矩阵， 也即有如下结论

命题 2.1 任意$a = [a_1, a_2, \cdots, a_{n-1}]^\top \in \mathbb{R}^{n-1}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^{n-1}$, 若满足

$$\gamma_k = a_k * c_k \geq 0$$

则 三对角矩阵

$$X = \mathrm{diag}(a, -1) + \mathrm{diag}(b) + \mathrm{diag}(c, 1), \qquad \text{and} \quad Y = \mathrm{diag}(\sqrt{\gamma}, -1) + \mathrm{diag}(b) + \mathrm{diag}(\sqrt{\gamma}, 1)$$

有相同的(实)特征值, 其中 $\sqrt{\gamma} = [\sqrt{a_1 c_1}, \sqrt{a_2 c_2}, \cdots, \sqrt{a_{n-1} c_{n-1}}]^\top$.

满足循环式(1)的正交多项式p_n, 其零点是如下对称三对角矩阵

$$S = \begin{bmatrix} b_1/a_1 & \gamma_1 & & \\ \gamma_1 & b_2/a_2 & \gamma_2 & \\ & \ddots & \ddots & \\ & & \gamma_{n-1} & b_n/a_n \end{bmatrix}$$

[3]其它方法也可以，如Newton 法

的特征值，其中$\gamma_k = \sqrt{c_{k+1}/(a_k a_{k+1})}$.

```
function SIG = eig_orth(a,b,c,n)
%      compute eigen value of orth poly
%      p_j+1 = (a*x - b)*p_j - c*p_j-1;
%      n = length(a) = length(b) = length(c); c0 = c(1)= 0
if nargin < 4,     n = length(a);     end
     a1 = c(2:n)./a(2:n);    b1 = b./a;     c1 = 1./a(1:n-1);
     gam = sqrt(a1.*c1);
     T = diag(b1) + diag(gam,1) + diag(gam,-1);    SIG = eig(T);
```

2.2 Legendre 多项式

正交多项式有许多共同的性质，比如n个互异零点，有递推式等等。 常用的有限区间内的正交多项式有 Legendre 多项式 [4] 与Chebyshev多项式。

在区间[-1,1]上权函数取常数1， 由$1, x, x^2, \ldots$ 正交化得到的即是Legendre多项式, 满足如下循环递推式和正交性质

$$\begin{cases} P_{n+1}(x) = \dfrac{2n+1}{n+1} x P_n(x) - \dfrac{n}{n+1} P_{n-1}(x), \\ P_0(x) = 1, \quad P_1(x) = x \end{cases}$$

满足 $(P_j, P_k) = 0,\ j \neq k; \qquad (P_k,\ P_k) = \dfrac{2}{2k+1}$

记$\tilde{P}_n(x)$ 是首项系数为1的Legendre多项式，也即是$P_n(x)$除以首项系数得到的多项式. 有如下结果成立。

Theorem 6 对定义在 $[-1, 1]$上的所有首项系数为1 的多项式, $\tilde{P}_n(x)$ 的L2范数最小.

证明：记 $Q_n(x)$ 是任意首项系数为1的n次多项式，则

$$Q_n(x) = \tilde{P}_n(x) + Q_{n-1}(x)$$

其中 Q_{n-1}是次数不超过$(n-1)$的多项式。则

$$\|Q_n\|^2 = (Q_n, Q_n) = (\tilde{P}_n + Q_{n-1}, \tilde{P}_n + Q_{n-1}) = (\tilde{P}_n, \tilde{P}_n) + (Q_{n-1}, Q_{n-1}) = \|\tilde{P}_n\|^2 + \|Q_{n-1}\|^2 \geq \|\tilde{P}_n\|^2$$

等号成立当且仅当$\tilde{P}_n = Q_n$. 证毕. ■

标注 2 上述结果表明：Legendre多项式与零函数的平方误差最小。从实用出发，可将Legendre多项式的零点(也称为Gauss点)插值 作为连续函数的最佳平方逼近多项式.

[4]本科生只掌握Legendre多项式的正交性和循环递推式。

以下代码用于计算Legender多项式的第0到m项。输入节点向量$x \in \mathbb{R}^N$, 正整数m; 输出为$N \times (m+1)$ 阶矩阵V, V的第$k+1$列元素为$P_k(x)$.

```
  function V = get_legend(x,m)
% for given points x, and intger m
% it computs legendreP P_k(x) on x, with degree from 0\to m
  x = x(:); N = length(x);
  V = zeros(N,m+1);   V(:,1) = 1;  V(:,2) = x;
    for j = 2:m
        V(:,j+1) = ( (2*j-1).*x.*V(:,j) - (j-1)*V(:,j-1) ) /j;
    end
```

2.3 Legendre多项式性质2(选讲)

下面考虑Legendre多项式的零点计算，由三项循环式

$$p_j(x) = (a_j x + b_j)\, p_{j-1}(x) - c_j p_{j-2}(x), \quad j = 1, 2, ..., n$$

知道Legendre多项式对应的$b_j = 0, a_j = \frac{2j-1}{j}, c_j = \frac{j-1}{j}$, 进而

$$\gamma_j = \sqrt{\frac{c_{j+1}}{a_j a_{j+1}}} = \sqrt{\frac{\frac{j}{j+1}}{\frac{2j-1}{j}\frac{2j+1}{j+1}}} = \sqrt{\frac{j^2}{(2j-1)(2j+1)}} = \sqrt{\frac{1}{4-1/j^2}}$$

则n次Legendre多项式$P_n(x)$的根可由如下算法给出

```
1 % Legendre pts via eig algorithm
2 n1 = 11;    j0 = (1:n1);  gamma = 1./sqrt(4 - 1./j0.^2);
3 S = diag(gamma,-1) + diag(gamma,1);            % matrix
4 xp = eig(S);                                   % eigenvalue = roots of Legendre
5 %
6 f = @(z) 1./(1+25*z.^2);         t = linspace(-1,1,2001)';
7 y = baryinterp0(xp,f(xp),t);                   % interp on "Legendre pts "
8 err = norm(y - f(t),inf),
9 figure(1); plot(t,f(t),'r-',t,y,'k--','LineWidth',2);
```

其中第2-4行即是计算$P_n(x)$的根；6-7行采用Legendre多项式零点插值来近似Runge函数，效果与Chebyshev点插值相当。 该例子表明：选择Chebyshev点插值和Legendre点（也称为Gauss点）插值都有较高精度 [5]。 读者应当注意到这个结果。后文在数值积分的部分，我们将继续讨论这两类正交多项式的异同。 正交多项式也是谱方法的基础，我们只给出常用的循环式.

Legendre 多项式的常用递推式

$$\mathrm{P}_{n+1}(x) = \frac{2n+1}{n+1} x \mathrm{P}_n(x) - \frac{n}{n+1}\mathrm{P}_{n-1}(x)$$
$$\mathrm{P}'_{n+1}(x) - \mathrm{P}'_{n-1}(x) = (2n+1)\mathrm{P}_n(x)$$
$$((1-x^2)P_n')' + (n+1)nP_n = 0$$

[5]对光滑函数，两种插值节点均可达到谱精度，误差估计和进一步学习可参考：Guo BenYu, Spectral Methods and Their Applications

2.4 Chebyshev 多项式

由于可以快速计算(FFT), 零点有显式表达式，Chebyshev多项式最为常用

$$T_k(x) = \cos(k \arccos x), \quad \text{或者} \quad \begin{cases} T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x), & k = 1, 2, \ldots \\ T_0 = 1, \quad T_1 = x \end{cases}$$

观察

$$T_n(x) = \cos\Big(n \arccos(x)\Big), \quad -1 \le x \le 1 \tag{2}$$

首先说明上式是多项式。记

$$T_n(x) = \cos(n\theta) \qquad \cos\theta \quad = x, \quad x \in [-1, 1]$$

利用三角恒等式

$$\cos((n+1)\theta) + \cos((n-1)\theta) = 2\cos(n\theta)\cos(\theta)$$

即得$T_{n+1}(x) + T_{n-1}(x) = 2xT_n(x)$, 结合$T_0 = 1,\ T_1 = x$, 可得循环递推多项式。说明(2)的确是多项式。 利用和差化积公式

$$\begin{aligned} &\frac{d}{dx}\left[\frac{1}{n+1}\cos\Big((n+1)\theta\Big) - \frac{1}{n-1}\cos\Big((n-1)\theta\Big)\right] \\ =&\Big[\sin(n-1)\theta - \sin(n+1)\theta\Big]\frac{d\theta}{dx} = 2\cos(n\theta)\underline{(-\sin\theta)\frac{d\theta}{dx}} = 2\cos(n\theta)\underline{(-\sin\theta)\Big/\frac{dx}{d\theta}} = 2\cos(n\theta) \end{aligned}$$

得到Chebyshev多项式导数递推式

$$2\,\mathrm{T_n}(x) = \frac{1}{n+1}\mathrm{T}'_{n+1}(x) - \frac{1}{n-1}\mathrm{T}'_{n-1}(x), \quad \mathrm{n} \ge 2$$

利用余弦函数的正交性

$$\Big(\cos mx,\ \cos nx\Big) = \int_0^\pi \cos mx \cos nx dx = \begin{cases} 0, & m \ne n \\ \frac{\pi}{2} & m = n \ne 0 \\ \pi & m = n = 0 \end{cases}$$

可得$T_n(x)$ 的(带权)正交性 （积分换元法）

$$\int_{-1}^{1} \frac{T_m(x)T_n(x)dx}{\sqrt{1-x^2}} = \begin{cases} 0 & (m \neq n) \\ \pi/2 & (m = n \neq 0) \\ \pi & (m = n = 0) \end{cases}$$

前文已知正交多项式必有n个互异零点，由表达式(2)可知其零点表达式为

$$x_j = \cos\frac{(2j-1)\pi}{2n}, \quad 1 \leq j \leq n,$$

进而得到Chebyshev的另一个表达式

$$T_n(x) = 2^{n-1}\prod_{j=1}^{n}(x - x_j)$$

注意到$T_n(x) = \cos(n\theta)$, 则

$$|T_n(x)| = |2^{n-1}\prod_{j=1}^{n}(x - x_j)| \leq 1 \quad \Rightarrow \quad \left|\prod_{j=1}^{n}(x - x_j)\right| \leq 2^{1-n}$$

上式说明首项系数为1的Chebyshev多项式的绝对值极其小。 事实上， 类似于首一的Legendre多项式$\tilde{P}_n(x)$在所有首项系数为1的多项式中，其平方范数最小； 首一的Chebyshev多项式$\tilde{T}_n(x) = 2^{1-n}T_n(x)$, 其最大模范数最小. 见下节定理7.

```
function T = get_cheb(x,m)
% for given points x and intger m
% it comput T(x) on x, with degree from 0\to m
x  = x(:);    N = length(x);
T = zeros(N,m+1);   T(:,1) = 1;  T(:,2) = x;
% % % recursive
  for j = 2:m
      T(:,j+1) = 2*x.*T(:,j) - T(:,j-1);
  end
```

标注 3 代码给出了Chebyshev多项式的前$(m+1)$项， 其中矩阵的每j列对应$T_{j-1}(x)$在节点x处的取值. 我们可称T为Chebyshev-Vandermonde矩阵， 与单项式产生的Vandermonde矩阵不同，T的条件数一般都较小(特别是取Chebyshev点等特殊点时， T几乎是正交矩阵)，因此Chebyshev多项式常被选作多项式空间中的一组基函数. 给定某个函数在多项式空间的Cheybshev展开系数， 借助于上述代码产生的矩阵T, 我们很容易计算该函数在x 处的值. 这也是 Trefethen 发展的工具包chebfun的做法——将光滑函数记录为其Chebyshev系数. 我们给出一个类似功能的函数mychebpolyval.m

```
function f = mychebpolyval(c,x)
% for given points x, and coeffs c
% it computs Tchebshev series f(x) = sum_k ck*Tk(x)
```

```
    m = length(c)-1;
    V = get_cheb(x,m);   f = V*c;
end
```

2.5 Chebyshev点插值

回顾Lagrange插值多项式余项表达式

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x-x_j)$$

上式中乘积与节点x_j选取有关系，如何选取使R_n绝对值最小?

Theorem 7 设 $\tilde{T}_n(x)$ 是首项系数为一的Chebyshev多项式(monic poly), 则

$$\max_{-1\le x\le 1}|\tilde{T}_n(x)| \le \max_{-1\le x\le 1}|p_n(x)|, \quad \forall p_n(x)\in\{\text{所有首项系数为一的n次多项式}\}$$

且

$$\max_{-1\le x\le 1}|\tilde{T}_n(x)| = \frac{1}{2^{n-1}}$$

证明： 定理表明对所有首一的n次多项式， 首一的Chebyshev多项式$\tilde{T}_n(x)$是最大模最小的. 设$p_n(x)$是任意首一的n次多项式， 则 $p_n(x) = x^n - Q_{n-1}(x)$. 若要$\|p_n\|_\infty$取最小，也即是求$f(x) = x^n$ 在空间 H^{n-1}中的最佳一致逼近多项式Q_{n-1}. 由Chebyshev定理(最佳一致逼近交错定理)，当误差函数$x^n - Q_{n-1}(x)$有$(n-1+2)$交错点时，Q_{n-1}即为所求。已知$x^n - Q_{n-1}(x) = \tilde{T}_n(x)$时， 恰有$(n+1)$个偏差点。又由于余弦函数可以取到极值点， 定理第二部分显然成立。 ■

标注 4 对任一给定的首一的$(n+1)$次多项式 $Q_{n+1}(x)$, 若$p_n^*(x)$是n次最佳一致逼近多项式， 则$R(x) = Q_{n+1} - p_n^*$必为$(n+1)$次首一的Chebyshev多项式. 借助这一结论，我们可计算Q_{n+1}的n次最佳一致逼近多项式.

$$\tilde{T}_{n+1}(x) = Q_{n+1}(x) - p_n^*(x) \quad \to \quad p_n^*(x) = Q_{n+1}(x) - \tilde{T}_{n+1}(x)$$

例 7 记$Q(x) = 3x^3 + 4x^2 + 5x + 6$, $x\in[-1,1]$, 则$Q(x)$的最佳一致逼近二次多项式为

$$p^*(x) = 3\left[\left(x^3 + \frac{4}{3}x^2 + \frac{5}{3}x + 2\right) - \tilde{T}_3(x)\right] = Q(x) - 3\tilde{T}_3(x)$$

取$\{x_j\}_{j=0}^2$为$T_3(x)$ 的零点(也是$\tilde{T}_3(x)$ 的零点)，则

$$p^*(x_j) = Q(x_j) - 3\tilde{T}_3(x_j) = Q(x_j), \quad 0\le j\le 2.$$

上式表明 $p^*(x)$与$Q(x)$在三点处函数值相等, 又因为p^*是二次多项式，则可通过插值唯一确定.

我们重述上述结论

Theorem 8 **((n+1)次多项式的n次最佳一致逼近)** 对于任意$Q(x)\in H_{n+1}[a,b]$为$n+1$次多项式， 其最佳一致逼近n次多项式 即是$Q(x)$ 的Chebyshev零点插值多项式.

```
f = @(x) 3*x.^3 + 4*x.^2 + 5*x + 6;
n = 3; j1 = 1:n; xp = cos((j1-0.5)'*pi/n);     % cheby zeros
t = linspace(-1,1,801)';  p = baryinterp0(xp,f(xp),t);
Err_cheb = norm(f(t)-p,inf),
figure; LW = 'LineWidth';  subplot(2,1,1);
plot(t,f(t), 'r-', t, p,'b-.',LW,1);  grid on;
subplot(2,1,2);  plot(t,f(t)-p, t,t*0,LW,1);
title('Abs␣error␣of␣minimax');   grid on;
```

在$[-1,1]$上取$(n+1)$个插值节点为Chebyshev多项式零点

$$x_k = \cos\frac{2k+1}{2(n+1)}\pi, \quad k = 0, 2, ..., n$$

则$\omega(x) = \prod_{j=0}^{n}(x - x_j)$ 是monic chebyshev 多项式, 进而

$$\max_{-1\le x\le 1}\left|\prod_{j=0}^{N}(x - x_j)\right| = \frac{1}{2^n}$$

Theorem 9 (Chebyshev零点插值余项) 设插值节点$\{x_j\}_{j=0}^{n}$是Chebyshev 多项式的零点, 被插值函数 $f(x) \in C^{(n+1)}[-1,1]$, 则:

$$\max_{-1\le x\le 1}|f(x) - L_n(x)| \le \frac{1}{2^n(n+1)!}\|f^{(n+1)}\|_\infty$$

标注 5 对于一般区间$[a,b]$, 可通过伸缩、平移变换得到任意区间的Chebyshev点

$$x_k = \frac{b-a}{2}\cos\left(\frac{2k+1}{2n+2}\pi\right) + \frac{a+b}{2}$$

实际应用时更多采用Chebyshev极值点插值, 也称为第二类Chebyshev点、或者 Chebyshev-Labatto 点 $x_j = \cos\frac{j\pi}{N}$, $0 \le j \le N$. Chebyshev 点插值有效避免Runge现象(李庆扬教材p65 例5.)

```
f = @(x) 1 ./ (1+25*x.^2);                 % f = @(x) abs(x) + x/2 - x.^2;
n = 211;  xp = cos((0:n)'*pi/n);          % cheby lobatto
% j1 = 1:n; xp = cos((j1-0.5)'*pi/n);     % cheby zeros
t = linspace(-1,1,1001)';  p = baryinterp0(xp,f(xp),t);
Err_cheb = norm(f(t)-p, inf),
figure; plot(t,f(t), 'r-', t,p,'b-.','LineWidth',1); grid on;
```

读者可以测试上述代码[6], 精度可达到10^{-15}. 有几点需要提醒读者注意

R-(I) 被插值函数要有表达式或者Chebyshev点的函数值需要精确给出;

R-(II) 当需要更多插值节点 ($n > 10^3$), Lagrange插值或者Barycentric 数值不稳定; Vander with Arnoldi可行, 但效率不高; 借助于余弦变换可以解决上万个节点的计算复杂度和数值稳定性问题;

R-(III) 光滑函数一般不需要过多的Chebyshev点插值; 第二条提到的情况涉及到不光滑函数 (如代码中注释的绝对值函数) ; 理论上多项式仍然可以以任意精度逼近该函数, 但是需要上万个节点 (一万次多项式) , 效率较差。非光滑函数更好的选择是有理逼近。

[6] baryinterp0.m 代码在上一章

2.6 第二类Chebyshev多项式(选讲)

$$T_k'(x) = \frac{\sin(k \arccos x) * k}{\sqrt{1-x^2}} = \frac{k \sin k\theta}{\sin\theta}$$

第二类Chebyshev多项式

$$U_{k-1}(x) = \frac{\sin k\theta}{\sin\theta} = \frac{\sin(k \arccos x)}{\sqrt{1-x^2}}$$

有$(k-1)$个不同零点

$$x_j = \frac{\cos j\pi}{k},\ 1 \le j \le k-1$$

也是$T_k(x)$的极值点。

由和差化积公式

$$U_{k+1}(x) + U_{k-1}(x) = \frac{\sin(k+2)\theta + \sin k\theta}{\sin\theta} = \frac{2\sin(k+1)\theta\cos\theta}{\sin\theta} = 2xU_k(x)$$

得递推式

$$\begin{cases} U_{k+1}(x) = 2xU_k(x) - U_{k-1}(x) \\ U_0 = 1, \quad U_1 = 2x \end{cases}$$

由正弦函数的正交性可得U的正交性

$$\int_0^\pi \sin mx \sin kx dx = \int_{-1}^1 U_{m-1}(t)U_{k-1}(t)\sqrt{1-t^2}dt = 0$$

例 8 记$x_j = \cos\frac{j\pi}{N}$, 证明：

$$\left|\prod_{j=0}^N (x - x_j)\right| \le 2^{1-N}$$

留作研究生作业。 并给出以Chebyshev-Lobatto点（一般都称为Chebyshev点，除非有意区分第一类或者第二类） 进行多项式插值，所得插值余项估计式。

标注 6 Chebyshev点插值可作为近似最佳一致逼近，但不是理论上的最佳一致逼近:

$$\|q^* - f\|_\infty := \min_{p\in H_n} \max_{x\in[a,b]} |f(x) - p(x)|$$

记$p^*(x)$是f 的 n次Cheybyshev零点插值，$\tilde{p}(x)$是Chebyshev级数的前n项和 [7]； 则成立

$$\|p^* - f\|_\infty \approx \|\tilde{p} - f\|_\infty \le \left(4 + \frac{4}{\pi^2}\log(n)\right)\|q^* - f\|_\infty$$

该估计式表明Chebyshev零点插值和最佳一致逼近的误差只相差有限常数倍 [8]，但是计算minmax是相对困难的. 一般计算连续函数的最佳一致逼近采用Remez(1934年)算法，Trefethen的程序包chebfun里的 minimax函数用的即是Remez算法[9]. 上式的误差估计式显示 Chebyshev点插值与最佳一致逼近的误差为5到20倍的关系，但这是估计 并非真实的误差倍数关系；以Runge函数为例，Chebyshev零点插值 与最佳一致逼近多项式的 最大模误差仅差两倍左右；另外， Chebyshev零点插值的2范数往往更小. 这也是大家往往不愿意花更多代价去计算最佳一致逼近的原因吧! 同样，离散数据的最佳一致逼近多项式的计算同样是困难的. 我们在后文给出计算该问题的Lawson迭代算法.

[7] 见4.2节; Chebyshev级数有限和p^* 通常作为近似最佳(near-best)一致逼近； 若忽略计算系数时产生的误差，Chebyshev级数与零点插值在数值计算上可视为相等.

[8] 数值计算时可视对数$\log(n)$为常数

[9] Remez算法是迭代法，较为复杂，参考代码： BestPolyInf.m，fzeronewton.m

3 连续函数最佳平方逼近

定义 7 对$f(x)\in \boldsymbol{C}[a,b]$, 以及$\boldsymbol{C}[a,b]$的一个子集（有限维子空间、也是线性空间）

$$\Phi=\text{span}\{\varphi_0(x),\varphi_1(x),\cdots,\varphi_n(x)\}$$

若 $s^\star\in\Phi$ 满足

$$\|f(x)-s^\star(x)\|_2=\min_{s\in\Phi}\|f(x)-s(x)\|_2$$

则称$s^\star(x)$ 是 $f(x)$ 在 Φ 中的<u>最佳平方逼近函数.</u>

若定义带权内积

$$(f,g)=\int_a^b\rho(x)f(x)g(x)dx,\quad \|f\|^2=(f,f)$$

则

$$\|f-s\|_2^2=(f-s,f-s)=\int_a^b\rho(x)[f(x)-s(x)]^2dx$$

记 $s(x)=\sum_{j=0}^{n}a_j\varphi_j(x)$, 最佳平方逼近问题等价于多元函数

$$I(a_0,a_1,\cdots,a_n)=\int_a^b\rho(x)\left[f(x)-\sum_{j=0}^{n}a_j\varphi_j(x)\right]^2dx$$

求极小值.

由Fermat 引理，得到极小值必要条件

$$\frac{\partial I}{\partial a_k}=2\int_a^b\rho(x)\left[f(x)-\sum_{j=0}^{n}a_j\varphi_j(x)\right]\varphi_k(x)dx=0,\quad 0\le k\le n$$

也即是

$$\sum_{j=0}^{n}\Big(\varphi_k,\varphi_j\Big)_\rho a_j=\Big(f,\varphi_k\Big)_\rho,\quad k=0,1,\cdots,n \tag{3}$$

注意：上述线性方程组称为法方程，系数矩阵为Gram矩阵.

例 9 (驻点存在唯一性) 若φ_j线性无关，则上述线性方程组解存在唯一.

Hint: 对任意指标k

$$\sum_{j=0}^{n}\Big(\varphi_k,\varphi_j\Big)a_j=0\Rightarrow\left(\sum_j a_j\varphi_j,\varphi_k\right)=0$$

$$\Rightarrow \left(\sum_j a_j\varphi_j, a_k\varphi_k\right) = 0 \Rightarrow \left(\sum_j a_j\varphi_j, \sum_k a_k\varphi_k\right) = 0$$

进而：$\sum_{j=0}^{n} a_j\varphi_j = 0$ 又由φ线性无关得到 $a_j = 0$. ■

例 10 (检验驻点为最小值点) 设$s^\star(x) = \sum_{j=0}^{n} a_j^\star\varphi_j(x)$是上述方程组的解. 证明

$$\|f - s^\star\| \le \|f - s\|, \qquad \forall s(x) \in \varphi$$

也即

$$\int_a^b \rho(x)\Big(|f - s^\star|^2 - |f - s|^2\Big)dx \le 0$$

Hint: 由平方差公式

$$|f - s^\star|^2 - |f - s|^2 = (2f - s^* - s)(s - s^*)$$

整理后得

$$\int_a^b \rho(x)\Big((s - s^\star)(2f - 2s^\star + s^\star - s)\Big)dx = -\|s - s^\star\|^2 \le 0$$

其中利用了等式 $\dfrac{\partial I}{\partial a_j} = (f - s^\star, \varphi_j) \equiv 0$, 得证. ■

Theorem 10 最佳平方逼近误差满足如下估计式

$$\|f - s^\star\|^2 = \Big(f - s^\star, f - s^\star\Big) = \Big(f, f - s^\star\Big) = \|f\|^2 - \Big(f,\ s^\star\Big)$$

由以上分析我们给出计算给定函数$f(x)$的最佳平方逼近多项式的算法过程

1) 取定基函数φ_j (例如 $\{\varphi_j = x^j,\ 0 \le j \le n\}$) 确认权函数(例如：$\rho(x) = 1$)

2) 根据(3)式，计算“法方程”的系数矩阵(Gram矩阵) A 和右端项 $\boldsymbol{r}$

$$A_{ij} = (\varphi_i, \varphi_j), \qquad \boldsymbol{r}_i = (\varphi_i, f), \qquad 0 \le i, j \le n$$

3) 求解方程$A\boldsymbol{a} = \boldsymbol{r}$得到系数$\boldsymbol{a}$ 及最佳平方逼近多项式 $s^*(x) = \sum_{i=0}^{n} a_i\varphi_i(x)$

例 11 (编程计算) 设 $f(x) = \sqrt{1 + x^2}$, 取基函数$\varphi_j(x) = x^j$. 计算[0,1]上的$n = 1, 2, 5, 10, 15, 20$次最佳平方逼近多项式$p_n(x)$. 输出函数$f(x)$和$p_n(x)$的图像，输出节点取$[0, 1]$区间上200个等分点.

计算可得

$$A_{ij} = (\phi_i, \phi_j) = \int_0^1 x^{i+j}dx = \frac{1}{i + j + 1}, \qquad \boldsymbol{r}_j = (f, x^j) = \int_0^1 x^j\sqrt{1 + x^2}dx$$

```
f = @(z) sqrt(1+z.^2);
% best polynomial approximation in L2 norm / by monomial basis
n = 12;   % test n = 2, 5, 10, 15, 20
j1 = 0:n; k1 = j1;       A = 1./(j1.' + k1 + 1);
r = zeros(n+1,1);      % rhs    (phi_j, f)
for j = 1:n+1
    r(j) = integral(@(x) f(x).*x.^(j-1), 0,1);
end
a = A \ r;             % coef of p_n; p_n = a0 + a1*x + a2*x^2 + ...
s = linspace(0,1,201)';  V = s.^(0:n);  p = V*a;
err_inf = norm(p - f(s),inf)
figure; plot(s,p,'r-',s,f(s),'b-','LineWidth',1);
legend('best␣poly␣L2␣approx␣', 'exact␣curve'); grid on,
```

其中法方程对应的系数矩阵A 为Hilbert矩阵, 形式如下：

$$H_5 = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 & 1/5 \\ 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 \\ 1/4 & 1/5 & 1/6 & 1/7 & 1/8 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \end{bmatrix}$$

当多项式次数 n 较大时，Hilbert矩阵是坏条件数的(ill-condition)，$\mathrm{cond}(H_{11}) = 5\times 10^{14}$, 这将导致严重的数值不稳定和计算误差。 当$n$较大时，应使用正交多项式：Chebyshev, Legendre多项式作为基函数。

3.1 以正交函数族为基的最佳平方逼近

在逼近函数待选取空间 Φ 中选择一组正交函数为基(仍记作ϕ)，即

$$\Phi = \mathrm{span}\{\phi_0, \phi_1, ..., \phi_n\}, \qquad \& \qquad \big(\phi_j, \phi_k\big) = 0,\ j \neq k$$

则目标函数

$$s(x) = \sum_{j=0}^{n} \alpha_j \phi_j$$

的系数满足法方程

$$\sum_{j=0}^{n} \Big(\phi_k, \phi_j\Big)\alpha_j = \Big(f, \phi_k\Big), \quad k = 0, 1, \cdots, n.$$

根据正交性得 $\alpha_j = \dfrac{(f,\phi_j)}{(\phi_j,\phi_j)}$. 则$f(x)$ 在线性空间Φ中的最佳逼近函数

$$s(x) = \sum_{j=0}^{n} \frac{(f,\phi_j)}{(\phi_j,\phi_j)}\phi_j(x)$$

3.2 最佳平方逼近多项式——Legender 级数

下面给出最佳平方逼近多项式的计算方法.

> 对给定的函数$f(x) \in C_{[-1,1]}$，可选取Legendre多项式$\{P_j(x)\}_{j=0}^n$作为基函数，利用正交性，计算 $f(x)$的最佳平方逼近n次多项式$Q_n(x)$
>
> $$Q_n(x) = \sum_{j=0}^n \alpha_j P_j(x) = \sum_{j=0}^n \frac{(f,P_j)}{(P_j,P_j)} P_j(x) = \sum_{j=0}^n \frac{(f,P_j)}{2/(2j+1)} P_j(x)$$
>
> 上式需要求出系数α_j或者$(f,P_j) = \int_{-1}^1 f(x)P_j(x)dx$. 当$n \to \infty$时，称为Legendre级数.

当$f \in C^1_{[-1,1]}$时，Legendre级数收敛

取n阶Guass-Legendre数值积分公式[10]

$$\int_{-1}^1 g(x)dx \approx \sum_{k=0}^n w_k g(x_k), \qquad x_k \text{ is Gauss points}$$

则此时计算所得（非精确的）最佳平方逼近多项式为

$$\tilde{Q}_n(x) = \sum_{j=0}^n \alpha_j P_j(x) = \sum_{j=0}^n \frac{\sum_{k=0}^n w_k f(x_k)P_j(x_k)}{2/(2j+1)} P_j(x)$$

记$Q^*(x) = \sum_{k=0}^n c_k P_k(x)$ 是以Gauss点为插值节点的n次插值多项式. 则

$$\tilde{Q}_n(x) = \sum_{j=0}^n \frac{\sum_{k=0}^n w_k Q^*(x_k)P_j(x_k)}{2/(2j+1)} P_j(x) = \sum_{j=0}^n \frac{(Q^*,P_j)}{2/(2j+1)} P_j(x) = Q^*(x)$$

上式第二个等号成立，利用了Gauss积分的代数精度是$(2n+1)$; 最后一个等号成立利用了$Q^*(x)$是n次多项式且$P_j(x)$ 是一组正交基, 于是 $Q^*(x)$在H_n中以$P_j(x)$为基函数的展开是其自身.

标注 7 以上分析表明, 若选择Legendre多项式为基函数，且计算系数采用n阶Gauss积分公式时，(非精确的)最佳平方逼近多项式等价于Gauss点插值. 记 $Q_n(x)$为理论上的最佳平方逼近多项式. 我们重新计算上式，并比较“精确的”最佳逼近多项式$Q_n(x)$ 与“实用的”$Q^*(x)$之间的差别 —— 数值上几乎没差别.

```
function [p1, c] = Legend_Series(f,n,t,a,b)
% input: f is given function
%        n is degree of best L2 approximation polynomial Q
%        t is the pts be evaluate
% output:  p1 = Q(t);  c is coeffs of Q in Legendre basis
% for example:
%     f = @(z) sqrt(1+z.^2);   a = 0.1; b = 1;
%     t = linspace(a,b,801)';  n = 22;
%     p = Legend_Series(f,n,t,a,b);  norm(p - f(t),inf)
N = max(32,2*n);      [Xp,Wp] = legpts(N);
```

[10] Gass公式对所有次数不超过$(2n+1)$的多项式精确成立，见第四章

```
if nargin>3
    X0 = Xp*(b-a)/2 + (b+a)/2;      t =  (2*t - (b+a))/(b-a);
end
% compute ck = (f,Pk) via Gauss-Legendre rule of (N) degree,
P = get_legend(Xp,n);        mj = f(X0).*Wp(:);      c = P'*mj;
c = c.*((0:n)'+0.5);   Pt = get_legend(t,n);        p1 = Pt*c;
```

代码用到了Gauss-Legendre 数值积分公式：legpts[11]

3.3 Chebyshev级数和Chebyshev点插值

对正交多项式序列ϕ_j, 当$n \to \infty$, 和给定函数 $f \in C_{[a,b]}$, 若

$$f(x) = \sum_{j=0}^{\infty} \alpha_j \phi_j, \quad \alpha_j = \frac{(f, \phi_j)}{(\phi_j, \phi_j)}$$

上式称之为广义Fourier级数. Legendre级数， Fourier级数即是特例。 当选取级数得有限项截断时， 所得有限项和即是最佳平方逼近（或三角）多项式.

如果$f(x) \in C_{[-1,1]}$, Chebyshev级数按下式展开

$$f(x) = \frac{C_0}{2} + \sum_{k=1}^{\infty} C_k T_k(x) \quad 其中 \quad C_k = \frac{2}{\pi} \int_{-1}^{1} \frac{f(x) T_k(x)}{\sqrt{1-x^2}} dx$$

由于级数的展开基函数是正交多项式， 任意的有限n 项截断对应 n 次最佳平方逼近多项式(以带权内积定义的范数)。 注意到连续的最佳平方逼近理论很完美，但是具体实现时涉及到计算连续内积(积分)。 读者在学习完数值积分后，可以进一步补充完善最佳平方逼近的算法实现。 一般采用 Chebyshev-Guass 数值积分公式计算系数C_k.

标注 8 我们给出带权最佳逼近多项式和Chebyshev点插值之间的关系. 由带权正交性可知, Chebyshev级数的前n项和即是$f(x)$的带权最佳平方逼近多项式

$$Q(x) = \sum_{k=0}^{n}{}' C_k T_k(x), \qquad C_k = \frac{2}{\pi} \int_{-1}^{1} \frac{f(x) T_k(x)}{\sqrt{1-x^2}} dx$$

上式中求和号上标表示首项权重取一半. 记$Q^*(x)$为以Chebyshev零点$\{x_j = \cos(\frac{(j+1/2)\pi}{n+1})\}_{j=0}^{n}$为插值节点的$n$次插值多项式. 记(由Chebyshev-Gauss 公式计算C_k [12])

$$\tilde{Q}(x) = \sum_{k=0}^{n}{}' \tilde{C}_k T_k(x), \quad \tilde{C}_k = \frac{2}{n+1} \sum_{j=0}^{n} f(x_j) T_k(x_j) = \frac{2}{n+1} \sum_{j=0}^{n} f(x_j) \cos(\frac{(j+1/2)k\pi}{n+1}) \approx C_k$$

则$Q^*(x) = \tilde{Q}(x)$, 即“实用的”带Chebyshev权的最佳平方逼近多项式等于 Chebyshev零点插值.

另外Chebyshev极值点(包含端点)也称为(Chebyshev-Lobatto点), $\{x_j = \cos(\frac{j\pi}{n})\}_{j=0}^{n}$ 更为常用, 此时

$$\hat{Q}(x) = \sum_{k=0}^{n}{}'' \hat{C}_k T_k(x), \quad \hat{C}_k = \frac{2}{n} \sum_{j=0}^{n}{}'' f(x_j) T_k(x_j) = \frac{2}{n} \sum_{j=0}^{n}{}'' f(x_j) \cos(\frac{jk\pi}{n}) \approx C_k$$

[11] 代码见 Chebfun工具包，建议所有同学下载使用.

[12] 由积分换元法$C_k = \frac{2}{\pi} \int_0^{\pi} f(\cos\theta) \cos(k\theta) d\theta$, Chebyshev-Gauss公式等价于对该周期函数使用 梯形或者中点公式积分.

上式求和上标表示首尾两项求和系数减半. 可以检验[13] $\hat{Q}(x)$等价于$f(x)$的Chebyshev-Lobatto点插值. 注意到 与$\hat{Q}(x)$ 的尾项不同，直观上，$\tilde{Q}(x)$是带权的最佳平方逼近n 次多项式， 而$\hat{Q}(x)$ 的第n 项系数减半，仅可视为$(n-1)$ 次带权最佳平方逼近. 但是由于光滑函数的Chebyshev系数衰减很快，实际计算效果几乎没有区别. 另外， 上述两种方法计算系数C_k的近似值都有显式表达式，一次余弦变换 （相应的余弦矩阵乘以向量）即可完成，计算量为$\boldsymbol{O}(n^2)$.

```
f = @(x) exp(x).*sin(5*x);
n = 7;        j0 = (0:n)';
% % cheb pts 2nd kind (lobatto)
xp = cos(j0*pi/n);  C = cos((j0*j0')*(pi/n));
f1 = f(xp);     f1([1,end]) = 0.5*f1([1,end]);
c1 = C*f1;      % cosine matrix multiple vector
c1 = c1*(2/n); c1([1,end]) = 0.5*c1([1,end]);
% %  cheby 1st pts (zeros)
xp2 = cos((j0+0.5)*pi/(n+1)); C = cos( (j0*(j0+0.5)')*(pi/(n+1)) );
c2 = C*f(xp2);  % cosine matrix multiple vector
c2 = c2*(2/(n+1)) ;  c2(1) = 0.5*c2(1);
% % -- output
t = linspace(-1,1,201)';    V = get_cheb(t,n);
p1 = V*c1;     p2 = V*c2;
err1 = norm(p1 - f(t),inf), err2 = norm(p2 - f(t),inf)
figure; plot(t,p1-f(t),'r-',t,p2-f(t),'b-','LineWidth',1); grid on,
legend('Cheb␣pts␣interp␣2nd␣kind','Cheb␣pts␣interp␣1st␣kind');
```

测试以上代码可发现：n次 Chebyshev零点插值几乎具有$(n+2)$ 个交错点，这符合 最佳一致逼近的理论. 若计算C_k 时采用更高精度的数值积分公式——$N(N \gg n)$点 Gauss-Chebyshev公式，则可得到精确的Chebyshev级数$Q(x)$. 读者可学习完数值积分后补充完善. 参考代码Cheby_Series.m

余弦矩阵与傅立叶矩阵(选讲)

Chebyshev点插值在实际计算中的广泛使用还得益于快速余弦变换(dct) (以FFT实现). 上节已经分析，采用Chebyshev点插值或者计算Chebyshev级数的前n项系数主要计算量 来自于一次余弦变换. 借助于FFT实现余弦变换，计算量可降低为$n\log n$. 我们以上节 Chebyshev-Lobatto点插值

$$\sum_{k=0}^{n} a_k T_k(x_j) = f(x_j), \quad j = 0, 1, ..., n$$

为例，给出$\boldsymbol{a}$和$\boldsymbol{f}$之间的关系. 注意上式是线性方程组 $\boldsymbol{T}\boldsymbol{a} = \boldsymbol{f}$. 从$\boldsymbol{a}$到$\boldsymbol{f}$ 就是矩阵$\boldsymbol{T}$ 对应的线性变换，由于此时$\boldsymbol{T}$是余弦矩阵，即 $\boldsymbol{T} = (C_{jk}) = \left(\cos\left(\frac{jk\pi}{n}\right)\right)$. 余弦矩阵是几乎正交矩阵，且矩阵乘法和矩阵求逆都可 快速实现。 由上节标注分析 $a_k = \hat{C}_k$， 可直接由$\boldsymbol{T}$乘以向量实现。 我们下面

[13]利用Chebyshev或者余弦函数的 离散正交性

给出$\boldsymbol{a} = \boldsymbol{T}^{-1}\boldsymbol{f}$ 更为高效的计算过程. 首先将$(n+1)$维向量扩充至$2n$维度， 记中间变量

$$u_j = \begin{cases} f_j, & 1 \le j \le n-1 \\ \frac{1}{2} f_j & j = 0,\ n \\ 0, & n+1 \le j \le 2n \end{cases}$$

则由$\hat{C}_k$ 的表达式得到

$$\frac{N}{2}\hat{C}_k = \sum_{j=0}^{n}{}'' f_j \cos(\frac{jk\pi}{n}) = \sum_{j=0}^{2n} u_j \cos(jk\frac{2\pi}{2n}) = \text{Real}\left(\sum_{j=0}^{2n} u_j e^{ijk\frac{2\pi}{2n}}\right), \qquad k = 0, 1, ..., n$$

则n维向量的余弦变换可通过$2n$维的Fourier变换实现. 当$n > 1000$, 利用FFT 实现 DCT比matlab矩阵乘法更高效. 读者可 选择不同的n测试以下代码.

```
f = @(x) exp(x).*sin(66*x);
n = 1024; j0 = (0:n)';  xp = cos(j0*pi/n);  f1 = f(xp);
C = cos((j0*j0')*(pi/n));
tic , for j = 1:1000, u1 = C*f1;    end,     toc,
tic , for j = 1:1000, u2 = mydct0(f1); end, toc,
err1 = norm(u1-u2,inf)
% %
function y = mydct0(x)
% C = cos(jk*pi/(m-1)), j,k = 0 -> m-1   % y = C*x
[m,n] = size(x);  y = zeros(2*(m-1),n);  y(1:m,:) = x(1:m,:);
y = fft(y);        y = real(y(1:m,:));
end
```

标注 9 Chebyshev级数是带权的最佳平方逼近多项式，通常也作为最佳一致逼近(minimax)的替代品. 实际应用时，多选择Chebyshev点插值来计算Chebyshev级数(二者略有不同)； 更多选择Chebyshev-Lobatto点$x_j = \cos(j\pi/n)$插值， 此时对应的系数矩阵是对称的, 计算也更为简单. 下列代码给出了基于余弦逆变换的Chebyshev点插值， 计算效率和稳定性对比前文的Lagrange、Barycentric插值形式均有大幅提高. 甚至于对于上万个节点的插值都是 数值稳定的。

```
function [px,c] = ChebyInterp(f,n,a,b,xout)
% for example :
%      f = @(x) exp(x).*sin(11*x);
%      a = 0.1; b = 1; xout = linspace(a,b,801)';    n = 2222;
%      [px,c] = ChebyInterp(f,n,a,b,xout); norm(px - f(xout),inf)
t = cos((0:n)'*pi/n);  X = t*(b-a)/2 + (b+a)/2;
fX = f(X);       c = imydct0(fX) ;
xt = ( 2*xout - (b+a) )/ (b-a);    V = get_cheb(xt,n);   px = (V*c);
end
% ===
function y = imydct0(x)  % ==>    inverse of mydct0
```

```
% C = cos(jk*pi/(m-1)), j,k = 0 -> m-1 //  y = C \ x
m = size(x,1);  x([1,m],:) = 0.5*x([1,m], :);
y = mydct0(x);  y([1,m],:) = 0.5*y([1,m], :);  y = y*(2/(m-1));
end
```

4 离散的最佳平方逼近 （最小二乘拟合）

问题： 已知离散点处的函数值

$$f(x_j) = y_j, \quad j = 0, 1, ..., m$$

目标： 给定$(n+1)$维线性空间 $(m > n)$

$$\Phi = \mathrm{span}\{\varphi_0(x), \varphi_1(x), \cdots, \varphi_n(x)\}$$

计算

$$s^\star(x) = \underset{\forall s\in\Phi}{\mathrm{argmin}} \|s(\boldsymbol{x}) - \boldsymbol{f}\|_2$$

也即寻找函数$s^\star(x) = \sum_{l=0}^{n} a_l\varphi_l(x)$, 使得

$$\sum_{j=0}^{m} |s^\star(x_j) - y_j|^2 = \min_{s\in\Phi} \sum_{j=0}^{m} |s(x_j) - y_j|^2$$

引入权函数$w(x)$, 更一般的问题提法为求

$$\sum_{j=0}^{m} w_j|s^\star(x_j) - y_j|^2 = \min_{s\in\Phi} \sum_{j=0}^{m} w_j|s(x_j) - y_j|^2, \qquad w_j = w(x_j)$$

相应带权内积为

$$\Big(\varphi_l, \varphi_k\Big)_w = \sum_{j=0}^{m} w_j\varphi_l(x_j)\varphi_k(x_j)$$

不引起歧义时，略去下标w。

最小二乘即是离散的最佳平方逼近. 类似于连续问题的推导，上述问题归结为多元函数求极值问题

$$I(a_0, ..., a_n) = \sum_{j=0}^{m} \underline{w_j} \left|\sum_{l=0}^{n} a_l\varphi_l(x_j) - y_j\right|^2$$

$$\frac{\partial I}{\partial a_k} = 2\sum_{j=0}^{m} \underline{w_j} \left(\sum_{l=0}^{n} a_l\varphi_l(x_j) - y_j\right)\varphi_k(x_j) = 0$$

$$\boxed{\sum_{l=0}^{n} a_l\Big(\varphi_l, \varphi_k\Big) = \Big(y, \varphi_k\Big), \quad k = 0, 1, ..., n}$$

法方程可解性

离散的最佳平方逼近与连续函数的最佳平方逼近所得到的法方程 形式上一样。若系数矩阵(Gram矩阵) G可逆，则法方程唯一可解。 对于连续的最佳平方逼近，基函数$\{\varphi_k(x)\}$彼此线性无关，则G可逆。 对于离散数据，基函数$\{\varphi_k(x)\}$在连续意义下线性无关，也可能在节点x_j处取值得到的 向量$\{\varphi_k(x_j)\}$ 线性相关，此时G不可逆。

例 12 函数$\{\sin x, \sin 2x\}$ 在$[0, 2\pi]$上是连续意义下正交的(线性无关)，但是在节点 $\{0, \pi, 2\pi\}$, 取值得到的矩阵是全零矩阵，线性相关。 相应的Gram矩阵也是零矩阵，不可逆。

离散问题的法方程解的存在唯一性不但要求所选基函数线性无关，还需要额外的条件，例如Haar条件。

定义 8 **(Haar条件)** 设$\{\phi_k\}_{k=0}^n$的任意线性组合在点集合$\{x_i, i = 0, 1, ..., m\}$上至多只有$n$个不同的零点。

例如: 取$\phi_k = x^k$, $m > n$时，矩阵

$$\begin{bmatrix} 1 & \boldsymbol{x} & \dots & \boldsymbol{x}^n \end{bmatrix}_{(m+1)\times(n+1)}$$

是列满秩矩阵，同时也 满足Haar条件。 我们有一下结论。

Theorem 11 **(离散最佳平方逼近函数)** 对于离散数据$\{(x_j, y_j)\}_{j=0}^m$, $\Phi = \text{span}\{\varphi_0, ..., \varphi_n\}$, 若$\varphi_k$满足Haar条件，则其离散的最佳平方逼近函数存在唯一.

推论 1 **(离散最佳平方逼近多项式)** 在定理条件下，若$m > n$，则离散的最佳平方逼近多项式存在唯一.

标注 10 Haar条件是离散最佳平方逼近函数的充分非必要条件！ 例如, 取

$$\Phi = \text{span}\{\sin x, \sin 3x\}, \quad x_0 = 0, x_1 = \frac{\pi}{3}, x_2 = \frac{2\pi}{3}, x_3 = 3$$

得Gram矩阵

$$G = V^\top V, \quad V = \begin{bmatrix} 0 & 0 \\ \frac{\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & 0 \\ \sin 3 & \sin 9 \end{bmatrix}$$

此时G对称正定，可逆，离散的最佳平方逼近解存在唯一。另一方面，Φ不满足Haar条件。

例 13 **(参考李庆扬教材 p75例9)** 根据表中数据给出线性拟合函数$s(x) = a_0 + a_1 x$

x_i	1	2	3	4	5
f_j	4	4.5	6	8	8.5
w_j	2	1	3	1	1

```
1  Xnodes = [1 2 3 4 5];     % sample pts
2  fX = [4 4.5 6 8 8.5];     % sample values
3  W = [2 1 3 1 1];          % weight vector
4  m = length(Xnodes);    p0 = ones(m,1);    p1 = Xnodes(:);
5  V = [p0, p1];             % vandermonde matrix
6  n = 1;  H = zeros(n+1);  r = zeros(n+1,1); % normal matrix and RHS
7  for j = 1:n+1
8      for k = 1:n+1
9          H(j,k) = W*(V(:,j).* V(:,k));
10     end
11     r(j) =  W*(V(:,j).*fX(:));
12 end
13 c = H \ r,              % p = c0 + c1*x
14 % dW = diag(W); c = (V'*dW*V) \ (V'*dW*fX(:)),    % line 15 = line 16
15 % dW = diag(sqrt(W)); c = (dW*V)\(dW*fX(:)),      % = lines 7-14
16 figure; LW = 'linewidth'; MK = 'markersize';
17 plot(Xnodes,fX,'ro',MK,8); hold on;
18 t = linspace(1,5,201)';   p = c(1) + c(2)*t;
19 plot(t,p,'k-',LW,1);  legend('sample␣data','approx␣line')
```

4.1 最小二乘的直接法

- 根据逼近函数空间基函数写出函数表达式

$$s(x) = \sum_{l=0}^{n} a_l \varphi_l(x)$$

- 根据$m+1$组离散数据 $s(x_j) = y_j$写出对应线性方程组 $Va = \boldsymbol{f}$

$$a_0\varphi_0(x_j) + a_1\varphi_1(x_j) + ... + a_n\varphi_n(x_j) = y_j,\ 0 \le j \le m$$

- 根据权函数改写等价方程组

$$w_j\Big[a_0\varphi_0(x_j) + a_1\varphi_1(x_j) + ... + a_n\varphi_n(x_j)\Big] = w_j y_j,\ 0 \le j \le m$$

- 上述系数矩阵A是$(m+1)\times(n+1)$阶的；一般情况下$(m > n)$ 记权函数 矩阵$W = \mathrm{diag}(w_0, w_1, \cdots, w_m)$, 最小二乘拟合等价于

$$\underline{\underline{V^{\mathrm{T}} * W * \underline{V\boldsymbol{a}}}} = V^{\mathrm{T}} * W * \boldsymbol{f}$$

例 14 上述线性方程组等价于离散的最佳平方逼近. (证明留作作业)

读者可测试上一代码注释部分. 对比可知：本节的算法过程更加简单易行!

几何解释：正交投影

例 15 (投影矩阵) 设$x, y \in \mathbb{R}^n$, x在y上的投影向量

$$P_{roj}x = y\alpha = y\frac{y^\top x}{y^\top y} = y(y^\top y)^{-1}y^\top x$$

记$A = [y_1, y_2] \in \mathbb{R}^{n\times 2}$, 则$x$在子空间span$(A)$ 中的投影为

$$P_{roj}x = A(A^\top A)^{-1}A^\top x$$

计算

$$\underset{\forall s\in\Phi}{\text{argmin}}\, \|s(\boldsymbol{x}) - \boldsymbol{f}\|_2 = \underset{\forall s\in\Phi}{\text{argmin}} \sum_{j=0}^{m} w_j |s(x_j) - f(x_j)|^2$$

也即寻找函数$s(x) = \sum_{l=0}^{n} a_l\varphi_l(x)$, 使得

$$(\boldsymbol{f} - s) \perp^w span(\Phi) \Leftrightarrow \big(\boldsymbol{f} - s,\ \varphi_k\big)_w = \Big(\boldsymbol{f} - \sum_{l=0}^{n} a_l\varphi_l(x),\ \varphi_k\Big)_w = 0, \quad k = 0, ..., n.$$

上式即是法方程$G\boldsymbol{a} = V^\top W\boldsymbol{f}$, 其中

$$G = V^\top WV, \quad V = [\varphi_0(\boldsymbol{x}), ..., \varphi_n(\boldsymbol{x})]$$

直接算法

求解法方程有两个缺点

1. 计算量大; 计算G的代价是$(mn)^2$, $V^\top W\boldsymbol{f}$的代价是nm
2. G的条件数可能很大($cond(V)$的平方)，例如$\phi_j = x^j$

需要其它算法来计算该问题。

记span$(\sqrt{W}V)$ 的一组标准正交向量组 横向串联的矩阵为$Q \in \mathbb{R}^{(m+1)\times(n+1)}$,

$$(\boldsymbol{f} - s) \perp^w span(\Phi) \Leftrightarrow \sqrt{W}(\boldsymbol{f} - s) \perp span(\sqrt{W}\Phi) \Leftrightarrow \sqrt{W}(\boldsymbol{f} - s) \perp Q(:, k),\ k = 0, ..., n$$

其中$\perp^w$表示带权正交，$\perp$表示欧式内积下正交。 矩阵形式

$$Q^\top\sqrt{W}(\boldsymbol{f} - V\boldsymbol{a}) = 0 \Leftrightarrow Q^\top\sqrt{W}V\boldsymbol{a} = Q^\top\sqrt{W}\boldsymbol{f}$$

利用矩阵的QR分解得$\sqrt{W}V = QR$, $R \in \mathbb{R}^{(n+1)\times(n+1)}$为上三角矩阵[14]， 得到

$$Q^\top QRa = Ra = Q^\top\sqrt{W}\boldsymbol{f}.$$

两点说明

1. 计算量减小：QR分解计算代价$\frac{1}{2}n^2m$; 求解三角形线性方程组的代价$\frac{1}{2}n^2$.
2. 条件数变小：$cond(R) = cond(V) = \sqrt{cond(G)}$.

[14]矩阵的QR分解见下节。

4.2 超定线性方程组求解和矩阵的QR分解

例 16 (**Gram-Schmidt正交化**) 给定m个线性无关的向量$a_k \in \mathbb{R}^n$, $k = 1, ..., m$ 且 $m < n$, 则由Gram- Schmidt正交化方法可得m个标准正交向量组 $q_k \in \mathbb{R}^n, k = 1, ..., m$. 公式如下

$$\begin{cases} q_1 = a_1 \\ q_{k+1} = a_{k+1} - \displaystyle\sum_{j=1}^{k} \frac{a_{k+1}^\top q_j}{q_j^\top q_j} q_j \end{cases}$$

将上式改写为

$$a_{k+1} = c_{1k}q_1 + ... + c_{kk}q_k + q_{k+1}$$

写为矩阵形式

$$[a_1, a_2, ..., a_m] = [q_1, q_2, ..., q_m] \begin{bmatrix} 1 & c_{11} & c_{12} & * \\ 0 & 1 & c_{22} & * \\ & & \ddots & \\ 0 & 0 & \cdots & 1 \end{bmatrix}_{m\times m} = [q_1, q_2, ..., q_m] * C$$

记$A = [a_1, ..., a_m]$, $Q = [\tilde{q}_1, ..., \tilde{q}_m]$, 其中$\tilde{q}_k = q_k/\|q_k\|$. 则$Q^\top Q = I_m$, 且

$$[q_1, q_2, ..., q_m] = [\tilde{q}_1, ..., \tilde{q}_m] \begin{bmatrix} \|q_1\| & & & \\ & \|q_2\| & & \\ & & \ddots & \\ & & & \|q_m\| \end{bmatrix} := Q \cdot D$$

得到$A = QR$, 其中$R = C * D$为上三角矩阵. A 分解为正交矩阵和上三角矩阵的乘积成为QR分解，当R的对角线符号指定时(这里符号全部取正)，QR分解是唯一的。

1. A, Q有相同的列空间(值域)，$span(A) = span(Q)$

2. 最小二乘问题，也即是求解超定方程$Ax = b$ 等价于

$$(b - Ax) \perp span(Q) \Leftrightarrow Q^\top(b - Ax) = 0 \Leftrightarrow Q^\top(QR)x = Q^\top b \Leftrightarrow Rx = Q^\top b$$

4.3 离散的最佳平方逼近多项式 (正交多项式实现最小二乘拟合)

在实际应用中，一方面数据样本大多是离散形式，另一方面多项式总是最受欢迎的。 因此离散的最佳平方逼近多项式的应用很广泛。前文提到，选择单项式作为基函数所得到的Hilbert矩阵或者 Vandermonde矩阵是 病态的。基函数的最佳选择是Chebyshev多项式。我们给出基于Chebyshev多项式的离散最佳平方逼近多项式的算法。

A-1) 转化自变量区间到标准区间$[-1, 1]$

A-1) 选择Chebyshev多项式为基函数

A-1) 计算超定方程的最小二乘解得到Chebyshev系数

A-1) 输出需要的节点处的近似值

以Runge函数为例, 我们给出测试代码， 读者可以测试取$n = 20, 40, 80, 200$时的误差变化.

```
    f = @(z) 1./(1+25*z.^2);  %   test func
%     sample pts
    N = 801;    X = linspace(-1,1,N)';
    n  = 11;    V = get_cheb(X,n);    c = V \ f(X);    cond(V),
%     output pts   and func values
    t = linspace(-1,1,N*2)';    Vt = get_cheb(t,n);    y1 = Vt*c;
    err_cheb = norm(y1-f(t),inf),   LW = 'linewidth';
    figure; plot(t,f(t),'r-', t,y1,'b-.',LW,1);
```

标注 11 测试发现，当$n > 200$ 时，即是基函数选择为Chebyshev 多项式，矩阵V 的条件数仍然很大. 若采用上述方法，不能得到高精度的解，采用Vandermonde via Arnoldi 可部分解决(Trefethen, SIREV2021)； 采用Chebyshev Vandermonde via Arnoldi 可进一步改善; 但是并没有完全解决问题的病态性造成的误差 污染. 教材p76第3.4.2节介绍的利用离散数据构造正交多项式的方法 [15]，当n较大时所得首一的正交多项式矩阵条件数极差，不如本节的算法精度高，且 不如本节方法计算简便. 下节给出了带权的以Chebyshev 多项式为基函数的离散最佳平方逼近多项式； 参考代码disc_cheb_appr_W.m 关于教材上介绍方法的改进版本可参考代码OrthpolyfromPts.m. 从实用的角度似乎没必要，但是从离散的数据点构造一个“带权的”正交多项式也是很有趣的过程.

4.4 离散的最佳一致逼近多项式算法实现(选讲)

上一小节给出的是离散的最佳平方逼近多项式, 本节我们借助于Lawson迭代，以最佳平方逼近为迭代初值，给出离散的最佳一致逼近. 给定数据$(x_j, f_j)_{j=1}^M$， 记$p_\infty(x)$是离散的最佳一致逼近多项式, 即

$$\max_{1\le j\le M} |p_\infty(x_j) - f_j| = \min_{\forall p\in H_n} \max_{1\le j\le M} |p(x_j) - f_j|$$

则$p_\infty(x)$可视为某个带权W的 离散最佳平方逼近多项式, 即

$$\sum_{j=1}^{M} w_j |p_\infty(x_j) - f_j|^2 = \min_{\forall p\in H_n} \sum_{j=1}^{M} w_j |p(x_j) - f_j|^2$$

这样，计算最佳一致逼近转化为计算某个带权的最佳平方逼近；而计算目标也转化为寻找相应的权重. Lawson迭代给出了以误差大小为 迭代权重的算法. 读者可测试代码，以检验最大模误差和L_2模的差别.

```
    N = 801;    X = linspace(-1,1,N)';   f = 1./(1+25*X.^2);
%     output pts   and func values
    t = linspace(-1,1,N*2)';    ft = 1./(1+25*t.^2);
```

[15]对比单项式基函数，教材给出方法的数值稳定性有提高. 当n不太大时，该方法奏效. 但教材称之为当前最好的方法，是不准确的. 当前最好的方法应该是 Trefethen, Vandermonde with Arnoldi, SIREV2021 和 张雷洪，苏仰峰，李仁仓的论文, Accurate Polynomial Fitting and Evaluation via Arnoldi, 2022

```
    n  = 11;     y1 = disc_cheb_appr_W(X,f,t,n);

%     lawson iteration for maximun norm approx
    W = 1/N*ones(N,1); Err2 = []; ErrInf = [];
    for k = 1:11
        [y,d] = disc_cheb_appr_W(X,f,X,n, W);
        r = abs(y - f);
        W = (W.*r)./(r'*W);
        Err2 = [Err2, norm(y-f,2)]; ErrInf = [ErrInf,norm(y-f,inf)];
    end
    y = mychebpolyval(d,t);
% %     output
    maxerr_l2 = norm(y1-ft,inf),  maxerr_lawson = norm(y-ft,inf),
    figure; LW = 'linewidth';  plot(X,f,t,y1,t,y,LW,1);
    title('lsq␣and␣minimax␣approximation'); legend('exact␣curve','lsq␣','minimax')
    figure; subplot(2,1,1); plot(t,ft-y,'r-',LW,1);
    title('AbsError␣of␣minmax␣approximation');
    subplot(2,1,2);  plot(t,ft-y1,'r-',LW,1);
    title('AbsError␣of␣lsq␣approximation');
    figure; subplot(2,1,1); plot(Err2,'o-'); title('L_2␣norm␣Error');
    subplot(2,1,2); plot(ErrInf,'*-'); title('L_{\infty}␣norm␣Error');
```

上述代码调用了加权的离散最佳平方逼近，这里给出代码.

```
function [y,d] = disc_cheb_appr_W(x1,f1,t,m, W)
% this algorithm implements the best least square approxmation
% for f1 = f(x1) at sample pts x1; min sum_j W_j*| f1 - p|^2
%
% intput: x1 is the sample pts
%         f1 is the values at x1 (m < n = length of x1)
%         m is the degree of polynomail of p
%         t is the nodes which p(t) need to be evaluated
%         W is weight vectors : V'*diag(W)*V*c = V'*(W.*f);
%
% output: d is the ceof of p by Tchebyshev polys, i.e., p(x) = sum_j d_j*T_j(x)
%         y = p(t)
% by Ya-nan Zhang, ynzhang@suda.edu.cn, 1/10/2022, Oct,1st,2022

a = x1(1);  b = x1(end);  n1 = length(x1);

if a*b ~= -1
   z = x1 - (b+a)/2;  z = z(:)/(b-a)*2;    % transf to [-1,1]
else
   z = x1(:);
```

```
end

B = ones(n1,m+1); B(:,2) = z;            % Tchebyshev series
for j = 2:m
    B(:,j+1) = (2.*z).*B(:,j) - B(:,j-1);
end

if nargin < 5
    d = B \ f1;       % least square meth
else
    W = sqrt(W);      W = W/sum(W(:));
    dW = sparse(1:n1,1:n1,W(:),n1,n1);
    d = (dW*B)\(W(:).*f1(:));      % weighted ls meth equals following 3 lines
%        W = W/sum(W(:));
%        dW = sparse(1:n1,1:n1,W(:),n1,n1);
%        d = (B'*dW*B)\(B'*(W(:).*f1(:)));
end
%                            % d: coefs of T_k
t = t(:);  zt = ( 2*t - (a+b) ) / (b - a);  % transf to [-1,1]
A = ones(size(zt,1), m+1);  A(:,2) = zt;
for j = 2:m
    A(:,j+1) = (2.*zt).*A(:,j) - A(:,j-1);
end
y = A*d;
```

标注 12 这里采用Lawson迭代计算离散最佳一致逼近多项式，有限步迭代只得到近似值。理论上当迭代步数$n \to \infty$时，极限为最佳一致逼近多项式。但Lawson迭代收敛较慢，需要进一步寻找加速的方法[16].

5 最佳平方三角逼近与三角插值

问题：如何逼近光滑的周期函数？ 正交多项式或者分段低次插值是否可行？

答：可行但不高效！ 形如下式的三角多项式更好！

函数的Fourier展开

$$f(x) \sim \sum_{k=0}^{\infty} a_k \cos kx + b_k \sin kx = a_0 + \sum_{k=1}^{\infty} \Big(a_k \cos kx + b_k \sin kx \Big)$$

[16]计算离散的最佳一致逼近多项式，目前计算效率最好的是内点法：Linyi Yang, Lei-Hong Zhang and Ya-Nan Zhang The Lq-weighted dual programming of the linear Chebyshev approximation and an interior-point method；从实用的角度出发，Vandermonde with Arnoldi 结合 Lawson 迭代最为简便易行。

> 利用三角函数正交性, 依次将$\sin lx$, $\cos lx$于上式作内积，可得系数表达式
>
> $$a_0 = \frac{1}{2\pi}(f,1) = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)dx$$
> $$a_l = \frac{1}{\pi}(f, \cos lx), \quad b_l = \frac{1}{\pi}(f, \sin lx),$$

给定周期函数$f(x) \in C_p[0, 2\pi)$, 利用三角函数的正交性，可知

$$P_N(x) = a_0 + \sum_{k=1}^{N}\Big(a_k \cos kx + b_k \sin kx\Big) \tag{4}$$

$f(x)$是在空间

$$\Phi = \text{span}\{1, \sin x, \cos x, \sin 2x, \cos 2x, \cdots, \sin Nx, \cos Nx\}$$

中的最佳平方逼近函数，其中系数a_k, b_k定义如上。取$(0, 2\pi)$上等距节点采用梯形公式计算a_k, b_k即可, 当节点数$M \gg N$时 梯形公式具有机器精度。

$$a_l = \frac{1}{\pi}\frac{2\pi}{M}\sum_{j=0}^{M-1} f(x_j)\cos(lx_j) = \frac{1}{M}\sum_{j=0}^{M-1} f(x_j)\cos(ljh), \quad x_j = jh = j\frac{2\pi}{M}$$

若取$M = 2N+1$, 即是下节介绍的三角插值.

```
% % Fourier Series
% f ~ p = a0 + sum_k ak*cos(kx) + bk*sin(kx)
L = 2*pi;    f = @(x) exp(sin(2*x) + cos(x));
n1 = 4;     M = max(3*n1,32);    h = L/M;
Xnodes = (0:h:L-h)';     fX = f(Xnodes);
a0 = sum(fX)/M;          j1 = (1:n1);
C =  cos(Xnodes*j1);    ak = (C'*fX)*(2/M);
S =  sin(Xnodes*j1);    bk = (S'*fX)*(2/M);
% % --- compute p(s)
s = (0:0.01:L)';    C = cos(s*j1);   S = sin(s*j1);
p = a0  + C*ak + S*bk;
err_inf = norm(p(:) - f(s), inf),
figure; LW = 'LineWidth';
subplot(2,1,1);  plot(s,f(s),'r-',s,p,'k-.',LW,1);
title('Fourier␣Series␣of␣f(x)'); grid on;
subplot(2,1,2);  plot(s,p-f(s),'k-',LW,1);
title('AbsError␣Fourier␣Series');  grid on;
```

5.1 复数形式的最佳平方三角逼近

为了书写简洁，引入Euler 公式, 首先给出指数函数的幂级数

$$e^x = 1 + x + \frac{x^2}{2!} + ... + \frac{x^n}{n!} + ...$$

令$x = i\theta$ 带入上式，分离实部和虚部 得到

$$\mathrm{e}^{i\theta} = \cos\theta + i\sin\theta$$

进而sine和cosine函数可以由指数形式表示

$$\cos x = \frac{e^{ix} + e^{-ix}}{2}, \quad \sin x = \frac{e^{ix} - e^{-ix}}{2i}$$

以及著名的数学公式 $\boxed{e^{i\pi} + 1 = 0.}$

引入复数和Euler公式，可得以2π为周期的函数Fourier展开

$$f(x) = \sum_{|k|<\infty} \widehat{f}_k \mathrm{e}^{\mathrm{i}kx}; \quad \widehat{f}_k = \frac{1}{2\pi}\int_0^{2\pi} f(x)\mathrm{e}^{-\mathrm{i}kx}dx$$

及如下形式的最佳N次三角平方逼近

$$P_N(x) = \sum_{k=-N}^{N} \widehat{f}_k \mathrm{e}^{\mathrm{i}kx} \tag{5}$$

例 17 (**留作研究生作业**) 证明: 对以2π为周期的 实函数，由(4) 定义的最佳平方逼近多项式$P_N(x)$, 与(5)定义的表达式是等价的. 并推导系数a_k, b_k与$\hat{f}_k$之间的关系.

例 18 (**证明留作习题**) 证明: 对以2π为周期且具有连续q阶导数的函数$f(x)$，由 (5)定义 的最佳平方逼近$P_N(x)$满足如下估计

$$\|f - P_N\|^2 \le \|f\| \cdot |f|_q \cdot \frac{1}{N^q}$$

其中$|f|_q = \int_0^{2\pi} f^{(q)}(x)dx$表示$H^q$半范数.

提示: 利用最佳平方逼近的误差估计

$$\|e_n\| = \sqrt{\int_0^{2\pi}\Big[P_N(x) - f(x)\Big]^2 dx} = \sqrt{\Big(f - P_N, f - P_N\Big)} = \sqrt{\Big(f, f - P_N\Big)}$$

进一步分析得到

$$\begin{aligned}
&\Big(f, f - P_N\Big) = \Big(f, \sum_{|k|>N} \hat{f}_k e^{ikx}\Big) \\
&= \sum_{|k|>N} \widehat{f}_k \Big(f, e^{ikx}\Big) = \sum_{|k|>N} \widehat{f}_k \int_0^{2\pi} f(x)e^{ikx}dx \\
&= \sum_{|k|>N} \widehat{f}_k \frac{1}{(-ik)} \int_0^{2\pi} f'(x)e^{ikx}dx \\
&= \sum_{|k|>N} \widehat{f}_k \frac{1}{(-ik)^q} \int_0^{2\pi} f^{(q)}(x)e^{ikx}dx \\
&\le \|f\| \cdot |f|_q * \frac{1}{N^q}
\end{aligned}$$

上式估计用到Parseval, Cauchy不等式. 当$f(x)$无穷光滑; 最佳平方三角逼近三角插值 能达到谱精度——指数级收敛。

标注 13 由Euler-Maclaurin公式，复合梯形公式计算光滑周期函数的积分时可以得到高精度结果。计算数值积分时，选取等分节点个数大于等于$2N+1$即可. 实际应用中，节点个数的选择和逼近系数在数目上是匹配的. 此时即是常用的三角多项式插值法.

设$f(x)$ 是以$L>0$ 为周期的函数，则其最佳平方逼近三角多项式

$$f(x) \approx P_N(x) = \sum_{k=-N}^{N} \widehat{f}_k \mathrm{e}^{\mathrm{i}kx\frac{2\pi}{L}}, \quad \widehat{f}_k = \frac{1}{L}\int_0^L f(x)\mathrm{e}^{-\mathrm{i}kx\frac{2\pi}{L}}\,dx \tag{6}$$

取 $M \gg N, x_j = j\frac{L}{M}$, 则 系数可由下式“精确”计算

$$\widehat{f}_k = \frac{1}{M}\sum_{j=0}^{M-1} f_j e^{-ikj\frac{2\pi}{M}} = \frac{1}{M}\sum_{j=0}^{M-1}\left(f_j e^{iNj\frac{2\pi}{M}}\right)e^{-i(k+N)j\frac{2\pi}{M}}, \quad -N \le k \le N$$

```
function [p,ck] = Four_Series(f,L,N,s)
% % Fourier Series on [0,L]
% f ~ p = sum_k ck*exp(ikx*(2*pi/L))
% ck = 1/L*int_0^L f(x)exp(-ikx*2*pi/L);
% input:   f:  given function
%          L:  length of periodic
%          N:  number of Fourier Series terms (-N<k<N)
%          s:  nodes be evaluated
% output:  ck: fourier coeffs
%          p:  P_N(s)
if nargin < 1
   L = 2;     f = @(x) exp(sin(2*pi*x) + cos(pi*x));
   N = 4;     s = linspace(0,L,401)';
elseif nargin<4
   s = linspace(0,L,401)';
end

M = max(3*N,32);    h = L/M;      Xnodes = (0:h:L-h)';     fX = f(Xnodes);
w = 2*pi/M;
F = exp(-1i*w*(-N:N)'*(0:M-1))/M;    ck = F*fX;
% % --- compute p(s)
Fs = exp(1i*2*pi/L*s*(-N:N));      p = Fs*ck;
if isreal(fX),      p = real(p);      end

if nargin < 1
figure; LW = 'LineWidth';
subplot(2,1,1);  plot(s,f(s),'r-',s,p,'k-.',LW,1);
title('Fourier␣Series␣of␣f(x)'); grid on;
subplot(2,1,2);  plot(s,p-f(s),'k-',LW,1);
title('AbsError␣Fourier␣Series');  grid on;
figure; semilogy((-N:N)', abs(ck), 'ro--');
```

```
title('AbsValue␣Fourier␣mode␣ck'); xlabel('-N<k<N')
end
```

5.2 三角插值

例 19 给定周期函数$f(x)$, 记

$$S(x) = a_0 + \sum_{k=1}^{N} \left(a_k \sin kx + b_k \cos kx\right)$$

对$[0, 2\pi)$ 做$2N+1$等分, 记节点$x_j = jh$, $h = \frac{2\pi}{2N+1}$, 令$f(x)$与$S(x)$在节点处相等，即

$$S(x_j) = f(x_j), \quad 0 \le j \le 2N$$

证明：满足上式插值条件的解a_k, b_k存在唯一。

提示：验证系数矩阵的列向量彼此正交。（留作研究生作业）

例 20 (验证 e^{ikx} 的正交性) 对任意整数k, l,

1. 验证复内积满足

$$\int_0^{2\pi} e^{ikx} \cdot e^{-ilx} dx = 2\pi * \delta_{kl} = 2\pi * \begin{cases} 1, & k = l \\ 0, & k \neq l \end{cases}$$

2. 将区间$[0, L]$ N等分，记$x_j = \frac{jL}{N}$, 检验离散的网格函数满足

$$\sum_{j=0}^{N-1} e^{ikx_j*2\pi/L} \cdot e^{-ilx_j 2\pi/L} = \sum_{j=0}^{N-1} e^{ikj*2\pi/N} \cdot e^{-ilj2\pi/N} = N * \delta_{kl}$$

留作研究生作业。 ■

复值函数三角插值

给定目标周期函数$f(x)$或者在等距节点处的函数值，可根据插值理论提出问题[17]

> 考虑$[0, L]$上周期函数$f(x)$, 已知等距节点处的函数值
>
> $$\{f(x_j)\}_{j=0}^{2n}, \quad x_j = jh, \quad h = L/(2n+1).$$
>
> 以
>
> $$\{1,\ \mathrm{e}^{\pm \mathrm{i}x\frac{2\pi}{L}}\ \mathrm{e}^{\pm \mathrm{i}2x\frac{2\pi}{L}},\ \ldots, \mathrm{e}^{\pm \mathrm{i}nx\frac{2\pi}{L}}\}$$
>
> 为插值基函数，确定如下形式的近似函数
>
> $$P_n(x) = \sum_{k=-n}^{n} \widehat{f}_k e^{ikx\frac{2\pi}{L}} \qquad \text{s.t.} \quad P_n(x_j) = f(x_j), \quad 0 \le j \le 2n \tag{7}$$
>
> 结合上节例题可知系数 $\widehat{f}_k$是唯一可解的.

[17]李庆扬 教材上关于复数形式的最佳平方三角逼近是错误的！ 其中实部只含有 余弦，虚部只含有正弦！

下面给出初等方法计算$\widehat{f}_k$ 的过程： 对$-n \le l \le n$, 两端同时乘以$e^{-i*lx_j \frac{2\pi}{L}}$ 并关于 j 求和

$$\sum_{j=0}^{2n} e^{-i*lx_j \frac{2\pi}{L}} f(x_j) = \sum_{j=0}^{2n} \sum_{k=-n}^{n} \widehat{f}_k e^{i*(k-l)x_j \frac{2\pi}{L}}$$

代入$x_j = jh = j/(2n+1)$, 交换求和顺序; 验证: 当$k \ne l$时

$$\sum_{j=0}^{2n} \left[e^{\left(i*(k-l)\frac{2\pi}{2n+1}\right)} \right]^j = (2n+1)\delta_{kl} = \begin{cases} 2n+1, & k = l \\ 0, & k \ne l \end{cases}$$

于是，对$-n \le l \le n$， 成立下式

$$\boxed{\widehat{f}_l = \frac{1}{2n+1} \sum_{j=0}^{2n} e^{-i*lx_j \frac{2\pi}{L}} f(x_j) = \frac{1}{2n+1} \sum_{j=0}^{2n} f(x_j) e^{-i*lj \frac{2\pi}{2n+1}}}$$

给定区间$[0, L]$ 上的周期函数$f(x)$, 三角插值计算步骤如下：

1) 给出网格等分数$N = 2n+1$, 网格节点x_j, 以及$f(x_j)$

2) 计算插值系数$\widehat{f}_l = \frac{1}{N} \sum_{j=0}^{N-1} f(x_j) \exp(-ilj\frac{2\pi}{N}), \quad -n \le l \le n$

3) 按表达式

$$P_n(x) = \sum_{k=-n}^{n} \widehat{f}_k \exp(i * kx\frac{2\pi}{L})$$

输出需要计算的x值

```
% --- tri poly by Fourier matrix
L = 2; f = @(x) exp(sin((2*pi)*x) + cos(pi*x));
n1 = 11; M = 2*n1+1;
h = L/M; x1 = (0:h:L-h)'; f1 = f(x1);
% --- compute fv by fx
j1 = 0:M-1; j2 = j1-n1;
F1 = exp((1i*2*pi/M)*(j1.'*j2));  fv = (F1'*f1)/M;
% -- the pts and exact values for compare
xx1 = 0:0.01:L; f1_ex = f(xx1);
F2 = exp(1i*(xx1'*j2)*(2*pi/L));
f11 = real(F2*fv);  norm(f11(:) - f1_ex(:),inf),
figure; subplot(2,1,1);  plot(x1,f1,'ro',xx1,f11,'k-.'); grid on;
subplot(2,1,2);  plot(xx1,f1_ex(:)-f11,'k-','LineWidth',1); grid on;
```

标注 14 对于无穷次光滑的周期函数，三角插值效果好且计算简单。 若特殊问题 n 较大时，有FFT 可以使用且数值稳定性更好.

5.3 偶数节点插值和FFT 简介

FFT 就是 DFT (Discrete Fourier Transform)。 因为可以快速计算（Fast），通常称为 Fast Fourier Transform 简称FFT[18]。 回顾三角插值的表达式和系数$\widehat{f_\ell}$

$$P_n(x) = \sum_{k=-n}^{n} \widehat{f_k} \exp(i * kx\frac{2\pi}{L}), \qquad \boxed{\widehat{f_\ell} = \frac{1}{2n+1}\sum_{j=0}^{2n} f(x_j)\exp(-i * \ell j\frac{2\pi}{2n+1})}$$

需要指出：对$[0, L]$进行$(2n+1)$奇数等份纯粹是为了推导时书写方便(看起来更对称)， 实际应用时更多选择偶数等分，特别是2的幂次。

记$M = 2n$, $\omega = \exp(-1i * \frac{2\pi}{M})$, 忽略三角插值多项式的最后一项，可得

$$P_n(x) = \sum_{\ell=-n}^{n-1} \widehat{f_\ell} \exp(1i * \ell x\frac{2\pi}{L}), \qquad \boxed{\widehat{f_\ell} = \frac{1}{M}\sum_{j=0}^{M-1} f_j\omega^{\ell j}, \quad -n \le \ell \le n-1}$$

改写$\widehat{f_\ell}$为

$$\widehat{f_\ell} = \frac{1}{M}\sum_{j=0}^{M-1} f_j\omega^{\ell j} = \frac{1}{M}\sum_{j=0}^{M-1} (f_j * \omega^{-nj})\underset{\sim\sim\sim}{\omega^{(\ell+n)j}} = \frac{1}{M}\sum_{j=0}^{M-1} g_j\underset{\sim\sim\sim}{\omega^{(\ell+n)j}},$$

其中$\omega^{(-nj)} = e^{1i*\pi*j} = (-1)^j$, 进而 $g_j = (-1)^j f_j$, 则计算系数$\widehat{f}$可由矩阵乘法$F * \vec{g}$实现，其中Fourier矩阵

$$F = \begin{bmatrix} 1 & 1 & & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{M-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(M-1)} \\ & & & \ddots & \\ 1 & \omega^{M-1} & \omega^{2(M-1)} & \cdots & \omega^{(M-1)(M-1)} \end{bmatrix}$$

标注 15 形式上看，M 阶矩阵乘以向量需要计算量$\boldsymbol{O}(M^2)$, 但是由于Fourier矩阵的特殊结构 该计算量可以降低到$\boldsymbol{O}(M\log M)$. 注意到矩阵的元素全是$\omega = e^{\frac{2\pi i}{M}}$ 的某一个幂次，而ω 是乘法单位元1在复数域的M重根，也即是

$$\left\{1 = \omega^0, \omega^1, \omega^2, ..., \omega^{M-1}\right\}$$

构成一个乘法循环群；ω的所有幂次全在上述集合里。形式上矩阵F有M^2个元素，实际上F 只有M个不同元素，也即是矩阵的第二行，他们在F里重复出现。进一步观察发现即使仅有M个不同元素， 他们共轭成对的出现在复平面的单位圆周上，实际上仅有$M/2$个不同的元素。 通过巧妙的计算技巧，避免重复计算，可以将算法复杂度降低[19]。 FFT 在科学计算中已经类似于四则运算、成为一个成熟的基础算法，细节不做讨论。

下段测试代码给出了以L 为周期的函数，在M 个等距节点上做三角插值 得到的复三角多项式系数c以及在节点s处的输出值.

[18]古龙：名字会起错、外号不会错！

[19]据记载从Gauss开始就有这种考虑， 现代文献一般引用Tukey & Cooley 65年发表的文章。也许其他人更早发现，Jack Good 1958年发表过类似文章。 Tukey 的65年的文章也引用了Good的文章。

```
function [p, c] = trig_interp_even(f,L,M,s)
% compute the trig_poly interpolation on (X,fX)
% sum_k (-n<k<n-1)ck*e^ikx
if mod(M,2)==1, error('Grid␣number␣M␣shound␣be␣even!'); end
   h = L/M;  Xnodes = (0:h:L-h)';  fX = f(Xnodes);
   g = fX.*((-1).^(0:M-1)');    c = fft(g) / M;     % by fft
% evalute the trig polynomial on s
   F = exp(1i*2*pi/L* (s*(-M/2:M/2-1)) );   p = F*c;
if isreal(fX), p = real(p); end
```

标注 16 以上测试代码均是对已知函数进行三角插值或者逼近. 实际应用时，我们往往只知道 样本点的数据$\{(x_j,y_j)\}_{j=0}^{M}$, 而不知道函数的表达式，此时基于数据的三角插值就 更为方便了. 另外，注意到测量数据往往不能保证自变量是"精确"等距的， 如果直接调用FFT计算会得到错误结果. 但是基于插值条件的方程组仍然成立，我们可 通过解方程的方法求出系数c. 由e^{ikx}的正交性可知，即使插值节点是非等距网格，所 求解矩阵的条件数依然很小. 另一方面， 数据来源往往带有噪声，实际应用时更多采用的是逼近（最小二乘）的方法来计算离散的 最佳平方三角多项式。例如去掉高频数据可使得曲线更加光滑，读者可测试并观察下段代码。

```
   L = 2; f = @(x) exp(sin((4*pi/L)*x) + cos(2*pi/L*x));
   M = 1000;     h = L/M;      X = (0:h:L-h)';
   rng(0);       fX = f(X)  + 0.1*randn(size(X));
% fourier lsq
   n = 5; F = exp(1i*2*pi/L*X*(-n:n));  fc = F\fX;  p = real(F*fc);
   figure; LW = 'LineWidth';
   subplot(3,1,1);  plot(X,fX,'r-',X,p,'k-',LW,2);
   title('trig␣apporx␣of␣f(x)'); grid on;
   subplot(3,1,2);  plot(X,p-f(X),'k-',LW,1);
   title('AbsError␣Fourier␣Series');  grid on;
   subplot(3,1,3);  semilogy((-n:n)', abs(fc), 'ro--');
   title('AbsValue␣Fourier␣mode␣ck'); xlabel('-N<k<N');
```

上机练习

1. 输出m阶Hilbert矩阵H_m (hilb)，并计算行列式(det)，条件数(cond); 取$m = 3,6,9,12$，观察Hilbert矩阵的行列式和条件数[20].

2. 取 $n = 20$, 将区间$[0,1]$n等分得到$\{x_k\}_{k=0}^{n}$，取$m = 2$; $\{\phi_j = x^j\}_{j=0}^{m}$,定义离散的内积

$$A_{ij} = \sum_{k=0}^{n} \phi_i(x_k) * \phi_j(x_k)$$

[20] 结果说明Hilbert矩阵的数值奇异性（病态性）.

输出矩阵

$$A = (A_{ij})_{(m+1)\times(m+1)}$$

并计算A的条件数；和H_3做比较； 再取$m = 5, 11$ 计算A 并与H_6, H_{11}比较条件数 [21].

2. 取m = 4; n = 2m+1, 将区间$[0, 2\pi]$n等分得到$\{x_i\}_{i=0}^{n}$，取

$$\phi_0(x) = 1; \phi_1(x) = \sin(x),\ \phi_2(x) = \cos(x),\ ...,\ \phi_{2k-1}(x) = \sin(kx),\ \phi_{2k}(x) = \cos(kx)$$

记

$$A(:, 2k) = \cos(x_j),\ A(:, 2k+1) = \sin(x_j),\ j = 0 : n-1$$

输出 A并计算

$$A^T * A$$

检验其为对角矩阵，进而表明A的列向量是彼此正交的。继续取m = 5, 6, 10或者更大的整数，验证并发现规律。 [22].

3. 利用Matlab反斜杠命令求解超定方程 $Ax = b$, 并与$(A^T * A)x = A^T * b$比较计算结果 [23].

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 2 & 2 \\ -1 & 2 \\ 3 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 5 \\ 6 \\ 2 \\ 4 \end{bmatrix}$$

4. 取Chebyshev-Lobatto点

$$x_j = \cos(j * \pi/n), \quad 0 \le j \le n$$

分别对函数

$$f(x) = \sin(\pi * x), \qquad f(x) = x/2 + |x| - x^2, \qquad x \in [-1, 1]$$

做多项式插值；取 $n = 10, 20, 40$, 在等距节点$t = -1 : 0.01 : 1$,上输出结果，并检验与精确解 的误差 [24].

5. 对Runge函数

$$f(x) = \frac{1}{1+25x^2}, \quad x \in [-1, 1]$$

取$N = 1001$, 记$x_j = -1 + \frac{2j}{N}$, $j = 0, ..., N$. 对离散的数据$\Big(x_j, f(x_j)\Big)$, 计算离散的m次最佳平方逼近多项式

$$p_m(x) = \sum_{k=0}^{n} a_k T_k(x), \quad T_k(x)\text{为Chebyshev多项式}$$

[21]本题的矩阵A即是法方程组的系数矩阵；结果说明 离散的"Hilbert矩阵"A, 与上题Hilbert矩阵具有类似的条件数.

[22]本题目标检验Fourier矩阵的正交性

[23]两种方式得到的解误差在1e-15量级，说明理论上反斜杠对超定矩阵用的是最小二乘.

[24]本题说明：对光滑函数(无论是否有周期性)chebyshev点插值效果好，而非光滑函数效果差.

要求:

1. 设$m = 20$, 计算$p(x_j)$ 并画出误差曲线$f(x_j) - p(x_j)$; 输出 $e_m = \max_{0\le N} |f(x_j) - p(x_j)|$

2. 取不同的$m = 10, 20, 30, ..., 100$, 计算相应的误差e_m; 画出误差随m变化的曲线: 横坐标为 m , 纵坐标为e_m.

小结和拓展(选讲)

本章前半部分介绍了函数$f(x)$的多项式逼近和三角多项式逼近. 无论是已知函数表达式或者 样本数据，我们通过最小二乘的思想给出了(带权)最佳平方逼近多项式或者三角多项式的具体形式. 无论是多项式或者三角多项式, 均可由正交基函数表示为其线性组合的形式. 这对进一步分析函数$f(x)$的性质和计算都有积极意义. 例如，对$f(x)$求导和求积分 都转换为对$f(x)$的最佳逼近(三角)多项式进行计算. 第四章数值积分完全可看作是对插值和逼近的应用. 许多初学者会误以为计算导数是容易的，特别是有表达式的光滑函数. 借助于求导的四则运算、复合函数链式法则，理论上没问题. 但是可以试想：如果$f(x)$本身就是多重复合函数且带有分式, 其结果也是很繁琐的. 如果$f(x)$是光滑函数，我们可以先求其高阶的最佳逼近多项式Q，进而求Q的导数作为 $f'(x)$的近似. 这一思想的应用对应了偏微分方程数值解中的一类重要方法——谱方法. 当然，如果函数本身不光滑或者只有一些离散点的值，数值求导仍然是困难的！

```
function [p1, ck] = Cheby_Series(f,n,t)
%  chebyshev series of f   // near best polynomial
%  Q = sum_k ck * Tk,      % Tk is Chebyshev poly degree(k)
%  ck = pi/2*(f,Tk) = pi/2*\int f(x)*Tk(x)/sqrt(1-x^2) dx
%  ck = pi/2*int_0^pi f(t)cos(k*t) dt   % can be computed by DCT with
%  large nubmer of pts N, which gives accurate ck
% input: f is given function
%         n is degree of best approximation polynomial Q
%         t is the pts be evaluate
% output:  p1 = Q(t);
%           c is coeffs of Q in Chebyshev Series
% ynzhang@suda.edu.cn, 16/10/2022
% note: In this code, ck is computed exactly, it is not cheby pts interpolation
N = max(64,2*n);     j0 = (0:N)';    Xp = cos(pi/N*j0);
f1 = f(Xp);  f1([1,end]) = 0.5*f1([1,end]);
% C = cos(pi/N*(j0*j0'));    ck = (C*f1);         % can be replaced by mydct
ck = mydct0(f1);
ck = ck(1:n+1)*2/N;  ck(1) = 0.5*ck(1);
  if nargin < 3
      p1 = [];
  else
```

```
        T = get_cheb(t,n);        p1 = T*ck;
    end
```

标注 17 以上代码给出了计算光滑函数的Chebyshev级数过程. 整个计算复杂度只有N次乘法和一次DCT. 这样$f(x)$ 就替换为Chebyshev有限和Q_N. 针对f的一切数值运算都由Q_N来完成, 大大简化了运算过程. 这也是chebfun 的核心思想——通过系数c_k的运算替代函数$f(x)$的运算, 即保证计算效率又保留了函数Q_N的表达式. 对Matlab平台, 即做到了像数组运算一样快, 又像函数运算一样方便! 例如, 借助于Chebyshev循环式计算 Q_N的一阶和二阶导数. 有了导数可借助于Newton法计算其零点、极值点.

```
function g = chebpolyval_diff(x,c,vargin)
% for given points x and coeffs c
% it computes the chebyshev series  Q = sum_k ck*Tk(x) at 'x'
% and also its 1st and 2nd derivatives Q'(x) and Q"(x)
%
%  input: x : the nodes should be evaluated
%           c : coeffs of Tchebyshev Series
%           vargin: 0 -> g = Q  \\
%                   1 -> g = Q' \\
%                   2 -> g = Q"
x  = x(:);  N = length(x);   m = length(c) - 1;
T = zeros(N,m+1);  dT = T;   d2T = T;
% %  inital
T(:,1) = 1;  T(:,2) = x;
dT(:,2) = 1; dT(:,3) = 4*x;  d2T(:,3) = 4;
% %  recursive
for j = 2:m
    T(:,j+1) = 2*x.*T(:,j) - T(:,j-1);
    if j>2
    dT(:,j+1) = ( 2*T(:,j) + 1/(j-2)*dT(:,j-1) )*j;
    d2T(:,j+1) = ( 2*dT(:,j) + 1/(j-2)*d2T(:,j-1) )*j;
    end
end
% T: Chebyshev Vandermonde matrix // dT and d2T defined similarly
% % output
    if vargin==0
        g = T*c;
    elseif vargin==1
        g = dT*c;
    elseif vargin == 2
        g = d2T*c;
    end
```

上机练习2

1. 给定被逼近函数

$$f(x) = \sin(\pi * x), \quad -1 \le x \le 1$$

编写程序计算其最佳平方逼近3次多项式；并画出相应的曲线比较； 进一步测试最佳平方逼近4次，5次，6次多项式； 说出观察到的现象。说明：此处的多项式空间的基函数可以选择单项式$1, x, x^2, x^3$; 也可选择Legendre多项式作为基函数。

2.分别使用三角插值和Chebyshev零点插值确定函数

$$f(x) = x^2 \cos x; \quad f(x) = \exp(\cos x)$$

取n = 10，20并与精确解比较误差 [25].

6 有理逼近(略)

对于光滑函数，Chebyshev多项式插值或者离散逼近均可；周期函数可采用三角插值。但是对于光滑度不高， 特别是有奇异性的函数，有理逼近是更佳的选择。例如：李庆扬教材给的例子$\log(1+x)$, 或者

$$f(x) = |x| + x/2 - x^2, \quad f(x) = \tan(x)$$

可以测试Chebyshev插值(级数) 对绝对值函数只有$O(n^{-1})$的精度。 寻找合适的有理逼近，特别是最佳有理逼近较为困难。有兴趣读者可以阅读文献 [26] 及其参考文献。 下面我们给出一个简单的算法来计算有理逼近(一般称为Cauchy方法).

考虑被逼近函数$f(x) \in C_{[a,b]}$, 若已知函数或者其导数存在奇异性，则较为理想的逼近方法是有理逼近

$$f(x) \approx \frac{p(x)}{q(x)} = \frac{\sum_{j=1}^m a_j x^j}{\sum_{j=1}^n b_j x^j} := \frac{N(x)}{D(x)} = R_{m,n}(x)$$

假设样本点$x = \{x_k\}_{k=1}^M$及数据f_k已知，则计算目标为

$$\min_{a,\, b} \sum_{k=1}^M \left[f_k - \frac{N(x_k)}{D(x_k)} \right]^2$$

对上式线性化

$$\min_{c=[a,\, b],\ \|c\|=1} \sum_{k=1}^M w_k \left[f(x_k) \cdot \sum_{j=1}^n b_j x_k^j - \sum_{j=1}^m a_j x_k^j \right]^2$$

其中$w_j \approx D(x_j)^{-1}$ 可视为权函数。 记

$$V_n = \left[1, x, x^2, ..., x^n\right], \quad D_f = \mathrm{diag}(f_j), \quad W = \mathrm{diag}(w_j)$$

[25]一般函数Cheby poly 效果好； 周期函数 Tri poly更佳.

[26]Trefethen, The Carathéodory–Fejér Method for Real Rational Approximation, SIAM Journal on Numerical Analysis, 1983; Trefethen, Robust Padé Approximation via SVD, SIAM REVIEW(2013) Vol. 55, No. 1, pp. 101–117

则计算目标为寻找单位向量$c=[a;b]$, 使得

$$W*\begin{bmatrix} D_f*V_n & \Big| & V_m \end{bmatrix}\cdot\begin{bmatrix} a \\ b \end{bmatrix}=W*A*c$$

的2范数取最小值, 即

$$\min_{\|c\|=1}\|WA*c\|$$

当$M\gg n+m$，且权矩阵W给定（例如W 取单位矩阵），可求其最小二乘解(Total Least Squares)。 可以证明：当$c=[a;b]$为A的最小右奇异向量时， $\min_{\|c\|=1}\|A*c\|$ 取最小值。

标注 18 上述算法对部分简单问题可能会取得较好的效果！优点在于m, n的选择比较自由。 缺点是算法极不稳定，例如对绝对值函数，即使放弃单项式基，选择Cheybshev多项式为基，效果仍不佳！

Theorem 12 (SVD分解) 设 $A\in\mathbb{C}^{m\times n}$, 且 $m\geq n$. 则存在如下分解： $A=U\Sigma V^T$, 其中 $U^TU=I_m, V^TV=I_n, \Sigma=\operatorname{diag}(\sigma_1,\ldots,\sigma_n)$, 满足 $\sigma_1\geq\cdots\geq\sigma_n\geq 0$. 正交矩阵$U$的列向量$u_1,\ldots,u_n$ 称为左奇异向量(left singular vectors). V 的列 $v_1,\ldots,v_n$ 称为 右奇异向量(right singular vectors). 非负实数σ_i 称为奇异值(singular values).

例 21 设$A\in\mathbb{R}^{m\times n}$, $m\geq n$,

- 验证$A^{\mathrm{T}}A$是对称矩阵，且$\forall x\in\boldsymbol{R}^n$, $x^{\mathrm{T}}(A^{\mathrm{T}}A)x\geq 0$
- 记$A=U\Sigma V^{\mathrm{T}}$, 检验

$$A^{\mathrm{T}}A=V\Sigma^2V^{\mathrm{T}}$$

- 根据矩阵分块规则，检验奇异值分解表达式

$$A=U\Sigma V^{\mathrm{T}}=\sigma_1u_1v_1^{\mathrm{T}}+\sigma_2u_2v_2^{\mathrm{T}}+\ldots+\sigma_nu_nv_n^{\mathrm{T}}$$

- 对任意的$x=\sum_{j=1}^n\alpha_jv_j$, 且$\|x\|=1$, 验证

$$\|Ax\|^2=\sum_{j=1}^n(\alpha_j\sigma_j)^2\geq\sigma_n^2$$

当$x=v_n$时，等号成立。

标注 19 由上例可知，当x取最小右奇异向量v_n时，$\|Ax\|$取最小值，即

$$\|Av_n\|=\min_{\forall x\in\mathbb{R}^n}\|Ax\|=\sigma_n$$

这就是给出了齐次方程的最小二乘解计算方法.

若考虑方程 $Ax=b$的完全最小二乘解, 可将问题转化为齐次问题

$$\min_{x\in\mathbb{R}^n}\|Ax-b\|=\min_{y\in\mathbb{R}^{n+1}}\|\begin{bmatrix}A,-1\end{bmatrix}y\|$$

其中矩阵$[A,-1]$表示将矩阵的最后一列扩展为-1.

6.1 Adaptive Antoulas – Anderson (AAA algorithm)

本节内容取自文献 [27]. 插值理论部分提到了稳定性好的 Barycentric interpolation polynomial 的形式

$$r(z) = \frac{n(z)}{d(z)} = \sum_{j=1}^{m} \frac{w_j f_j}{z - z_j} \Big/ \sum_{j=1}^{m} \frac{w_j}{z - z_j}$$

1. 当 $w_j = \dfrac{1}{\prod_{k\neq j}(z_j - z_k)}$ 时，$r(z)$ 表示满足插值条件$\{f_j = r(z_j)\}$ 的多项式。
2. 当w_j选为自由变量时，$r(z)$对应一个$(m-1,\ m-1)$的有理式。
3. 上述表达式满足

$$\lim_{z\to z_j} r(z) = \lim_{z\to z_j} \sum_{j=1}^{m} \frac{w_j f_j}{z - z_j} \Big/ \sum_{j=1}^{m} \frac{w_j}{z - z_j} = f_j$$

计算目标w

$$\min \left\| F * D - N \right\| = \min \left\| F_{sample} * C * w - C * f_{support} * w \right\| = \min \|A * w\|$$

其中 $C = \left(\frac{1}{x_i - z_j}\right)_{ij}$ 为Cauchy矩阵, x_i为输出节点，z_j为support points. 读者可以测试以下代码；结果表明对于非光滑函数(例如绝对值函数)，有理逼近效果优于多项式逼近。

```
f = @(z)  abs(z) + z/2   - z.^2;
x1 = -1:0.001:1;  x1 = x1(:); f1 = f(x1);
m = 120;  n = 22;
[y,d] = disc_cheb_appr(x1,f1,x1,m); % poly degree(m)
cheb_err = norm(y - f1, inf),
[R] = rat_aaa_nn(x1,f1,n) ;   % rational (n,n )
aaa_err = norm(R(x1) - f1, inf),
```

```
function [R, zj, wj, fj, mJ, J, err] = rat_aaa_nn(x1,f1,n)
%  rational approx R(n,n) on the sample data (x1,f1)
%  R(Z) = n/d = sum (wj*fj)/(Z-zj) / sum (wj)/(Z-zj)
%  zj is support pts  & , R(zj) = fj = f(zj) : interpolation case
% output:  R :  function handle,  R(x1) ~ f1;
%          zj:  support pts
%          wj:  weight of barycentric formula
%          fj:  f(zj)
%          mJ:  index of zj in x1
%          J :  index of x1 except zj
%         err:  maximum error of R(x1) - f1
%
M = length(x1);
F = f1(:);      SF = sparse(1:M,1:M,F,M,M);  % frequencies
J = 1:M;        Z = x1;
```

[27] Trefethen, The aaa algorithm for rational approximation, SINUM, 2018

```
zj = [];       fj = [];     C = [];        mJ = [];
R = mean(F);
for k = 1:n
    [~,j] = max(abs( F - R ));         %  find max error pts
    mJ = [mJ, j];        %     record the index
    zj = [zj; Z(j)];    %     zj is support pts
    fj = [fj; F(j)];
%   delete z from sample pts
    J(J==j) = [];
    C = [C,   1./ ( Z - Z(j) ) ];        %  Cauchy/Loewner matrix
    Sf = diag(fj);
    A = SF*C - C*Sf  ;
    [~,~,V] = svd(A(J,:), 0);    wj = V(:,end);
    wj = wj ./norm(wj,inf);
    num = C*(wj.*fj);    denom = C*wj;
%
    R = F;  R(J) = num(J) ./ denom(J);
    err = norm(F - R,inf);
    if err <  eps*norm(F,inf), break , end
end
 R = @(zz) reval(zz, zj, fj, wj);  cond(A(J,:))
%  zer_N = prz(wj.*fj,zj); zer_D = prz(wj,zj);
end
```

标注 20 理论上support pts 任意取即可得$R(m,m)$中满足

$$\min \|R * D - N\|$$

的有理式 D/N. 实际计算时，support pts的取法对算法影响很大。另外，重心坐标下有两种方式表达有理式

$$r(z) = \frac{n(z)}{d(z)} = \sum_{j=1}^{m} \frac{w_j f_j}{z - z_j} \Big/ \sum_{j=1}^{m} \frac{w_j}{z - z_j} \tag{8}$$

$$r(z) = \frac{n(z)}{d(z)} = \sum_{j=1}^{m} \frac{\alpha_j}{z - z_j} \Big/ \sum_{j=1}^{m} \frac{\beta_j}{z - z_j} \tag{9}$$

表达式(8)隐含了信息

$$\lim_{z \to z_j} r(z) = f_j$$

而(9)无此限制。

6.2 一般形式的有理逼近 $\boldsymbol{R}(m,n)$ （略）

AAA 算法要求分子、分母阶数相同。而有时候有理逼近问题有明确的多项式阶数的要求，

且$m \neq n$.

$$\mathcal{R}_{m,n} = \left\{ \frac{p}{q} : p \in \mathbf{R}_m[x], q \in \mathbf{R}_n[x] \right\}$$

设 函数f 和 正整数m,n,给定，计算

$$\min_{r \in \mathcal{R}_{m,n}} \|f - r\|$$

满足上式的r为最佳有理逼近。前文已提到最佳逼近难以取到。在有明确$(m \neq n)$要求的情况下，可以采用强制方法， 首先改写AAA算法的表达式为

$$r(z) = \frac{n(z)}{d(z)} = \sum_{j=1}^{n} \frac{\alpha_j}{z - z_j} \Big/ \sum_{j=1}^{n} \frac{\beta_j}{z - z_j}$$

记

$$\ell(z) = \prod_{j=1}^{n} (z - z_j)$$

则

$$r(z) = \frac{n(z)\ell(z)}{d(z)\ell(z)} = \frac{\sum_{j=1}^{n} \frac{\alpha_j}{z - z_j} \prod_{j=1}^{n} (z - z_j)}{\sum_{j=1}^{n} \frac{\beta_j}{z - z_j} \prod_{j=1}^{n} (z - z_j)} = \frac{p(x)}{q(x)}$$

形式上是一个$(n-1, n-1)$ 的有理式。 若系数$\alpha = [\alpha_1, ..., \alpha_n]$满足

$$V_m * \alpha = 0, \quad V_m = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_n \\ \vdots & \vdots & & \vdots \\ z_1^{n-1-m} & z_2^{n-1-m} & \cdots & z_n^{n-1-m} \end{bmatrix}$$

有如下结论[28]

Theorem 13 $r(z)$是 $(m-1, n-1)$次有理式， 即

$$\text{deg } (p) = m - 1, \quad p(z) = \sum_{j=1}^{n} \frac{\alpha_j}{z - z_j} \prod_{j=1}^{n} (z - z_j)$$

等价于 $\alpha \in \text{Null } (V_m)$,

$$\sum_{j=1}^{n} \alpha_j (z_j)^k = 0, \quad k = 0, 1, ..., n - 1 - m$$

容易检验

$$p(z) = \sum_{j=1}^{n} \alpha_j \prod_{l \neq j, l=1}^{n} (z - z_l) = b_{n-1} z^{n-1} + b_{n-2} z^{n-2} + ... + b_1 z + b_0$$

[28]Berrut, Jean Paul Mittelmann, Hans D. Matrices for the direct determination of the barycentric weights of rational interpolation, Journal of Computational and Applied Mathematics, 1997

的系数

$$b_{n-1}=\sum_{j=1}^{n}\alpha_j,$$

也即是最高次项系数是否为零等价于$\sum_j \alpha_j$是否为零，但是对于其它幂次检验较繁琐。

记

$$\Phi(z)=z^{n-1}p(\frac{1}{z})=b_{n-1}+b_{n-2}*z+b_{n-3}*z^2+...+b_0z^{n-1}$$

则

$$b_{n-1}=\lim_{z\to 0}\Phi(z),$$

而

$$\Phi(z)=z^{n-1}\sum_{j=1}^{n}\alpha_j\prod_{l\neq j,l=1}^{n}(1/z-z_l)=\sum_j\alpha_j\prod_{l\neq j}(1-z*z_l)\to\sum_j\alpha_j.$$

若$b_{n-1}=0$, 类似令(不引起歧义)

$$\Phi(z)=z^{n-2}p(\frac{1}{z})=b_{n-2}+b_{n-3}*z+...+b_0z^{n-2}$$

则

$$b_{n-2}=\lim_{z\to 0}\Phi(z).$$

代入

$$\frac{1}{1-z*z_j}=1+z*z_j+(z*z_j)^2+...$$

到

$$\Phi(z)=\frac{1}{z}\sum_{j=1}^{n}\alpha_j\prod_{l=1,l\neq j}^{n}(1-z*z_l)=\frac{1}{z}\prod_{l=1}^{n}(1-z*z_l)\sum_{j=1}^{n}\frac{\alpha_j}{1-z*z_j}$$

可得

$$\lim\Phi(z)=\lim\frac{1}{z}\sum_j\alpha_j\Big(1+z*z_j+(z*z_j)^2+...\Big)$$

进而

$$\lim\Phi(z)=b_{n-2}=0\Leftrightarrow\sum_j\alpha_j=0\quad\&\quad\sum_j\alpha_j*z_j=0$$

其它情形不逐一检验。建议研究生按照归纳法自己证明。 ■

下面考虑将定理的限制条件加入到有理逼近算法中得到$(m-1,n-1)$阶逼近。记$\mathrm{span}(A)$为矩阵A 的列空间，由

$$V_m\alpha=0\Leftrightarrow\alpha\in\mathsf{Null}(V_m)$$

和

$$\mathsf{Null}(V_m)\perp\mathrm{span}(V_m^\top),$$

得到$\alpha \perp \text{span}(V_m^\top)$. 为了保证$\alpha \perp \text{span}(V_m^\top)$, 可以对$V_m^\top$ 进行完全QR分解，即

$$V_m^\top = [Q_m \mid Q] * \begin{bmatrix} R_m \\ 0 \end{bmatrix}$$

则

$$\alpha \perp \text{span}(V_m^T) \Leftrightarrow \alpha \in \text{span}(Q).$$

进而 $\alpha = Q * \hat{\alpha}$. (注意到这里的记号$Q_m$是$n-m$维得，则$\alpha$是$m$维)

基于以上分析，记$C = 1/(x_j - z_k)$ 是Cauchy矩阵， 我们的目标是计算如下问题

$$n - f * d \approx 0 \;\; \Leftrightarrow \;\; C * Q * \hat{\alpha} - \text{Diag}(f) * C * \beta \approx 0$$

上述$\hat{\alpha}$, β可以由奇异值分解SVD 得到. 我们将算法给出

Algorithm 1 有理逼近的AAA-Lawson 算法

Input: 数据$\{(x_j, f_j)\}_{j=1}^M$; 有理式的阶数(m, n), &$m < n$

Output:

$$r = \sum_{k=1}^{n} \frac{\alpha_k}{x - z_k} \Big/ \sum_{k} \frac{\beta_k}{x - z_k} \quad \& \quad V_m \alpha = 0$$

1: AAA算法得到support points $\{z_k\}_{k=1}^n$
2: 记Vander matrix $V_m = \left[1, z, z^2, ..., z^{n-m-1}\right]$
3: qr 分解得到$Q \in \boldsymbol{R}^{n \times m}$
4: 记Cauchy矩阵C, 组装矩阵$\left[C * Q, \quad -\text{Diag}(f) * C\right]$
5: 计算最小奇异向量v, $\hat{\alpha} = v(1:m)$, $\beta = v(m+1:end)$.
6: $r = \left(C * Q * \hat{\alpha}\right) \Big/ \left(C * \beta\right)$.

标注 21 AAA-lawson算法迭代几次(maxI=1 OR 10)，尚未定论！ 权重bigS如何取未知！组装Cauchy矩阵时，取样本点时 丢掉support pts 处理起来简单，效果也好！

```
function [R, zj, wj, fj, zer_D, zer_N ] = rat_aaa_mn2(x1,f1,m,n,maxI)
if (m>n),  error('m>n,␣pls␣check!');  end
% get support pts by aaa
[R0, zj, wj, fj, mJ, J, err] = rat_aaa_nn(x1,f1,n);
%  zj : support pts
%  mJ : index of zj in x1
%  J  : index of x1 // zj
%  err: max error of aaa algrithm
Z = x1(J);  F = f1(J);   M = length(Z);   errAAA = err;  err0 = err;
% wj0 = wj; fj0 = fj;     % Save parameters in case Lawson fails
if err0 < 5e-11
    fprintf('AAA␣has␣error␣%.4e,␣lawson␣not␣used', err)
    R = R0;
```

```
    zer_N = prz(fj.*wj,zj);
    zer_D = prz(wj,zj);
    return
end
% % % % % % % % % % % % %
C = 1./(Z - zj.');
[Q, ~] = getbasis(zj,n,m);
A = [C*Q, -diag(F)*C];
A = [A; Q, -diag(fj)];  MM = length(x1);   FF = [F;fj];
wt_new = ones(MM,1);
 for k = 1:maxI
        wt = wt_new;
        W = spdiags(sqrt(wt),0,MM,MM);
        [~,~,V] = svd(W*A,0);
        c = V(:,end);     c = c./norm(c,"inf");
%
        alp = Q*c(1:m); bet = c(m+1:end);
        denom = C*bet; num = C*alp;
        R = num./denom;
        R(M+1:MM) = alp./bet;
        err = FF - R;    maxerr = norm(err,"inf");
        if maxerr < errAAA
           errAAA = maxerr;
           XI = 1;
        else
           XI = 0.5;
        end
        wt_new = wt.*abs(err).^XI; wt_new = wt_new / norm(wt_new,inf);
        if maxerr < (0.2*err0) & maxerr < 5e-11
            fprintf('iteration stop at k = %d',k);
            break,
        end
 end
    wj = bet; fj = alp./bet;
    R = @(zz) reval(zz, zj, fj, wj);
  zer_N = prz(fj.*wj,zj);
  zer_D = prz(wj,zj);
end
% % % % % % % % % % % % % % % % % % % % % % % % %
function [Q,V] = getbasis(Z,n,m)
% Z : support pts
% R(n,n) \to R(m,n)
```

```
if (m==n), Q = eye(n); V = [];  return , end
V = Z.^(0:n-m-1);
[Q,~] = qr(V);
Q = Q(:,n-m+1:end) ;
end
```

以下代码用于计算重心坐标形式的有理式的零点和pole

$$p_n(x) = \prod_{j=0}^{n}(x - x_j) * \sum_{j=0}^{n} \frac{w_j}{x - z_j}$$

```
%% % % % % % % % % % % % %    pols and zers
function pol = prz(w,z)
m = length(w); B = eye(m+1); B(1,1) = 0;
E = [0 w.'; ones(m,1) diag(z)];
pol = eig(E,B); pol = pol(~isinf(pol));
```

6.3 离散数据的最佳一致有理逼近

本节内容选自[29]. 文章理论较为深刻，有兴趣的读者可参考阅读。本节只从算法出发， 从容易理解的角度(非严谨数学数学推导) 给出算法和Cauchy法的关系.

我们首先回顾Cauchy法. 假设样本点$x = \{x_k\}_{k=1}^{M}$及数据f_k已知，则计算目标为

$$\min_{a,\,b} \sum_{k=1}^{M} \left[f_k - \frac{N(x_k)}{D(x_k)} \right]^2$$

对上式线性化

$$\min_{c=[a,\,b],\ \|c\|=1} \sum_{k=1}^{M} w_k \left[f(x_k) \cdot \sum_{j=0}^{n} b_j x_k^j - \sum_{j=0}^{m} a_j x_k^j \right]^2$$

其中$w_j \approx D(x_j)^{-1}$ 可视为权函数。 记

$$V_n = \left[1, x, ..., x^n\right], \quad F = \mathrm{diag}(f_j), \quad W = \mathrm{diag}(w_j)$$

则计算目标为寻找单位向量$c = [a; b]$, 使得

$$W \cdot \left[FV_n, \quad -V_m\right] \cdot \begin{bmatrix} b \\ a \end{bmatrix} = W \cdot A \cdot c \tag{10}$$

的2范数取最小值, 即 $\min_{\|c\|=1} \|WA \cdot c\|$. AAA- Lawson算法是基于上述Cauchy法框架设计的. 借助于Vandermonde with Arnoldi算法， (10) 可转化为如下形式

$$\min_{[\tilde{a};\tilde{b}]\in\mathbb{C}^{m+n+2}} \left[FQ_n, \quad -Q_m\right] \cdot \begin{bmatrix} \tilde{b} \\ \tilde{a} \end{bmatrix} \tag{11}$$

其中Q_m, Q_n都是列正交矩阵.

[29] LeiHong Zhang, et al., A convex dual programming for the rational minimax approximation and Lawson's iteration

张雷洪分析认为 (11) 和原最佳逼近问题 不等价. 借助于对偶理论，可将最佳一致有理逼近问题 转化为带约束的优化问题

$$\min_{\substack{[\tilde{a};\tilde{b}]\in\mathbb{C}^{m+n+2}\\ \|\sqrt{W}Q_n b\|_2=1}} \begin{bmatrix} FQ_n, & -Q_m \end{bmatrix} \cdot \begin{bmatrix} \tilde{b} \\ \tilde{a} \end{bmatrix} \tag{12}$$

再通过复杂的推导, 上述问题对应 的向量 $\tilde{b}$等价于计算

$$(Q_m^{\perp})^{\mathrm{H}} \cdot F \cdot Q_n$$

的最小右奇异向量.

标注 22 紧凑形式$(Q_m^{\perp})^{\mathrm{H}} \cdot F \cdot Q_n$可以看作是对(11) 左乘 $(Q_m^{\perp})^{\mathrm{H}}$ 得到. 当数据量$M \gg m$时，例如$M = 10000, m = 20$, 计算正交补$(Q_m^{\perp})$ 是耗时的. 此时，可等价计算

$$(Q_m^{\perp})(Q_m^{\perp})^{\mathrm{H}} \cdot F \cdot Q_n = (I_M - Q_m Q_m^{\mathrm{H}}) \cdot F \cdot Q_n$$

的右奇异向量. 此时计算上式的计算量为$\mathbb{O}(M^2n^2)$, 远远小于QR分解的计算量M^3.

例 22 设$k \geq m \geq n$, $A \in \mathbb{C}^{m\times n}$, $Q \in \mathbb{C}^{k\times m}$, $B = QA \in \mathbb{C}^{k\times n}$, 其中$Q$是列正交矩阵且列向量2模为1. 则

$$A_{m\times n}V = U_{m\times n}\Sigma \quad \Rightarrow \quad BV = QU_{m\times n}\Sigma$$

其中U, V分别为A的左右奇异向量. 上式表明B与A有相同的右奇异向量.

提示：由于Q, U 均是列正交向量，则$(QU)\cdot(QU)^{\mathrm{H}} = I$, 即$(QU)$也是列正交矩阵.