XX IMEKO World Congress
Metrology for Green Growth
September 9−14, 2012, Busan, Republic of Korea

CHEBYSHEV FITTING OF COMPLEX SURFACES FOR PRECISION METROLOGY

Xiangchao Zhang[1], Min Xu[1], Hao Zhang[1], Xiaoying He[1], Xiangqian Jiang[1,2]

1. Shanghai Ultra-Precision Optical Manufacturing Engineering Centre, Fudan University, Shanghai, 200433, P. R. China
2. Centre for Precision Technologies, University of Huddersfield, Huddersfield, HD1 3DH, UK

Abstract: The form qualities of precision components are essential for their functionalities. The Peak-to-Valley parameters are widely adopted to assess the form accuracy of optical components. The commonly used least squares method is prone to over-estimation, thus the Chebyshev fitting should in turn be implemented. In this paper the original minimax optimization problem is converted into an unconstrained differentiable minimization problem by exponential penalty functions. The fitting accuracy and numerical stability are balanced by employing an active-set strategy and adjusting the configuration parameters adaptively. Finally some benchmark data sets are applied to demonstrate the validity and efficiency of this method.

Keywords: form error, minimum zone, minimax problem, exponential penalty function, Chebyshev fitting

1. INTRODUCTION

In precision engineering, the form qualities of precision components play an essential role in their functionalities, e.g. the transmission accuracy of gears, vibration of shafts, scanning uniformity of f-θ lenses, power efficiency of fan blades etc. Form errors are defined as the relative deviation between the measured data and the nominal shapes. The reference nominal surfaces are usually calculated in four approaches: least squares elements, minimum zone elements, maximum inscribed elements and minimum circumscribed elements [1].

Currently the Peak-to-Valley (PV) value is the most widely adopted parameter for the assessment of form accuracies of optical components, although the Root-of-Mean-Squared (RMS) parameter of the form deviation is more closely related to their optical functionalities. The form errors is evaluated by fitting a nominal function to the measured data set. In coordinate metrology, it is the required form error parameters that determine the objective functions of the fitting program. The Chebyshev fitting should be implemented to calculate the PV form error according to the numerical optimisation theory. However, the Chebyshev fitting is not continuously differentiable, thus very difficult to be solved. At present most researchers and commercial software in precision engineering implement least squares fitting due to its simplicity, and the difference between the maximal and minimal residuals are taken as the PV parameter. But the solution is not consistent with the definition of form errors in ISO standards [2] and the PV values will be seriously over-estimated, which will lead to unnecessary rejection of acceptable parts.

For simple geometries, like spheres, cylinders and cones, the tolerance zone is defined as the radial separation between two concentric elements. If considering symmetric tolerance only, it is equivalent to minimising the maximal orthogonal distance from the data points to the fitted surface,

$$\min_{\mathbf{x}} \max_{i \le j \le N} \left\| \mathbf{p}_j - \mathbf{q}_j \right\| \qquad (1)$$

where $\mathbf{q}_j$ is the projection point of a data point $\mathbf{p}_j$ onto the surface. $\mathbf{x} \in \mathbb{R}^n$ are the unknown motion parameters $\mathbf{m}$ (rotation angles and translation vector) and shape parameters $\mathbf{s}$. For the sake of computational simplicity, the transformation is always performed on the data points and the surface stays at the standard position.

For complex surfaces, the shapes of the two banding surfaces do not keep similar, as a consequence the geometric meaning of the tolerance zone is generalised as the covered space by a moving sphere whose centre travels on the nominal surface. Therefore the minimum zone fitting is equivalent to minimising the radius/diameter of the sphere and the resultant PV parameter is the sphere diameter *d*, as shown in Figure 1.

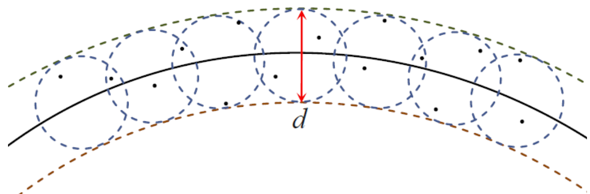

Figure 1. Minimum zone fitting of complex surfaces

Various methods have been proposed to calculate the MZ form errors of simple geometries, e.g. computational geometry methods [3], support vector machines [4], simplex methods [5] and so on. But these techniques have serious limitations in restricted applicability. They usually need to identify different cases and then appropriate manipulation will be implemented according to the specific shape or point distribution. Therefore they are very inconvenient to be applied in practice.

From the point view of numerical optimization, the MZ fitting is a discrete minimax problem. It can be approximated as a differentiable p-norm fitting problem. The solution approaches the Chebyshev fitting when $p\to\infty$ [6]. Subsequently the problem can be solved using the reweighted least squares technique. Some researchers turned it into a sequential linear programming [7] or sequential quadratic programming [8] problem, and the Gauss-Newton or quasi-Newton algorithm can be employed iteratively. These algorithms suffer from slow convergence when the solution approaches the region of the optimum point.

With the development of computer science and global optimization techniques, some heuristic optimisers have also been adopted for the form error evaluation of complex shapes, e.g. genetic algorithms [9], immune evolutionary algorithm [10], particle swarm optimization [11], differential evolution [12] etc. These algorithms are very powerful and in most cases can obtain global optimal solutions. But their computational complexity is of NP order and the results are not deterministic, which causes very large uncertainty in the solutions.

In this paper a fast and powerful method is presented to evaluate the MZ form errors of general complex surfaces. The original minimax optimization is transferred into an unconstrained differentiable minimization problem by exponential penalty functions.

2. CHEBYSHEV FITTING WITH EXPONENTIAL PENALTY FUNCTIONS

2.1 The exponential penalty function method

A discrete minimax problem

$$\min_{\mathbf{x}} F(\mathbf{x}) \text{ with } F(\mathbf{x}) = \max_{1\le j\le N} d_j(\mathbf{x}) \tag{2}$$

can be converted into a continuously differentiable problem using the exponential penalty function (also termed the aggregate function) [13]

$$\min_{\mathbf{x}} F_p(\mathbf{x}) \text{ with } F_p(\mathbf{x}) = \frac{1}{p}\log\sum_{j=}^{N}\exp\{pd_j(\mathbf{x})\} \tag{3}$$

From the numerical approximation theories it is known that

$$F(\mathbf{x}) \le F_p(\mathbf{x}) \le F(\mathbf{x}) + \frac{\log N}{p} \tag{4}$$

$F_p(\mathbf{x}) \to F(\mathbf{x})$ monotonically as $p\to\infty$. Therefore the approximation error can be quantitatively controlled by the smoothing parameter p. A larger p is preferred in terms of fitting accuracy. But the ill-conditioning problem will be caused, as a consequence proper trade-off need to be carefully made.

Now we investigate solving the differentiable optimisation problem in Equation (3) using the Newton's algorithm. Its gradient and Hessian matrices can be calculated as,

$$\nabla F_p(\mathbf{x}) = \sum_{j=1}^{N}\zeta_j \nabla d_j(\mathbf{x}) \tag{5}$$

$$\nabla^2 F_p(\mathbf{x}) = \sum_{j=1}^{N}\zeta_j \nabla^2 d_j(\mathbf{x}) + p\left(\sum_{j=1}^{N}\zeta_j \nabla d_j(\mathbf{x})\nabla d_j(\mathbf{x})^T - \sum_{j=1}^{N}\zeta_j \nabla d_j(\mathbf{x})\sum_{j=1}^{N}\zeta_j \nabla d_j(\mathbf{x})^T\right) \tag{6}$$

where $\zeta_j = \dfrac{\exp\{pd_j(\mathbf{x})\}}{\sum_{j=1}^{N}\exp\{pd_j(\mathbf{x})\}}$.

To improve the numerical stability and enlarge the convergence domain, the Hessian matrix is modified as [14],

$$\mathbf{B}_p(\mathbf{x}) = \nabla^2 F_p(\mathbf{x}) + \mu\mathbf{I}$$

where $\mu = \max\{0, \delta - e(\mathbf{x})\}$ with $e(\mathbf{x})$ the smallest eigenvalue of $\nabla^2 F_p(\mathbf{x})$, and δ a user-set parameter. In this way the Hessian matrix can always be guaranteed positive definite. As a result divergence of the solutions can be effectively avoided.

Then the solution can be iteratively updated as

$$\mathbf{x} \leftarrow \mathbf{x} - \mathbf{B}_p^{-1}\nabla^2 F_p(\mathbf{x})$$

During the optimization process, most of the computation effort is spent on calculating the gradient and Hessian matrices. To improve the efficiency, only a subset of data points are applied. It is obvious that when p is sufficiently large and $d_j(\mathbf{x}) < F(\mathbf{x})$, the term $\dfrac{\exp\{pd_j(\mathbf{x})\}}{\exp\{pF(\mathbf{x})\}}$ is approximately equal to zero, thus the term $\exp\{pd_j(\mathbf{x})\}$ has little contribution to $F_p(\mathbf{x})$ and can be ignored. An ϵ-active set is defined as

$$\Omega_p := \{j \mid F_p(x) - d_j(x) \le \varepsilon, 1 \le j \le N\}$$

Furthermore, the smoothing parameter p is adjusted adaptively during the optimization process. Combining the algorithms in [13] and [15], a simple and flexible adaptive active-set exponential penalty smoothing algorithm is presented as follows:

1. Initialise solution $\mathbf{x}^{(0)}$, configuration parameters $\alpha, \beta, \kappa \in (0,1)$, $p_0 >> 1$, $\varepsilon_0 > 0$, $\xi > 1$, $\varsigma > 1$, $i = j = 0$, and ϵ-active set $\Omega_0 = \Omega_{\varepsilon_0}(\mathbf{x}^{(0)})$.
//Here α and β control the step length of the solution
//incremental, κ controls numerical stability of matrix
//inversion, ξ is used to update the smoothing parameter
//p, and ς is used to update the parameter ϵ. i counts
//the iterates of solution incremental, and j counts the
//iterates of adjustment of p and ϵ.

2. Find descent direction. Compute $\mathbf{B}_{p_i,\Omega_i}(\mathbf{x}^{(i)})$ and its Cholesky factor $\mathbf{R}$ such that $\mathbf{B}_{p_i,\Omega_i}(\mathbf{x}^{(i)}) = \mathbf{R}\mathbf{R}^T$ and the reciprocal condition number $c(\mathbf{R})$.
If $c(\mathbf{R}) \geq \kappa$, the inverse of the Hessian matrix is stable, then

$$\mathbf{h}^{(i)} = -\mathbf{B}_{p_i,\Omega_i}(\mathbf{x}^{(i)})^{-1} \nabla F_{p_i,\Omega_i}(\mathbf{x}^{(i)}) \tag{7}$$

//the Newton's algorithm

Else

$$\mathbf{h}^{(i)} = -\nabla F_{p_i,\Omega_i}(\mathbf{x}^{(i)}) \tag{8}$$

//The steepest descent direction

Endif

3. Compute the step length by line search. Find the smallest integer k satisfying

$$F_{p_i,\Omega_i}(\mathbf{x}^{(i)} + \beta^k \mathbf{h}^{(i)}) - \nabla F_{p_i,\Omega_i}(\mathbf{x}^{(i)}) \leq -\alpha\beta^k \left\|\mathbf{h}^{(i)}\right\|^2$$

//make sure the objective function is decreased

and

$$F_{p_i,\Omega_i}(\mathbf{x}^{(i)} + \beta^k \mathbf{h}^{(i)}) - F(\mathbf{x}^{(i)} + \beta^k \mathbf{h}^{(i)}) \geq -10 \log N / p_i$$

//make sure the point reduction by ϵ-active set does not
//influence the solution seriously

4. Update the solution

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \beta^k \mathbf{h}^{(i)}$$

$$\Omega_{i+1} = \Omega_i \cup \Omega_{\varepsilon_i}(\mathbf{x}^{i+1})$$

5. Adjust smoothing parameter.
If $\left\|\nabla F_{p_i,\Omega_i}(\mathbf{x}^{(i+1)})\right\| \leq 2\varepsilon_i$, the solution stagnates for this configuration, then set

$$\mathbf{x}^*_{(j)} = \mathbf{x}^{(i+1)}$$

$$p_{i+1} = \xi p_i$$

$$\varepsilon_{i+1} = \varepsilon_i / \varsigma$$

//accept the solution as optimum for this configuration,
//enlarge smoothing parameter p and reduce point
//number in Ω_{ε_i}

$$i \leftarrow i+1$$

$$j \leftarrow j+1$$

Else set

$$p_{i+1} = p_i$$

$$\varepsilon_{i+1} = \varepsilon_i$$

$$i \leftarrow i+1$$

Endif

If termination criterion is satisfied
    Break
Else
    Go to **Step 2.**
Endif

2.2 Mathematical formulation of Chebyshev fitting

Without losing generality, the nominal geometric shape is represented in an implicit form as $f(\mathbf{q};\mathbf{s}) = 0$. Here $\mathbf{q} = [x, y, z]^T$ are the coordinates of an arbitrary point on the surface and $\mathbf{s} \in \mathbb{R}^q$ are the shape parameters (intrinsic characteristics) of the surface, e.g. the half axes lengths of an ellipsoid, half vertex angle of a cone etc.

The Chebyshev fitting problem in Equation (1) is equivalent to

$$\min_{\mathbf{x}} \max_{i \leq j \leq N} \left\|\mathbf{p}_j - \mathbf{q}_j\right\|^2$$

This formulae is applied in this paper hereafter. The calculation of the gradient and Hessian matrices is discussed in detail below.

In Equations (5) and (6), $d_j = \left\|\mathbf{p}_j - \mathbf{q}_j\right\|^2 = (\mathbf{p}_j - \mathbf{q}_j)^T (\mathbf{p}_j - \mathbf{q}_j)$, hence

$$\nabla d_j = 2(\mathbf{p}_j - \mathbf{q}_j)^T \frac{\partial(\mathbf{p}_j - \mathbf{q}_j)}{\partial \mathbf{x}} \tag{9}$$

$$\nabla^2 d_j = 2\left[\frac{\partial(\mathbf{p}_j-\mathbf{q}_j)}{\partial \mathbf{x}}\right]^T \frac{\partial(\mathbf{p}_j-\mathbf{q}_j)}{\partial \mathbf{x}} + 2(\mathbf{p}_j-\mathbf{q}_j)^T \frac{\partial^2(\mathbf{p}_j-\mathbf{q}_j)}{\partial \mathbf{x}^2} \tag{10}$$

Now we consider the term $(\mathbf{p}_j-\mathbf{q}_j)^T \frac{\partial(\mathbf{p}_j-\mathbf{q}_j)}{\partial \mathbf{x}}$. The matrix $\frac{\partial \mathbf{q}_j}{\partial \mathbf{x}}$, i.e. the dependency between the projection points and the motion/shape parameters, is very difficult to be calculated. Most researchers ignore it in their programs, but it is proved that the convergence rate will be seriously reduced [16]. We investigate calculating it in a tactful way. Since $f_j := f(\mathbf{q}_j; s) = 0$, thus its complete differentiation is,

$$\frac{\partial f_j}{\partial \mathbf{x}} = \frac{\partial f_j}{\partial \mathbf{q}_j}\frac{\partial \mathbf{q}_j}{\partial \mathbf{x}} + \frac{\partial f_j}{\partial \mathbf{x}} = 0$$

It is obvious that the vector $\frac{\partial f_j}{\partial \mathbf{q}_j}$ is along the normal vector of the surface, therefore it is parallel to $(\mathbf{p}_j-\mathbf{q}_j)$. This leads to

$$\begin{aligned}
&(\mathbf{p}_j-\mathbf{q}_j)^T \frac{\partial(\mathbf{p}_j-\mathbf{q}_j)}{\partial \mathbf{x}} \\
&= \frac{sign\left\langle \mathbf{p}_j-\mathbf{q}_j, \frac{\partial f_j}{\partial \mathbf{q}_j}\right\rangle \left\|\mathbf{p}_j-\mathbf{q}_j\right\|}{\left\|\frac{\partial f_j}{\partial \mathbf{q}_j}\right\|}\left(\frac{\partial f_j}{\partial \mathbf{q}_j}\frac{\partial \mathbf{p}_j}{\partial \mathbf{x}} - \frac{\partial f_j}{\partial \mathbf{q}_j}\frac{\partial \mathbf{q}_j}{\partial \mathbf{x}}\right) \\
&= \frac{sign\left\langle \mathbf{p}_j-\mathbf{q}_j, \frac{\partial f_j}{\partial \mathbf{q}_j}\right\rangle \left\|\mathbf{p}_j-\mathbf{q}_j\right\|}{\left\|\frac{\partial f_j}{\partial \mathbf{q}_j}\right\|}\left(\frac{\partial f_j}{\partial \mathbf{q}_j}\frac{\partial \mathbf{p}_j}{\partial \mathbf{x}} + \frac{\partial f_j}{\partial \mathbf{x}}\right)
\end{aligned}$$

As for the term $\frac{\partial^2(\mathbf{p}_j-\mathbf{q}_j)}{\partial \mathbf{x}^2}$, the variables $\mathbf{x}$ are separated into two groups $\mathbf{m}$ and $\mathbf{s}$. It is obvious that $\frac{\partial^2 \mathbf{p}_j}{\partial \mathbf{s}^2} = 0$ and $\frac{\partial^2 \mathbf{p}_j}{\partial \mathbf{s}\partial \mathbf{m}} = 0$. $\frac{\partial^2 \mathbf{p}_j}{\partial \mathbf{m}^2}$ is straightforward to be obtained, and $\frac{\partial^2 \mathbf{q}_j}{\partial \mathbf{m}^2}$ and $\frac{\partial^2 \mathbf{q}_j}{\partial \mathbf{m}\partial \mathbf{s}}$ are ignored in practice.

The flowchart of the Chebyshev fitting program is shown in Figure 2,

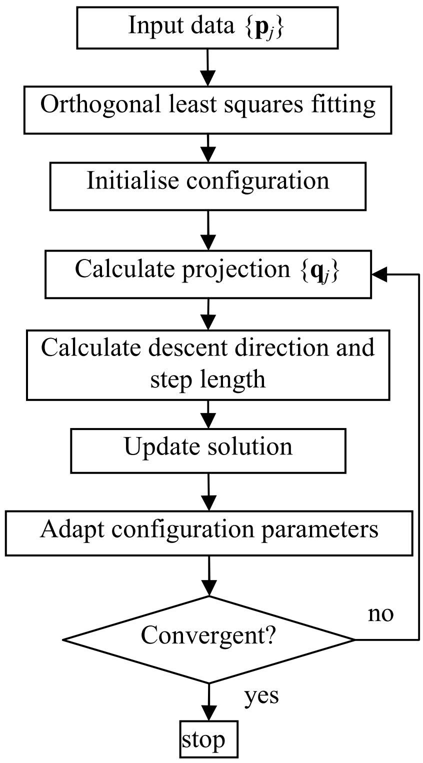

Figure 2. Flowchart of the fitting program

3. NUMERICAL VALIDATION

To demonstrate the validity of this algorithm, three examples were applied:

an ellipsoid $ax^2 + by^2 + cz^2 = 1$,

a hyperbolic paraboloid $ax^2 - bz^2 - y = 0$,

and an aspheric surface

$$z = \frac{r^2/R}{1+\sqrt{1-(1+k)r^2/R^2}} + a_4 r^4 + a_6 r^6$$

with $r = \sqrt{x^2 + y^2}$.

The same data points were adopted as in [12] and [17]. These three data sets have 56, 42 and 8100 data points respectively. Both the exponential penalty function (EPF) method and the differential evolution (DE) algorithm are applied for the Chebyshev fitting. The programs were coded in MATLAB and run on a PC with Intel Core (TM) -i5-2400 CPU 3.10GHz and 4.00 GB RAM in the Ubuntu Linux system. The obtained results are listed in Table 1.

Table 1. Comparison of DE algorithm with EPF method

data sets	DE results		EPF results	
	PV	time	PV	time
ellipsoid	0.063688 mm	56.17 s	0.063688 mm	1.86 s
hyper parab	0.017796 mm	38.56 s	0.017796 mm	1.52 s
asphere	3.21 μm	5.40 s	3.21 μm	2.34 s

Every run the DE program obtains different solutions, which causes very large uncertainty, and it is a inferiority of DE compared to EPF. In the table only the best results are recorded. The data set of the aspheric surface includes 8100 points. If running the DE optimisation program directly using all the data points, it will take more than two minutes. Therefore the alpha hull technique was employed first to reduce the point number before fitting.

In the table it can be seen that the EPF can always obtain the global optima, which are as good as DE. But the running time of EPF is much less than that of DE for ellipsoid and hyperbolic paraboloid fitting, and even less than aspheric surface fitting by DE with alpha-shape point reduction. If carrying out orthogonal distance least squares fitting direct to these three point sets, the obtained PV parameters will be 0.086813mm, 0.022467mm and 3.79μm respectively, which are 36.3%, 26.2% and 18.0% greater than the PV parameters obtained by the EPF Chebyshev fitting.

These examples effectively demonstrated the necessity and validity of Chebyshev fitting to obtain the correct PV form error parameters. Furthermore, they also revealed the superiority of EPF minimum zone fitting of complex shaped surfaces over the differential evolution algorithm in terms of solution uncertainty and computational efficiency.

4. CONCLLUSIONS

The Chebyshev fitting can obtain the correct PV parameter of the form error. The proposed exponential penalty function method is computationally efficient and can overcome the premature convergence problem, which shows great superiority over other smoothing approximation techniques and heuristic searching algorithms. The fitting accuracy and numerical stability are balanced by employing an active-set strategy and adjusting the precision parameters adaptively. It is especially competitive when the optimisation problem is of a large scale, i.e. the number of data points is up to several thousand.

5. REFERENCES

[1] D. Whitehouse. Surfaces and their measurement. London: Hermes Penton Science, 2002

[2] ISO 1101. Geometrical product specifications-geometrical tolerancing–tolerances of form, orientation, location and run-out; 2004

[3] N. Venkaiah and M. S. Shunmugam. “Evaluation of form data using computational geometric techniques-part II: cylindricity error”. Int. J. Mach. Tools & Manufact. 47:1237-1245, 2007

[4] A. M. Malyscheff, T. B. Trafalis and S. Raman. From support vector machine learning to the determination of the minimum enclosing zone. Comput. Ind. Eng. 42:59-72,2002

[5] T. Kanada. Evaluation of spherical for errors: computation of sphericity by means of minimum zone method and some examination with using simulated data. Proc. Eng. 17:281-289,1995

[6] Y. G. Shi. “Weighted simultaneous Chebyshev approximation”. J Approx. Theory. 32: 305-315, 1981

[7] I. Al-Subaihi and G. A. Watson. “Fitting parametric curves and surfaces by l∞ distance regression”. BIT Numer. Math. 45; 443-461, 2005

[8] F. Wang and Y. Wang. “Nonmonotone algorithm for minimax problems”. Appl. Math. Comput. 217: 6296-6308, 2011

[9] C. H. Liu, W. Y. Jywe and C. K. Chen. Quality assessment on a conical taper part based on the minimum zone definition using genetic algorithms. Int. J. Mach. Tools Manuf. 44:183-190,2004

[10] X. Wen and A Song. An immune evolutionary algorithm for sphericity error evaluation. Int. J. Mach. Tools. Manuf. 44:1077-1084, 2004

[11] Y. Kovvur, H. Ramaswami and R. B. Annad. Minimum-zone form tolerance evaluation using particle swarm optimization. Int. J. Intell. Syst. Technol. Appl. 4:79-96, 2008

[12] X. Zhang, X. Jiang and P. J. Scott. Minimum zone evaluation of the form errors of quadric surfaces. Proc. Eng. 35:383-389, 2011

[13] Y. Xiao and B. Yu. “A truncated aggregate smoothing Newton method for minimax problems”. App.l Math. Comput. Vol. 216: 1868-1879, 2010

[14] J. Nocedal, S. J. Wright, “Numerical optimization”, in: P. Glynn, S. M. Robinson (Eds.), Springer Series in Operations Research, Springer-Verlag, New York, 1999

[15] E. Y. Pee and J. O. Royset. “On solving large-scale finite minimax problems using exponential smoothing”. J Optim Theory Appl. 148: 390-421, 2011

[16] S. J. Ahn. Least squares orthogonal distance fitting of curve and surfaces in space. Springer, 2004

[17] X. Zhang, X. Jiang and P. J. Scott. “A minimax fitting algorithm for ultra-precision aspheric surfaces”. Proc 13th Intel. Conf. Metrology and Props of Eng. Surf. pp. 285-289, 2011