

用 Zernike 多项式进行波面拟合的几种算法*

鄢静舟¹ 雷 凡² 周必方¹ 高志强¹

(1 中国科学院南京天文仪器研制中心 南京 210042)

(2 北京市气象局 北京 100089)

摘 要 介绍了几种实现波面 Zernike 多项式拟合的常用算法。为避免因直接构造法方程组而引入计算误差, 这些方法归结为两种思路: 一是从基底函数系入手, 通过变换函数族的基底来改善法方程组状态; 二是直接从矛盾方程组入手, 应用 Householder 变换把系数矩阵正交三角化, 直接求解拟合系数。特别是第二种方法由本文第一次提出, 它避免了构造法方程组, 从而避免了以前的方法因构造的法方程组出现严重病态而引入的计算误差, 并且易于编程, 因而是一种比较理想的实现 Zernike 多项式拟合的算法。

关键词 波面拟合 Zernike 多项式 Gram-Schmidt 法 Householder 变换

中图分类号 TH703 **文献标识码** A

1 引 言

被测光学面形或光学系统的波面或表面总是光滑和连续的, 波面拟合就是选择一个线性无关的基底函数系的组合 $W(x, y)$ 来拟合离散波差函数 $w(x_i, y_i)$ ^[4], 由连续的 $W(x, y)$ 函数表征被测系统的波象差函数或面形。通常选用 Zernike 多项式^[1]进行波面拟合。

将被测波面用 n 项 Zernike 多项式表示为

$$\begin{aligned} w(x, y) &= q_1 Z_1(x, y) + q_2 Z_2(x, y) + \dots + q_n Z_n(x, y) \\ &= \mathbf{q}^T \mathbf{Z} \end{aligned} \quad (1)$$

现有 m 个离散测量数据点 $w_i(x_i, y_i)$, $i = 1, 2, \dots, m$

令 $a_{ij} = Z_j(x_i, y_i)$, $i = 1, 2, \dots, m; j = 1, 2, \dots, n_0$

* 本文受天文基金资助

收稿日期: 1999-03-03

修稿日期: 1999-07-05

代入(1)式得到矛盾方程组($m > n$)

$$\left. \begin{aligned} a_{11}q_1 + a_{12}q_2 + \dots + a_{1n}q_n &= w_1 \\ a_{21}q_1 + a_{22}q_2 + \dots + a_{2n}q_n &= w_2 \\ \dots\dots\dots \\ a_{m1}q_1 + a_{m2}q_2 + \dots + a_{mn}q_n &= w_m \end{aligned} \right\} \quad (2)$$

简记作 $Aq = W$ (2)

其中 $A = (a_{ij})$ 为 $m \times n$ 矩阵, $q = (q_1, q_2, \dots, q_n)^T$, $W = (w_1, w_2, \dots, w_m)^T$ 。

矛盾方程组(2)一般不存在通常意义下的解,即对任何 n 维向量 q , 一般 $W - Aq \neq 0$, 此时, 用最小二乘准则来求解参数 q^1, q^2, \dots, q^n , 从而导出线性方程组

$$\sum_{j=1}^n \left[\sum_{i=1}^m a_{ik}q_{ij} \right] q_j = \sum_{i=1}^m a_{ik}w_i, \quad (k = 1, 2, \dots, n) \quad (3)$$

(3)式即为求解最小二乘问题的法方程组(或正则方程组), 求解此方程组即可得到 $q_1 \sim q_n$ 。但在实际应用中, 特别是对于本文所处理的数据量较大的情况, 通过上面的方法直接构造的法方程组往往是严重病态的, 造成计算错误, 使拟合失败。为避免直接解法方程组, 我们可以有两类方法, 一是从基底函数系入手, 通过变换函数族的基底来改善法方程组的状态, 如 Gram-Schmidt 正交法和协方差矩阵法; 另一种方法是不用构造法方程组, 而直接从矛盾方程组 $Aq = W$ 入手, 应用 Householder 变换把系数矩阵 A 正交三角化, 从而得到精确求解。下面分别论述。

2 算法描述

2.1 Gram-Schmidt 正交化方法

在最小二乘问题求解中, 法方程组常常出现病态, 这是作为基底函数的 Zernike 多项式在离散采样点上不正交造成的。因此, 我们可以利用 Gram-Schmidt 正交化方法求出在波差数据点上离散正交的一组基底函数 V , 就可以避免系数矩阵出现病态, 从而准确求解系数向量 q 。

正交基底函数 V 可以用 Zernike 多项式 Z 的线性组合来表示, 即

$$V = DZ \quad (4)$$

上式中, D 是元素为 d_{ij} 的系数矩阵, V 中的各元素满足下列方程

$$\sigma V_r^1 V_r^2 = \begin{cases} 0 & r_1 \neq r_2 \\ 1 & r_1 = r_2 \end{cases} \quad (5)$$

式中, σ 为离散波差数据点的集合。

由 Gram-Schmidt 正交法给出 d_{ij} 如下:

$$d_{ij} = \begin{cases} 0 & i < j \\ \left[\frac{1}{\sigma} Z_i^2 - \sum_{r=1}^{i-1} \left(\frac{1}{\sigma} Z_i V_r \right)^2 \right]^{\frac{1}{2}} & i = j \\ - \sum_{r=1}^{i-1} d_{ir} d_{rj} \left[\frac{1}{\sigma} Z_i V_r \right] & i > j \end{cases} \quad (6)$$

由(1)和(4)式得

$$W(x, y) = \mathbf{q}^T \mathbf{Z} = \mathbf{q}^T \mathbf{D}^{-1} \mathbf{V} = \mathbf{B}^T \mathbf{V} \quad (7)$$

其中, $\mathbf{B}^T = \mathbf{q}^T \mathbf{D}^{-1}$, 于是有

$$\mathbf{q} = \mathbf{D}^T \mathbf{B} \quad \mathbf{B} = (B_1, B_2, \dots, B_n) \quad (8)$$

利用基底 V 的正交性, \mathbf{B} 可以由波面离散波差值用最小二乘法求出

$$B_i = \frac{w(x_i, y_i) V_i}{\sigma} \quad (9)$$

将此式代入(8)式就可得到系数 \mathbf{q} , 从而获得将已知离散数据点上波差值用多项式的线性组合表示的波面函数。

2.2 协方差矩阵法^[5-6]

这是一种简化的 Gram-Schmidt 法, 它避开了正交化过程, 通过多项式的协方差矩阵的线性变换来求解波差函数。这种方法可以大大减少计算量, 且便于程序的编制。

具体方法如下:

我们定义 A_{ij} 表示 Z_i 与 Z_j 的协方差, 令

$$A_{kl} = \frac{1}{n} \sum_{i=1}^m (Z_{ki} - \bar{Z}_k)(Z_{li} - \bar{Z}_l) = \frac{1}{n} \sum_{i=1}^m Z_{ki} Z_{li} - \bar{Z}_k \bar{Z}_l \quad (10)$$

其中 $\bar{Z}_j = \frac{1}{m} \sum_{i=1}^m Z_{ji}$ ($j = 1, 2, \dots, n$), 且 $A_{lk} = A_{kl}$

用 Gram-Schmidt 法进行 n 项 Zernike 多项式拟合时, 协方差矩阵的前 n 行所组成的矩阵

$$\begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} & A_{1,n+1} \\ A_{21} & A_{22} & \dots & A_{2n} & A_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nn} & A_{n,n+1} \end{pmatrix}$$

就是解线性方程组的增广矩阵。推导证明见文献[6]。

我们只要求解下列方程组

$$\begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix} = \begin{pmatrix} A_{1,n+1} \\ A_{2,n+1} \\ \vdots \\ A_{n,n+1} \end{pmatrix} \quad (11)$$

就可以得到所需的 Zernike 多项式的系数, 也就得到连续光滑的波像差函数 $W(x, y)$ 。

拟合的方差为^[6] $\sigma_n^2 = A_{n+1,n+1} - \sum_{j=1}^n q_j A_{n+1,j}$

2.3 Householder 变换法

法方程组(3)式可以写为

$$\mathbf{A}^T \mathbf{A} \mathbf{q} = \mathbf{A}^T \mathbf{W} \quad (12)$$

法方程组的条件数为 $K(\mathbf{A}^T \mathbf{A})$, 矛盾方程组的条件数为 $K(\mathbf{A})$, 两者关系为 $K(\mathbf{A}^T \mathbf{A}) = K^2(\mathbf{A})$, 而 $K(\mathbf{A}) \gg 1$ ^[2-3], 故法方程组的条件数要比矛盾方程组的条件数大, 法方程组的条件比矛盾方程组的条件要差, 这常常使解法方程组所引起的误差会比直接解矛盾方程舍入 \mathbf{A} 时引起的误差大

得多,更难捉摸的是实际计算出来的 $A^T A$ 有可能不是正定的,法方程组是严重病态的,造成计算错误,从而使求解失败。

因此我们可以考虑直接从矛盾方程组入手,用 Householder 变换对 A 进行正交三角化,直接求解最小二乘问题从而避免法方程组的构造,避免因构造的法方程组出现严重病态而引入的计算误差。下面叙述 Householder 变换及用它解最小二乘问题的正交化方法。

Householder 变换,又称镜像映射法或反射变换。

定义 1 Householder 矩阵是指形式为

$$H = I - 2uu^T \quad (13)$$

的矩阵,其中 u 为列矩阵, $u \in R^n$, $u^T u = 1$ 。

根据定义可以看出,Householder 矩阵 H 具有良好的性质:对称性($H^T = H$),正交性($H^T H = I$)和对合性($H^2 = I$)。

应用中经常是确定一个 Householder 矩阵 H 的 u 并不是单位向量,把它归化 $\frac{u}{\|u\|}$ 需要计算 $\|u\|$,这是比较费时的,为此,改用如下定理的方法。

定理 1 设 $u \in R^n$, 令 $\rho = \frac{1}{2} \|u\|^2$, 则 $H = I - \rho^{-1} uu^T$ 是一个 Householder 矩阵。

Householder 矩阵的一个关键性质是它和被用来把零元引进向量中,具体地说就是下列定理:

定理 2 设 $x \in R^n$, $\sigma = \pm \|x\|$, 且假定 $x_1 = -\sigma e_1$, 则可找到一个 Householder 矩阵 H , 使

$$Hx = -\sigma e_1 \quad (14)$$

其中 $e_1 = (1, 0, \dots, 0)^T$ 。

证明 作 $u = x + \sigma e_1$, 求出 $\rho = \frac{1}{2} \|u\|^2$, 则 $H = I - \rho^{-1} uu^T$ 即为所求的 Householder 矩阵。因为

$$\begin{aligned} \rho &= \frac{1}{2} (x + \sigma e_1)^T (x + \sigma e_1) = \frac{1}{2} (x^T x + 2\sigma x_1 + \sigma^2) \\ &= \sigma^2 + \sigma x_1 \end{aligned}$$

所以

$$\begin{aligned} Hx &= (I - \rho^{-1} uu^T)x = x - \frac{(x + \sigma e_1)(x + \sigma e_1)^T}{(\sigma^2 + \sigma x_1)} x \\ &= x - (x + \sigma e_1) = -\sigma e_1 \end{aligned}$$

上述定理的证明是构造性的,即它叙述了计算 u 与 ρ 的过程,一旦 σ 的正负号确定后,就可以求出

$$\begin{aligned} u_1 &= x_1 + \sigma \\ u_i &= x_i \quad (i = 2, \dots, n) \end{aligned}$$

以及 $\rho = \sigma(\sigma + x_1)$ 。

如果 σ 与 x_1 异号,在计算 u_1 时就会发生抵消,因此,取 $\sigma = \text{sign}(x_1) \|x\|$ 。在实际计算中,常常是在形成 u 后不再需要 x 的情况,为节省内存, u 的分量可以冲掉 x 的分量。

以上可归纳为下列算法:

算法一 已知 $x \in R^n$, 利用本算法可求出 σ, ρ, u , 使得 $Hx = (I - \rho^{-1}uu^T)x = -\sigma e_1$

$$\textcircled{1} \text{ 计算 } \sigma = \text{Sign}(x) \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$$

$$\textcircled{2} x_1 \quad u_1 = x_1 + \sigma$$

$$\textcircled{3} \rho = \sigma u_1$$

这个算法在个别情况下, 可能出现上溢或下溢而无法进行, 补救的办法是对 x 进行尺度变换。于是算法一修改成:

算法二 本算法中的 u 和 ρ 虽已改变, 但所得 Householder 矩阵 H 仍与算法一相同。

$$\textcircled{1} \text{ 计算 } \eta = \max_i |x_i|$$

$$\textcircled{2} x_i \quad u_i = x_i / \eta \quad (i = 1, 2, \dots, n)$$

$$\textcircled{3} \text{ 计算 } \sigma = \text{Sign}(u_1) \sqrt{u_1^2 + u_2^2 + \dots + u_n^2}$$

$$\textcircled{4} u_1 \quad u_1 + \sigma$$

$$\textcircled{5} \text{ 计算 } \rho = \sigma u_1$$

$$\textcircled{6} \sigma \quad \eta \sigma$$

现在描述矩阵的正交三角化过程。设 $A \in R^{m \times n}$, 把 A 的列向量记作 $a_j = (a_{1j}, a_{2j}, \dots, a_{mj})^T$, $j = 1, 2, \dots, n$ 。

第一步, 令 $A_1 = A = (a^1, a^2, \dots, a^n)$, 利用定理 2, 取 $x = a^1 = (a_{11}, a_{21}, \dots, a_{m1})^T$, 求出 $\sigma = \pm \|x\|$, $u = x + \sigma e_1$, $\rho_1 = \frac{1}{2} \|u\|^2$, $H_1 = I - \rho_1^{-1} u u^T$, $H_1 a^1 = -\sigma e_1$, 令 $\alpha_1 = -\sigma$, 于是

$$A_2 = H_1 A_1 = \begin{pmatrix} \alpha_1 & a_{12}^{(2)} & \dots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{m2}^{(2)} & \dots & a_{mn}^{(2)} \end{pmatrix}$$

第二步, 取 $x = (a_{22}^{(2)}, \dots, a_{m2}^{(2)})^T \in R^{m-1}$, 利用定理 2 构造 $H_2 = I_{m-1} - \rho_2^{-1} u u^T \in R^{(m-1) \times (m-1)}$, 从而得 $m \times n$ 矩阵, $H_2 = \begin{pmatrix} 1 & 0 \\ 0 & H_2 \end{pmatrix}$ 。

$$\text{于是 } A_3 = H_2 A_2 = \begin{pmatrix} \alpha_1 & a_{12}^{(2)} & \alpha_1^{(2)} & \dots & a_{1n}^{(2)} \\ 0 & \alpha_2 & a_{23}^{(3)} & \dots & a_{2n}^{(3)} \\ 0 & 0 & a_{33}^{(3)} & \dots & a_{3n}^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & a_{m3}^{(3)} & \dots & a_{mn}^{(3)} \end{pmatrix}, \text{ 一般 } A_k = H_{k-1} A_{k-1} = \begin{pmatrix} R_k & r_k & B_k \\ 0 & C_k & D_k \end{pmatrix}。 \text{ 如此继}$$

续, 直到 $r = \min(m-1, n)$ 步为止, 此时 $A_{r+1} = H_r H_{r-1} \dots H_2 H_1 A = \begin{cases} \text{上梯形矩阵} & n > m \\ \text{上三角矩阵} & n = m \end{cases}$ 。

关于存储, 可用 $u^k = (u_{k,k}, u_{k+1,k}, \dots, u_{m,k})^T$ 冲掉 a^k , 数 ρ_k 和对角元 α_k 可以存放在数组 A 的两个附加行上, 如下列形式所示:

$$\begin{pmatrix} u_{11} & x & x & \dots & x \\ u_{21} & u_{22} & x & \dots & x \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{m1} & u_{m2} & x & \dots & x \\ \rho_1 & \rho_2 & & & \\ \alpha_1 & \alpha_2 & & & \end{pmatrix}$$

在实际中,一旦 H 由算法二确定后,则计算 HA 的问题(其中 $A = (a^1, a^2, \dots, a^n)$ 为一已知矩阵),通过 $HA = (Ha^1, Ha^2, \dots, Ha^n)$, 就化为计算 Ha 的问题了。由

$$Ha = (I - \rho^{-1}uu^T)a = a - (\rho^{-1}uu^T)a \quad (15)$$

可以看出,实际上没有必要构造出矩阵 $H = I - \rho^{-1}uu^T$ 。

根据以上的讨论,我们写出矩阵的正交三角化的完整算法。

算法三 设 $A \in R^{m \times n}$ 和 $r = \min\{m-1, n\}$, 利用本算法求出 Householder 矩阵 H_1, H_2, \dots, H_r , 使 $A_{r+1} = H_r H_{r-1} \dots H_1 A$ 为上梯形阵或上三角阵, Householder 矩阵 $H_k = I - \rho_k^{-1}u_k u_k^T$ 中的向量 $u_k = (0, \dots, 0, u_{kk}, \dots, u_{mk})^T$, 非零元 u_{ik} 冲掉 a_{ik} , 数 ρ_k 放在 $a_{m+1,k}$ 的位置上, A_{r+1} 的非零元 u_{ik} (即 R_{r+1} 的元素) 除对角元 α_k 放在 $a_{m+2,k}$ 上外, 冲掉 A 中的相应元素。

对 $k = 1, 2, \dots, r$, 执行第①到第⑧步:

① 计算 $\eta = \max_{i=m}^1 \{ |a_{ik}| \}$;

② 如果 $\eta = 0$

i 置 $a_{m+1,k} = 0$

ii 重新开始对 k 的下一个值循环;

③ $a_{ik} \quad u_{ik} = \frac{a_{ik}}{\eta} \quad (i = k, k+1, \dots, m)$;

④ 计算 $\sigma = \text{Sign}(u_{kk}) \sqrt{u_{kk}^2 + \dots + u_{mk}^2}$;

⑤ 计算 $u_{kk} = u_{kk} + \sigma$;

⑥ $a_{m+1,k} \quad \rho_k = \sigma u_{kk}$;

⑦ $a_{m+2,k} \quad \alpha_k = -\eta\sigma$;

⑧ 对 $j = k+1, k+2, \dots, n$, 执行下列运算:

i 计算 $\tau = \rho_k^{-1} \sum_{i=k}^m u_{ik} a_{ij}$

ii $a_{ij} \quad a_{ij} - \tau u_{ik} \quad (i = k, k+1, \dots, m)$;

⑨ 如果 $m > n$, 则 $a_{m+2,m} \quad a_m = a_{m+2,m}$ 。

在第⑧步计算 $H_k D_k$ 时, 我们利用(8)式的结果, 并用 $H_k D_k$ 冲掉 D_k 。

我们这里的情况是 $m > n$, 此时算法大约需要 $mn^2 - \frac{1}{3}n^3$ 次乘法。这个算法十分稳定, 如果计算精确地进行, 算法将产生一个正交矩阵:

$$Q = H_r H_{r-1} \dots H_1 \quad (16)$$

使得 $A_{r+1} = QA$ 是上三角矩阵。

用 Householder 变换解最小二乘问题就是应用 Householder 变换把矛盾方程(2)式的系

数矩阵 A 正交三角化, 使

$$QA = \begin{pmatrix} R \\ O \end{pmatrix} \quad (17)$$

其中 R 为 n 阶上三角阵, O 为 $(m-n) \times n$ 的零矩阵, Q 是由 (9) 式得到的一个 m 阶正交矩阵。并把 m 维向量 QW 相应地分块成 n 维向量 C 与 $(m-n)$ 维向量 d , 即

$$QW = \begin{pmatrix} C \\ d \end{pmatrix} \quad (18)$$

于是

$$Q\delta = QW - QAq = \begin{pmatrix} C \\ d \end{pmatrix} - \begin{pmatrix} R \\ O \end{pmatrix} \cdot q = \begin{pmatrix} C - Rq \\ d \end{pmatrix} \quad (19)$$

因 Q 是正交矩阵, 所以

$$\delta^2 = Q\delta^2 = C - Rq^2 + d^2 \quad (20)$$

如果选择 q , 使得

$$C - Rq = O \quad (21)$$

那么 δ^2 将达到极小值, 此时, 由 (20) 式可得 $\delta^2 = d^2$, 而且从 (19) 式有 $Q\delta = \begin{pmatrix} O \\ d \end{pmatrix}$ 。

因此

$$\delta = Q^T \begin{pmatrix} O \\ d \end{pmatrix} \quad (22)$$

由 (21) 式可知, n 阶上三角形方程组

$$Rq = C$$

的解 q 就是最小二乘解, 它是非常容易求解的。

等式 (18)、(21) 和 (22) 实际上给出了一个用算法三的输出结果求解最小二乘问题的有效方法。这里, 可用形式 $H_1 H_2 \dots H_n W$ 来计算 QW , 而不用先计算 Q 。类似的, 也可在 (22) 式中用 Q^T 的分解来计算 δ 。

算法四 矩阵 A $R^{m \times n}$ 为列满秩, 数组 A 含有算法三的输出, 且 W R^m 。本算法求解使 $\delta = W - Aq$ 的范数达到极小的线性最小二乘问题, 达到极小时的残差 δ 冲掉 W 。

① 对 $i = 1, 2, \dots, n$ 执行 $W = H_i W$

② $C_i = w_i$ ($i = 1, 2, \dots, n$)

③ 解方程组 $Rq = C$

④ 置 $w_i = 0$ ($i = 1, 2, \dots, n$)

⑤ 对 $i = n, n-1, \dots, 1$ 执行

$$W = H_i W$$

3 计算实例

为考核上述算法的计算精度, 选择具有代表性的协方差法和本文提出的 Householder 变换法分别进行计算比较。一是对一套实际采集的波面数据进行拟合, 离散波面数据见表 1, 拟

合所得的多项式系数和拟合精度见表 2; 二是取一系列数学波面 $f(x, y)$, 对其进行离散采样, 得到波面的离散数据点, 对离散点进行拟合, 得到一用 Zernike 多项式表达的数学波面 $g(x, y)$, 令 $w(x, y) = f(x, y) - g(x, y)$, 根据 $w(x, y)$ 的 P-V 值和 RMS 值判断该算法的精度, 结果如表 3。

Table 1 Data from a real wavefront (fringe order and coordinates)

coordinate of the image center point ($XC = 965, YC = 1100$) radius $R = 500$
 the figure with shading represents the coordinate X ;
 the underlined figure is the fringe order.

Coordinate Y	- 4	- 3	- 2	- 1	0	1	2	3	4	5
1540					828	934	1025			
1460				761	856	945	1034	1134	1250	
1380			685	780	863	949	1030	1125	1226	
1300		595	706	790	868	946	1028	1117	1214	1339
1220		622	713	794	869	947	1026	1113	1208	1322
1140		634	725	798	874	954	1033	1121	1205	1327
1060	512	646	729	804	884	962	1041	1123	1214	1326
980	510	647	739	817	893	971	1049	1130	1220	1343
900		641	736	820	895	979	1055	1143	1239	
820		613	729	815	901	986	1070	1160	1278	
740			710	813	902	989	1081	1190		
660				813	913	1015	1132			

Table 2 The Zernike polynomial coefficients from two different algorithms

item	Housholder	covariance matrix	item	Householder	covariance matrix
1	5.427926E0	5.42763E0	20	2.817164E- 1	2.814276E- 1
2	1.408930E0	1.409159E0	21	2.654130E- 2	2.657814E- 2
3	- 1.810653E0	- 1.810711E0	22	5.615170E- 3	5.372822E- 3
4	2.755189E0	2.754634E0	23	3.570307E- 3	3.575357E- 3
5	2.937777E- 2	2.943707E- 2	24	- 2.661590E- 2	- 2.670163E- 2
6	- 6.527254E- 1	- 6.531707E- 1	25	- 8.517817E- 3	- 8.575439E- 3
7	8.377225E- 1	8.379200E- 1	26	2.391989E- 1	2.392499E- 1
8	9.195976E- 1	9.197510E- 1	27	- 2.036670E- 1	- 2.037312E- 1
9	7.432610E- 2	7.407740E- 2	28	3.165893E- 2	3.168977E- 2
10	- 8.335545E- 1	- 8.335499E- 1	29	- 3.307827E- 3	- 3.395677E- 3
11	1.572843E0	1.572351E0	30	- 2.387464E- 2	- 2.384520E- 2
12	8.704625E- 2	8.710876E- 2	31	- 3.681429E- 2	- 3.693016E- 2
13	- 1.115767E- 1	- 1.119794E- 1	32	1.373663E- 2	1.374274E- 2
14	3.105965E- 1	3.106627E- 1	33	- 2.508924E- 2	- 2.517268E- 2
15	- 2.359183E- 1	- 2.360796E- 1	34	1.330564E- 2	1.330002E- 2

Table 2(continue)

item	Householder	covariance matrix	item	Householder	covariance matrix
16	- 9.894806E- 1	- 9.896194E- 1	35	6.433435E- 3	6.414015E- 3
17	8.414636E- 4	8.927174E- 4	36	1.953641E- 3	1.953245E- 3
18	- 2.064192E- 3	- 2.259672E- 3	P-V	9.649459E- 2	9.650947E- 2
19	- 2.479020E- 1	- 2.478814E- 1	RMS	1.890423E- 2	1.890421E- 2

Table 3 Different mathematical descriptions of wavefront and the P-V, RMS value with different algorithms

mathematical descriptions	P-V		RMS	
	Householder	covariance matrix	Householder	covariance matrix
$(x^2 + y^2)^2$	1.693002E- 2	1.693332E- 2	3.017007E- 3	3.016843E- 3
$- 2x + 3xy^2 + 3x^2$	3.809254E- 2	3.809467E- 2	6.788266E- 3	6.787919E- 3
$y^2/4 - x^2/5$	6.348757E- 4	6.348407E- 4	1.131378E- 4	1.131315E- 4
$1 + 6y^2 - 6x^2 + 6y^4$ $+ 12x^2y^2 + 6x^3$	3.809254E- 2	3.809900E- 2	6.788266E- 3	6.787911E- 3
$5x$	1.265654E- 14	4.863739E- 5	2.076015E- 15	1.000861E- 5
$1 - 6y^2 - 6x^2 + 6y^3$ $+ 12x^2y^2 + 6x^3$ $+ x^4 + y^4$	6.348757E- 2	6.347907E- 2	1.131378E- 2	1.131324E- 2

4 结 论

上述的三种方法,前两种方法实际上都是基于 Gram-Schmidt 正交化方法,但它仍然离不开方程组的构造,只能在一定程度上改善法方程组的状态,而将 Householder 变换解线性最小二乘问题用于波面拟合则完全避免法方程组的构造,因而是一个非常稳定的方法,经过实际计算应用证明是可靠的。它的缺点是计算量较大,由算法三约化 A 大约需要 $\frac{2}{3}n^3$ 次乘法,而用 Gauss 消去法解法方程只需要 $\frac{1}{3}n^3$ 次乘法,但多开销的计算量对于现代计算机来说是可以接受的,也是值得的;算法的推导较复杂,但计算过程却相当简单,并且拟合方差的求解也比较简便。因而该方法不失为一种较理想的精确求解 Zernike 多项式系数的方法。

参 考 文 献

- 1 Malacara D. Optical Shop Testing. New York: John Wiley & sons Inc, 1978. 493
- 2 Thomas Kin J 著.数值计算引论.林成森等译.南京:南京大学出版社,1993
- 3 刘钦圣编著.最小二乘问题计算方法.北京:北京工业大学出版社,1989

- 4 Loomis John. Fringe User's Manual. Tucson: Version 2. Optical Science Center University of Arizona, 1976
- 5 刘月爱. 干涉图像的数字处理. 光学工程, 1985(4), 1 ~ 13
- 6 刘月爱. 条纹分析中一种简单的 Zernike 多项式拟合方法. 光学学报, 1985(5), 368 ~ 373

Algorithms for Wavefront Fitting Using Zernike Polynomial

YAN Jing-Zhou¹, LEI Fan², ZHOU Bi-Fang¹, GAO Zhi-Qiang¹

(¹ *Nanjing Astronomical Instruments Research Center,
Chinese Academy of Sciences, Nanjing 210042*)

(² *Beijing Meteorology Bureau, Beijing 100089*)

Abstract

Several algorithms for wavefront fitting using Zernike polynomial are studied. In order to avoid the computational error introduced by direct constructing normal equation group, the algorithms can be divided into two classes. One is Gram-Schmidt or covariance matrix method, in which we transform the base function set to meliorate the condition of the normal equation group. The other is called Householder transformation method, in which the matrix of inconsistent equation group is orthogonalized and triangulated using Householder transformation, and then the Zernike coefficients can be worked out by using a backsubstitution technique. It is proposed for the first time. Being a method that can avoid the computational error effectively and can be easily performed, it has proven to be a feasible and efficacious algorithm.

Key words: Wavefront fitting, Zernike polynomial, Gram-Schmidt orthogonalization method, Covariance matrix, Householder transformation

鄢静舟 男, 生于 1972 年 5 月 4 日。1994 年本科毕业于四川大学光电科学技术系。同年考入中科院南京天文仪器研制中心攻读硕士学位, 从事计算全息数字波面干涉仪的研究工作。1997 年至今在北京航天自动控制研究所从事图像处理及航天自动控制方面的工作。