

Algorithms and software for areal surface texture function parameters

I M Smith¹, P M Harris¹, R K Leach², C Giusca³, X Jiang⁴,
P Scott⁴ and L D Todhunter²

¹National Physical Laboratory, Teddington TW11 0LW, UK

²Manufacturing Metrology Team, Faculty of Engineering, University of Nottingham, Nottingham NG7 2RD, UK

³Precision Engineering Institute, Cranfield University, MK43 0AL, UK

⁴Centre for Precision Technologies, School of Computing and Engineering, University of Huddersfield, Huddersfield HD1 3DH, UK

E-mail: ian.smith@npl.co.uk

Abstract. Software for the evaluation of areal surface texture function parameters is described. Definitions of the parameters, expressed in terms of the inverse areal material ratio function, are provided along with details of the numerical algorithms employed in the software to implement calculations to evaluate approximations to the parameters according to those definitions. Results obtained using the software to process a number of data sets representing different surfaces are compared with those returned by proprietary software for surface texture measurement. Differences in the results, arising from different choices being made when implementing the steps in the parameter evaluation process, are discussed.

Keywords: surface texture, areal function parameters, parameter definitions, software comparison

Submitted to: *Meas. Sci. Technol.*

1. Introduction

Areal surface texture parameters provide a means to characterize quantitatively the topography of a surface and predict its functional performance. Since the use of surface texture parameters was first proposed [1], there has been significant related activity in standardization. For example, ISO 25178-2:2012 [2] provides definitions for a number of areal surface texture parameters. These parameters have been, and continue to be, adopted by industry and incorporated into measuring instruments and software.

In previous work [3], software measurement standards [4] were developed and disseminated for testing software for evaluating profile parameters [5]. Software measurement standards have also been developed by the National Institute of Standards and Technology (NIST, US) for profile and areal surface texture parameters [6, 7], and by the Physikalisch-Technische Bundesanstalt (PTB, Germany) for profile parameters [8]. More recently [9], reference software was described for the evaluation of a subset of areal surface texture parameters that are classified either as ‘height’ parameters, ‘spatial’ parameters or ‘hybrid’ parameters. For those parameters, a comparison of results obtained using the reference software and a number of proprietary software packages was described [10].

This paper is concerned with a subset of areal surface texture parameters that are classified as ‘function’ parameters, and describes software [11] that has been developed at the National Physical Laboratory (NPL, UK) to evaluate approximations to those parameters[‡]. The paper focusses primarily on the definitions of the parameters considered and the numerical algorithms employed in the software to implement calculations to evaluate approximations to the parameters according to those definitions. In developing the software various choices have been made, but these choices are believed to be in the spirit of the relevant standards, and are made explicit in order to make clear the approach implemented to evaluate approximations to the parameters.

The aim of the work described, undertaken jointly by NPL and the universities of Nottingham and Huddersfield, is to use a comparison for four test data sets of the results returned by different software for evaluating surface texture function parameters, including NPL’s software and various proprietary software packages, to highlight possible differences in the algorithms implemented by the software. In this regard, the algorithms implemented by NPL’s software are transparent whereas those implemented by the other software are not. The intention of the work reported here is to highlight that differences between the results returned by software can arise, and to encourage harmonization of the ways to evaluate the parameters in order to reduce those differences.

The paper is organized as follows. Section 2 provides an overview of the main functions of the software and indicates any conditions that are assumed to apply. The

[‡] In addition to evaluating function parameters, the software also evaluates areal surface texture parameters that are classified as ‘feature’ parameters. Algorithms and software for these feature parameters will be described in a future publication.

algorithms implementing the main functions of the software are described in sections 3 to 5, focussing on, respectively, the surface representation used, the evaluation of areas in terms of that surface representation, and the evaluation of parameters in terms of those calculated areas. The comparison methodology is described in section 6 while section 7 presents and discusses the results of the comparison. A summary is given in section 8.

2. Overview

The main functions of the software are as follows:

- Read data defining an *S–F surface* [2, clause 3.1.5], i.e., a surface that has been S–filtered (to remove small scale components) and from which form has been removed (using an F–operator);
- Apply a Gaussian areal filter [12] to the data defining an S–F surface to obtain data defining the corresponding *S–L surface* [2, clause 3.1.6], i.e., a surface derived from the S–F surface by removing large scale components using an L–filter [9, section 3];
- Remove points from the data defining an S–L surface, i.e., ‘trim’ the data, to produce a surface defined on an area which has sides with lengths that are integer multiples of the cut-off wavelength (nesting index) for the Gaussian areal filter used to obtain the surface [9, section 4];
- Apply bilinear interpolation to obtain representations of the S–F and trimmed S–L surfaces (section 3);
- Evaluate the areas of approximations to regions over which the S–F and trimmed S–L surfaces have heights greater than or equal to a specified value (section 4);
- Evaluate approximations to the inverse areal material ratio functions for the S–F and trimmed S–L surfaces in the form of piecewise cubic interpolants to data defining those functions (sections 5.1–5.2);
- Evaluate approximations to function parameters in terms of those representations of the inverse areal material ratio functions (sections 5.3–5.15) [12];
- Write the values of the areal surface texture parameters to an output file.

The areal surface texture function parameters to be evaluated are S_k (core height), S_{mr1} (peaks material ratio), S_{mr2} (dales material ratio), S_{pk} (reduced peak height), S_{vk} (reduced dale height), S_{xp} (peak extreme height), V_{mp} (peak material volume), V_{mc} (core material volume), V_{vv} (dale void volume) and V_{vc} (core void volume).

The parameters S_k , S_{mr1} , S_{mr2} , S_{pk} and S_{vk} can be considered as the areal analogues of parameters for profiles. The volume parameters V_{mp} , V_{mc} , V_{vv} and V_{vc} provide

§ For clarity, and to ensure the notation is mathematically unambiguous, the notation S_k , etc., is used throughout to denote areal surface texture parameters in place of the (more familiar) notation Sk , etc., used in specification standards.

information, for example, on the lubrication performance of a surface, the amount of wear, and the ability of the surface to retain fluid.

For the representation of the surface and the specification of areal surface texture parameters, a rectangular Cartesian set of axes is assumed [2, clause 3.1.2]. Specifically, if the nominal surface is a plane (or portion of a plane), the x - and y -axes are taken to lie in the surface with the z -axis orthogonal to the surface in an outward direction, i.e., from the material to the surrounding medium. The S-F surface data is defined by uniform sampling intervals $\Delta_x > 0$ and $\Delta_y > 0$ in, respectively, the x - and y -directions with, in general, $\Delta_x \neq \Delta_y$, and a matrix \mathbf{Z} of dimension $m_x \times m_y$ containing the surface heights $z_{i,j}$, $i = 1, \dots, m_x$, $j = 1, \dots, m_y$. The value $z_{i,j}$ is considered to correspond to the height of a surface $z(x, y)$ at the point (x_i, y_j) with $x_i = (i-1)\Delta_x$ and $y_j = (j-1)\Delta_y$, i.e., $z_{1,1}$ is the height at the origin of the xy -coordinate system and z_{m_x, m_y} at the point ‘diagonally opposed’ to the origin.

The following conditions are assumed to apply:

- (A1) The input data is provided in X3P format [13], a format for defining surface data that allows the smooth interchange of data between different measuring systems. Software implementing the data format has been developed by the ‘openGPS’ consortium as freeware [14];
- (A2) For the S-F surface, the *evaluation area* [2, clause 3.1.10], i.e., the area in the xy -plane used to specify the portion of the surface under evaluation, is the rectangle $[x_1, x_{m_x}] \times [y_1, y_{m_y}]$ with sides of lengths $L_x = (m_x - 1)\Delta_x$ in the x -direction and $L_y = (m_y - 1)\Delta_y$ in the y -direction containing the measured surface data||;
- (A3) For Gaussian areal filtering, the same cut-off wavelength $\lambda > 0$ is applied in the x - and y -directions, with λ an integer multiple of the sampling intervals Δ_x and Δ_y ;
- (A4) The lengths L_x and L_y are at least three times the cut-off wavelength in order to minimize any distortion of the filtered surface due to the finite extent of the surface;
- (A5) There are at least fifty points per cut-off wavelength in each direction in order to minimize any distortion of the filtered surface due to the finite number of data points per cut-off wavelength;
- (A6) For the S-L surface, the evaluation area is a rectangle of largest area with sides having lengths equal to integer multiples of λ that is contained within the rectangle obtained from the evaluation area for the corresponding S-F surface by removing a border of width λ . The evaluation area for the S-L surface is generally not uniquely defined. A particular choice is made by requiring that it has approximately the same centroid as the evaluation area for the corresponding S-F surface: see [9, section 4].

|| The term ‘evaluation area’ is used to refer both to the rectangular portion of the surface under evaluation and the area of that portion, i.e., the product of its length and width. When the term is used in this paper, its meaning should be clear from the context of its use.

3. Surface representation

Let $f(x, y)$ denote the S–F surface $z(x, y)$ or the trimmed S–L surface $r(x, y)$ with heights $f_{i,j} \equiv f(x_i, y_j)$ at points (x_i, y_j) . The evaluation area A for the surface $f(x, y)$ is expressed as the union of rectangular regions $A_{i,j}$, defined by the locations (x_i, y_j) of the heights for the surface:

$$A = \bigcup_{i,j} A_{i,j},$$

where

$$A_{i,j} = \{(x, y) : x_i \leq x \leq x_{i+1}, y_j \leq y \leq y_{j+1}\},$$

and $A_{i,j}$ has area $|A_{i,j}| = \Delta_x \Delta_y$. On each rectangular region $A_{i,j}$, the function f is defined to be the bilinear surface that interpolates the surface heights $f_{i,j}$, $f_{i,j+1}$, $f_{i+1,j+1}$ and $f_{i+1,j}$ at the vertices of the region. In terms of normalized variables

$$X = \frac{x - x_i}{x_{i+1} - x_i} = \frac{x - x_i}{\Delta_x}, \quad Y = \frac{y - y_j}{y_{j+1} - y_j} = \frac{y - y_j}{\Delta_y},$$

the function f can be written as

$$f(X, Y) = (1 - X)(1 - Y)f_{i,j} + X(1 - Y)f_{i+1,j} + (1 - X)Yf_{i,j+1} + XYf_{i+1,j+1}, \quad (1)$$

and is termed bilinear since for fixed Y , f is linear in X and for fixed X , f is linear in Y . A consequence of this choice of surface interpolant is that

$$\min_{(x,y) \in A_{i,j}} f(x, y) = \min\{f_{i,j}, f_{i,j+1}, f_{i+1,j+1}, f_{i+1,j}\},$$

and

$$\max_{(x,y) \in A_{i,j}} f(x, y) = \max\{f_{i,j}, f_{i,j+1}, f_{i+1,j+1}, f_{i+1,j}\}.$$

4. Area evaluation

It is required to evaluate the area $|A(c)|$ of the region $A(c)$ corresponding to that part of an S–F or trimmed S–L surface with height greater than or equal to a specified value c , i.e.,

$$|A(c)| = \iint_{(x,y) \in A(c)} dx dy = \sum_{i,j} \iint_{(x,y) \in A_{i,j}(c)} dx dy, \quad (2)$$

where

$$A(c) = \{(x, y) \in A : f(x, y) \geq c\}, \quad A_{i,j}(c) = \{(x, y) \in A_{i,j} : f(x, y) \geq c\}.$$

Let the rectangular regions $A_{i,j}$ be classified into three groups:

- (i) G_1 : those with indices (i, j) for which

$$\min_{(x,y) \in A_{i,j}} f(x, y) \geq c;$$

(ii) G_2 : those with indices (i, j) for which

$$\max_{(x,y) \in A_{i,j}} f(x, y) \leq c;$$

(iii) G_3 : those with indices (i, j) for which

$$\min_{(x,y) \in A_{i,j}} f(x, y) < c < \max_{(x,y) \in A_{i,j}} f(x, y).$$

It follows that a lower bound for $|A(c)|$ is

$$|A_L^0(c)| = n_0 \Delta_x \Delta_y,$$

and an upper bound is

$$|A_U^0(c)| = |A| - m_0 \Delta_x \Delta_y,$$

where $|A|$ is the evaluation area and n_0 and m_0 are, respectively, the numbers of regions in the groups G_1 and G_2 .

The indices in G_3 identify those regions which are ‘cut’ by the boundary of $A(c)$ corresponding to the contour of the surface at height c . The areas of these regions are counted in the evaluation of the upper bound for $|A(c)|$ but not in the evaluation of the lower bound. For each such region $A_{i,j}$ in G_3 , consider dividing the region into n_1^2 rectangular sub-regions defined by uniform sub-divisions of the intervals $X \in [0, 1]$ and $Y \in [0, 1]$ each into n_1 sub-intervals of length $1/n_1$, and using the form (1) to evaluate the function f at the vertices of those sub-regions. Then, as above, the sub-regions can be classified into groups according to whether the minimum value of the function f over the sub-region is greater than or equal to c , the maximum value is less than or equal to c , or c lies strictly between the minimum and maximum values, with the numbers of sub-regions in the first two groups defined by, respectively, $n_{i,j}$ and $m_{i,j}$. It follows that corrected values for the lower and upper bounds for $|A(c)|$ are, respectively,

$$|A_L(c)| = |A_L^0(c)| + \frac{\Delta_x \Delta_y}{n_1^2} \sum_{(i,j) \in G_3} n_{i,j}, \quad (3)$$

and

$$|A_U(c)| = |A_U^0(c)| - \frac{\Delta_x \Delta_y}{n_1^2} \sum_{(i,j) \in G_3} m_{i,j}, \quad (4)$$

and the area of an approximation to the region $A(c)$ is given by

$$|\hat{A}(c)| = \frac{1}{2} (|A_L(c)| + |A_U(c)|).$$

The area calculated in this way constitutes an approximation to the ‘true’ area for the piecewise bilinear surface $f(x, y)$, $(x, y) \in A$, defined by expression (1). The quality of the approximation is controlled by the number n_1 of intervals used in the subdivision of each region $A_{i,j}$. One measure of the quality of the approximation is the

absolute deviation between the lower and upper bounds for $|A(c)|$, viz., $|A_U(c)| - |A_L(c)|$. The number n_I of intervals also controls the cost, in terms of processing time, of the calculation of the approximation $|\widehat{A}(c)|$.

Note that the approximate approach described above is akin to applying Monte Carlo integration [15] to evaluate the integral (2), except that the sampling is carried out deterministically according to the behaviour of the function $f(x, y)$ rather than randomly, and areas are counted rather than samples. Furthermore, as an alternative to increasing n_I to reduce the approximation error, the approach can be applied recursively (with a fixed value of n_I) by calculating corrections (as in expressions (3) and (4)) at one level in terms of the sub-regions identified at the previous level as being cut by the contours at height c .

Note also that for the piecewise bilinear surface $f(x, y)$, $(x, y) \in A$, defined by expression (1), it is possible to evaluate analytically the area of a region enclosed by a contour to the surface at height c . The application of the analytical approach is discussed in detail in Appendix A [provided as supplementary data]. Compared to the analytical approach, the approximate approach has several advantages: it is independent of the choice of surface interpolant implemented (and is therefore quite general), it is straightforward to implement, special cases (e.g., relating to particular values of the surface heights at the vertices) do not have to be identified and treated separately, and it is less likely to encounter numerical issues in software. For some choices of surface interpolant, an analytical approach may simply not be available.

5. Function parameter evaluation

5.1. Areal material ratio (function)

The areal material ratio $S_{\text{mr}}(c)$ [2, clause 4.4.1, 4.4.2] of a scale limited surface is defined as the ratio of the area of the material at a specified height c to the evaluation area. $S_{\text{mr}}(c)$ is usually expressed as a percentage and is therefore given by

$$S_{\text{mr}}(c) = 100 \frac{|A(c)|}{|A|} \text{ \%}.$$

The evaluation of the area $|\widehat{A}(c)|$ of an approximation to the region $A(c)$ is described in section 4, and an approximation to the areal material ratio at height c is then given by

$$\widehat{S}_{\text{mr}}(c) = 100 \frac{|\widehat{A}(c)|}{|A|} \text{ \%}.$$

The function $S_{\text{mr}}(c)$ can be approximated by $\widehat{S}_{\text{mr}}(c)$, and a measure of the quality of this approximation is given by

$$Q = \max_c Q(c),$$

where

$$Q(c) = 100 \frac{|A_U(c)| - |A_L(c)|}{|A|} \text{ \%}.$$

An approximation \widehat{Q} to Q is given by evaluating $Q(c)$ at a discrete set of height values c_k , $k = 1, \dots, n_c$, viz.,

$$\widehat{Q} = \max_{k=1, \dots, n_c} \{Q(c_k)\}.$$

Figure 1 shows as a colour map the S–F surface of test data set 1 (section 6). Figure 2 shows \widehat{Q} , calculated for $n_c = 101$ uniformly-spaced heights between the minimum and maximum heights, plotted against n_I for test data set 1. The higher the value of n_I , the better is the quality of the approximation, i.e., as n_I increases, the approximation approaches the ‘true’ value. Figure 3 shows the approximations to the values of the areal material ratio function, calculated for $n_c = 101$ and $n_I = 101$, plotted against height for test data set 1.

5.2. Inverse areal material ratio (function)

The areal material ratio function $S_{mr}(c)$ [2, clause 4.4.3] is a monotonic function and therefore its inverse, referred to as the inverse areal material ratio function, exists. The inverse areal material ratio $S_{mc}(p)$ of a scale limited surface is defined as the height at which the ratio of the area of the material at that height to the evaluation area is p .

An approximation $c = \widehat{S}_{mc}(p)$ to the inverse areal material ratio function is constructed as a shape-preserving piecewise cubic polynomial interpolant to the data points $(p_k \equiv \widehat{S}_{mr,k}, c_k)$, $k = 1, \dots, n_c$, obtained from the evaluation of the approximation $\widehat{S}_{mr}(c)$ to the areal material function. MATLAB functions `griddedInterpolant` or `interp1`, with the option `pchip`, can be used for this purpose. The use of a shape-preserving interpolant is important, particularly when n_c is small, to ensure the approximating curve to the inverse areal material function is monotonic.

Figure 4 shows the approximation to the inverse areal material ratio function (height), calculated at 1 001 uniformly-spaced points, plotted against areal material ratio for test data set 1.

5.3. Equivalent straight line for inverse areal material ratio function

The equivalent straight line [2, clause 5.2] is required for the calculation of areal function parameters core height, peaks material ratio, dales material ratio, reduced peak height and reduced dale height. It is identified by the areal material ratio p_E such that the (absolute) gradient of the secant joining the points $(p_E, S_{mc}(p_E))$ and $(p_E + 40 \%, S_{mc}(p_E + 40 \%))$ is the smallest of all the (absolute) gradients of the secants joining the points $(p, S_{mc}(p))$ and $(p + 40 \%, S_{mc}(p + 40 \%))$, $0 \% \leq p \leq 60 \%$. The equivalent straight line is then defined to be the secant extrapolated to the interval $[0 \%, 100 \%]$.

Given an approximation $c = \widehat{S}_{mc}(p)$ to the inverse areal material ratio function in the form of a shape-preserving piecewise cubic function, an approximation to the equivalent straight line is determined as follows:

- (i) Define areal material ratios p_l , $l = 1, \dots, n_p$, and q_l , $l = 1, \dots, n_p$, where

$$p_l = \left(\frac{l-1}{n_p-1} \right) 60 \%, \quad q_l = p_l + 40 \%.$$

- (ii) Of all possible secants joining points $(p_l, \widehat{S}_{\text{mc}}(p_l))$, and $(q_l, \widehat{S}_{\text{mc}}(q_l))$, determine the secant that has the smallest (absolute) gradient. (If two or more secants have the same gradient, choose the secant for which the value of l is smallest.)
- (iii) The approximation to the equivalent straight line is then defined to be the secant identified in step (ii) extrapolated to the interval $[0 \%, 100 \%]$.

The quality of the approximation to the equivalent straight line is controlled by the number n_p of points used to define areal material ratios p_l in step (i).

Note that an approximation to the equivalent straight line may be determined from the approximation $\widehat{S}_{\text{mr}}(c)$ to the areal material ratio function without forming explicitly an approximation $\widehat{S}_{\text{mc}}(p)$ to its inverse, e.g., using an iterative algorithm to find $p_l = \widehat{S}_{\text{mr}}(c_l)$ and $q_l = p_l + 40 \% = \widehat{S}_{\text{mr}}(d_l)$ such that the (absolute) gradient of the secant joining the points (p_l, c_l) and (q_l, d_l) is a minimum. However, not all the areal surface texture parameters can be calculated directly from $\widehat{S}_{\text{mr}}(c)$, and so the calculations of an approximation to the equivalent straight line are undertaken in terms of $\widehat{S}_{\text{mc}}(p)$.

Let the secant determined in step (iii) be identified by areal material ratios p_L and q_L where $q_L = p_L + 40 \%$. The approximation to the equivalent straight line has gradient $(\widehat{S}_{\text{mc}}(q_L) - \widehat{S}_{\text{mc}}(p_L))/40 \%$, passes through the point $(p_L, \widehat{S}_{\text{mc}}(p_L))$, and is described by the equation

$$\widehat{S}_{\text{esl}}(p) = \widehat{S}_{\text{mc}}(p_L) + \frac{(p - p_L)(\widehat{S}_{\text{mc}}(q_L) - \widehat{S}_{\text{mc}}(p_L))}{40 \%}. \quad (5)$$

Figure 5 shows the absolute gradients of the secants, calculated for $n_p = 1001$, for test data set 1. The approximation to the equivalent straight line is shown in figure 4.

5.4. Core height

The value $S_{\text{mc,H}}$ of the equivalent straight line (5) that corresponds to $p = 0 \%$ identifies the highest level of the core surface, and an approximation to $S_{\text{mc,H}}$ is given by

$$\widehat{S}_{\text{mc,H}} \equiv \widehat{S}_{\text{esl}}(0 \%) = \widehat{S}_{\text{mc}}(p_L) - \frac{p_L(\widehat{S}_{\text{mc}}(q_L) - \widehat{S}_{\text{mc}}(p_L))}{40 \%}.$$

Similarly, the value $S_{\text{mc,L}}$ of the equivalent straight line (5) that corresponds to $p = 100 \%$ identifies the lowest level of the core surface, and an approximation to $S_{\text{mc,L}}$ is given by

$$\widehat{S}_{\text{mc,L}} \equiv \widehat{S}_{\text{esl}}(100 \%) = \widehat{S}_{\text{mc}}(p_L) + \frac{(100 \% - p_L)(\widehat{S}_{\text{mc}}(q_L) - \widehat{S}_{\text{mc}}(p_L))}{40 \%}.$$

The function parameter S_k [2, clause 4.4.4.2] is defined as the distance between the highest and lowest levels of the core surface, and an approximation to S_k is given by

$$\widehat{S}_k = \widehat{S}_{\text{mc,H}} - \widehat{S}_{\text{mc,L}} = \frac{5(\widehat{S}_{\text{mc}}(p_L) - \widehat{S}_{\text{mc}}(q_L))}{2}.$$

Figure 6 shows the relationships of S_k to the approximations of the inverse areal material ratio function for test data set 1.

5.5. Peaks material ratio

The function parameter S_{mr1} [2, clause 4.4.4.5] is defined as the ratio of the area of the material at the intersection line which separates the protruding hills from the core surface to the evaluation area. It is determined as the solution to the equation

$$S_{mc}(S_{mr1}) = S_{mc,H},$$

which gives the value of the areal material ratio corresponding to the height $S_{mc,H}$.

If the approximation to the inverse areal material ratio function is stored as a piecewise polynomial in MATLAB structure format, the MATLAB function `fzero` can be used to solve

$$\widehat{S}_{mc}(\widehat{S}_{mr1}) = \widehat{S}_{mc,H}$$

for an approximation \widehat{S}_{mr1} to S_{mr1} , with the knowledge that 0 % and 100 % constitute a bracket for the solution. The function implements an algorithm that uses a combination of bisection, secant, and inverse quadratic interpolation methods [17, 18]. If the approximation to the inverse areal material ratio function comprises a set of evenly-spaced areal material ratio function values and corresponding height values, the MATLAB function `interp1` can be used instead.

Figure 6 shows the relationship of S_{mr1} to the approximation of the areal material ratio function for test data set 1.

5.6. Dales material ratio

The function parameter S_{mr2} [2, clause 4.4.4.6] is defined as ratio of the area of the material at the intersection line which separates the protruding dales from the core surface to the evaluation area. It is determined as the solution to the equation

$$S_{mc}(S_{mr2}) = S_{mc,L},$$

which gives the value of the areal material ratio corresponding to the height $S_{mc,L}$.

As in section 5.5, MATLAB functions `fzero` or `interp1` can be used to obtain an approximation \widehat{S}_{mr2} to S_{mr2} .

Figure 6 shows the relationship of S_{mr2} to the approximation of the areal material ratio function for test data set 1.

5.7. Reduced peak height

The function parameter S_{pk} [2, clause 4.4.4.3] is defined as the average height of the protruding peaks above the core surface. It is the height of the right-angled triangle

constructed to have the same area as the ‘hill area’, and an approximation to S_{pk} is given by

$$\widehat{S}_{\text{pk}} = \frac{\int_{0\%}^{\widehat{S}_{\text{mr1}}} \widehat{S}_{\text{mc}}(p) - \widehat{S}_{\text{mc,H}} dp}{\widehat{S}_{\text{mr1}}/2}.$$

If the approximation to the inverse areal material ratio function is stored as a piecewise polynomial in MATLAB structure format, the MATLAB function `integral` can be used to integrate the polynomial $\widehat{S}_{\text{mc}}(p) - \widehat{S}_{\text{mc,H}}$ between 0 % and \widehat{S}_{mr1} . If the approximation to the inverse areal material ratio function comprises a set of evenly-spaced areal material ratio function values and corresponding height values, the MATLAB function `trapz` can be used to implement the numerical integration using the trapezoidal method.

Figure 7 shows the relationship of S_{pk} to the approximation of the areal material ratio function for test data set 1.

5.8. Reduced dale height

The function parameter S_{vk} [2, clause 4.4.4.4] is defined as the average height of the protruding dales below the core surface. It is the height of the right-angled triangle constructed to have the same area as the ‘dale area’, and an approximation to S_{vk} is given by

$$\widehat{S}_{\text{vk}} = \frac{\int_{\widehat{S}_{\text{mr2}}}^{100\%} \widehat{S}_{\text{mc,L}} - \widehat{S}_{\text{mc}}(p) dp}{(100\% - \widehat{S}_{\text{mr2}})/2}.$$

As in section 5.7, MATLAB functions `integral` or `trapz` can be used to evaluate the integral in the above expression.

Figure 7 shows the relationship of S_{vk} to the approximation of the areal material ratio function for test data set 1.

5.9. Peak extreme height

The function parameter S_{xp} [2, clause 4.4.7] is defined as the difference in heights at areal material ratios 2.5 % and 50 % [19, clause B.1.2], and an approximation to S_{xp} is given by

$$\widehat{S}_{\text{xp}} = \widehat{S}_{\text{mc}}(2.5\%) - \widehat{S}_{\text{mc}}(50\%).$$

Figure 8 shows the relationship of S_{xp} to the approximation of the areal material ratio function for test data set 1.

5.10. Material volume

The material volume $V_{\text{m}}(p)$ [2, clause 4.4.6] at a given material ratio p is defined as the volume of the material per unit area calculated from the (inverse) areal material ratio function, and an approximation to $V_{\text{m}}(p)$ is given by

$$\widehat{V}_{\text{m}}(p) = \frac{K}{100\%} \widehat{I}_{\text{m}}(p),$$

where

$$\widehat{I}_m(p) = \int_{0\%}^p \widehat{S}_{mc}(q) - \widehat{S}_{mc}(p) \, dq$$

and K is a constant to convert to units of millilitres per metres squared (for measured data having units of micrometres, $K = 1$).

5.11. Peak material volume

The function parameter V_{mp} [2, clause 4.4.6.1] is defined as the material volume at material ratio $p = 10\%$ [19, clause B.1.2], and an approximation to V_{mp} is given by

$$\widehat{V}_{mp} = \widehat{V}_m(10\%) = \frac{K}{100\%} \widehat{I}_m(10\%).$$

MATLAB functions `integral` or `trapz` can be used to evaluate the integral in the above expression.

Figure 9 shows the relationship of the area used in the evaluation of V_{mp} to the approximation of the areal material ratio function for test data set 1.

5.12. Core material volume

The function parameter V_{mc} [2, clause 4.4.6.2] is defined as the difference in material volumes at material ratios $p = 10\%$ and $q = 80\%$ [19, clause B.1.2], and an approximation to V_{mc} is given by

$$\widehat{V}_{mc} = \widehat{V}_m(80\%) - \widehat{V}_m(10\%) = \frac{K}{100\%} \left[\widehat{I}_m(80\%) - \widehat{I}_m(10\%) \right].$$

MATLAB functions `integral` or `trapz` can be used to evaluate the integrals in the above expression.

Figure 9 shows the relationship of the area used in the evaluation of V_{mc} to the approximation of the areal material ratio function for test data set 1.

5.13. Void volume

The void volume $V_v(p)$ [2, clause 4.4.5] at a given material ratio p is defined as the volume of the voids per unit area calculated from the (inverse) areal material ratio function, and an approximation to $V_v(p)$ is given by

$$\widehat{V}_v(p) = \frac{K}{100\%} \widehat{I}_v(p),$$

where

$$\widehat{I}_v(p) = \int_p^{100\%} \widehat{S}_{mc}(p) - \widehat{S}_{mc}(q) \, dq$$

and K is a constant to convert to units of millilitres per metres squared (for measured data having units of micrometres, $K = 1$).

5.14. Dale void volume

The function parameter V_{vv} [2, clause 4.4.5.1] is defined as the dale volume at material ratio $p = 80 \%$ [19, clause B.1.2], and an approximation to V_{vv} is given by

$$\widehat{V}_{vv} = \widehat{V}_v(80 \%) = \frac{K}{100 \%} \widehat{I}_v(80 \%).$$

MATLAB functions `integral` or `trapz` can be used to evaluate the integral in the above expression.

Figure 9 shows the relationship of the area used in the evaluation of V_{vv} to the approximation of the areal material ratio function for test data set 1.

5.15. Core void volume

The function parameter V_{vc} [2, clause 4.4.5.2] is defined as the difference in void volumes at material ratios $p = 10 \%$ and $q = 80 \%$ [19, clause B.1.2], and an approximation to V_{vc} is given by

$$\widehat{V}_{vc} = \widehat{V}_v(10 \%) - \widehat{V}_v(80 \%) = \frac{K}{100 \%} \left[\widehat{I}_v(10 \%) - \widehat{I}_v(80 \%) \right].$$

MATLAB functions `integral` or `trapz` can be used to evaluate the integrals in the above expression.

Figure 9 shows the relationship of the area used in the evaluation of V_{vc} to the approximation of the areal material ratio function for test data set 1.

6. Comparison methodology

Comparison of NPL and other software was undertaken using the same four test data sets used in [9, 10]. Data sets 1, 2 and 3 represent surfaces obtained by applying the surface finishing techniques of, respectively, honing, lishing and polishing. Data set 4 represents a surface that is periodic in the x - and y -directions. All four data sets comprise 512×512 surface heights at points defining a uniform grid with a sampling interval of $1 \mu m$ in both x - and y -directions.

For the evaluation of S-L surface parameters [2], an areal Gaussian filter with cut-off wavelength of $\lambda = 100 \mu m$ was applied.

The software packages report results for the parameters to different numerical precision. In the tables of numerical results provided in section 7, parameter values are presented according to the following rule: if the value is provided to at most three decimal digits, all available digits are presented; otherwise the value is presented after rounding to three decimal digits.

7. Results

Tables 1–4 give the results obtained from a number of proprietary software packages (labelled A–C)¶ and the NPL software (labelled NPL) for the four test data sets, respectively, for both S–F and S–L surfaces. Software packages A and B calculate all parameters, while software package C calculates all parameters apart from S_{xp} . Default units are used for all parameters [19, clause C.1.4, C.1.5].

The values in the final column (labelled CV) in each table provide an indication of the dispersion of the parameter values through the ‘coefficient of variation’. For each parameter, the value of this coefficient is given by the ratio of the standard deviation of all provided values of that parameter (including the NPL parameter value) to the mean parameter value, expressed as a percentage. The smaller the value of the coefficient of variation, the greater the (relative) agreement between the values of the parameter.

For the S–F surfaces, agreement between the results returned by the four software packages is generally good, with the coefficient of variation having a maximum value of 3.3 %. Compared to software packages A and B, the results returned by software package C are often closer to those returned by the NPL software, suggesting that there may be similarities in the ways in which software package C and the NPL software implement the calculations of the function parameters.

For the S–L surfaces, agreement between the results returned by the four software packages is noticeably worse than for the S–F surfaces, reflected in the generally higher values of the coefficient of variation. Again, agreement between the results returned by the NPL software and software package C is generally better than that between the NPL software and software packages A and B. For the fourth data set, agreement between the results returned by the NPL software and the software packages A and B is much worse than for the first three data sets.

Differences between the results returned by the software packages for an S–F surface may arise due to differences in their approaches to one or more of the steps involved in parameter evaluation. Choices must be made regarding:

- (i) The representation of the surface;
- (ii) The calculation of approximations to the areas of regions of the surface with height greater than or equal to a specified value;
- (iii) The determination of an approximation to the areal material ratio function;
- (iv) The determination of an approximation to the inverse areal material ratio function;
- (v) The determination of the equivalent straight line for the inverse areal material ratio function (for parameters S_k , S_{mr1} , S_{mr2} , S_{pk} and S_{vk});
- (vi) The calculation of the area enclosed by the inverse areal material ratio function (for parameters S_{pk} , S_{vk} , V_{mp} , V_{mc} , V_{vv} and V_{vc}).

¶ The degree to which comparison of results can be carried out is limited by the small number of proprietary software packages currently available.

For an S–L surface, differences in results may be caused by differences in the approaches not only to the above steps but to the application of a Gaussian areal filter to the data defining an S–F surface to determine the S–L surface and to the way the evaluation area (A_6) is chosen.

In general, the proprietary software packages do not provide information on their approaches to the steps involved in parameter evaluation. Based on the results provided by the three software packages for the four data sets, it is reasonable to conclude that the combination of steps implemented within software package C provide results that compare most favourably with those returned by the NPL software.

Differences between results are also seen to be more or less pronounced depending on the nature of the surface being processed, as illustrated by the fourth data set.

8. Summary

This paper describes software, available free to download from the website of the National Physical Laboratory (NPL), for the evaluation of approximations to a set of ten areal surface texture function parameters. Definitions for the parameters, expressed in terms of the inverse areal material ratio function, are provided, along with details of the numerical approaches implemented within the software to evaluate approximations to the parameters.

Results obtained using the NPL software are compared with those returned by a number of proprietary software packages. Differences are observed, both between NPL results and the results returned by the other software packages, and between the results obtained by pairs of software packages themselves. Those differences are observed to be greater for S–L surfaces than for S–F surfaces. In both cases, differences can reasonably be expected. For several steps within the parameter evaluation process, choices have to be made regarding the implementation of those steps and can be made on the basis of both numerical accuracy and processing time. It is intended to undertake more detailed investigation into the sources of the differences in the results.

Future work will focus on areal surface texture parameters that are classified as ‘feature’ parameters. Approximations to the parameters may also be evaluated by the NPL software described in this paper.

Acknowledgments

The National Measurement and Regulation Office of the UK Department for Business, Energy and Industrial Strategy supported this work as part of its Engineering Measurement and Materials and Modelling programmes. R K Leach would like to thank EPSRC Grant No. EP/M008983/1 for supporting the project. R K Leach and L D Todhunter would like to thank EPSRC Grant No. EP/I017933/1, Digital Surf and NPL for supporting the project. Alistair Forbes (NPL) made valuable comments on a draft of this paper.

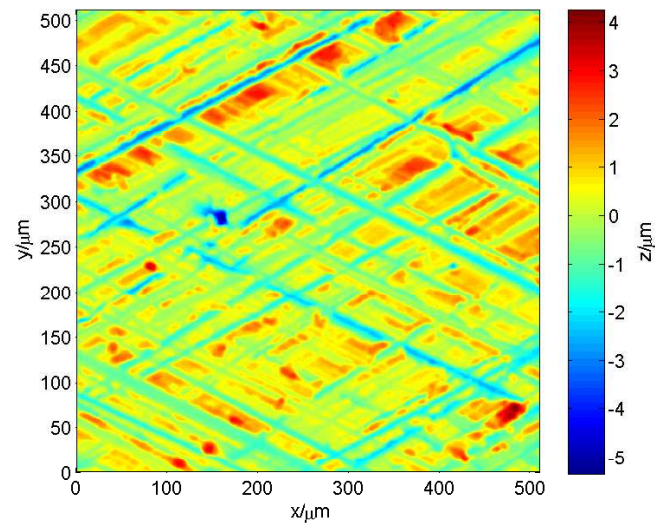


Figure 1. Data set 1: S-F surface.

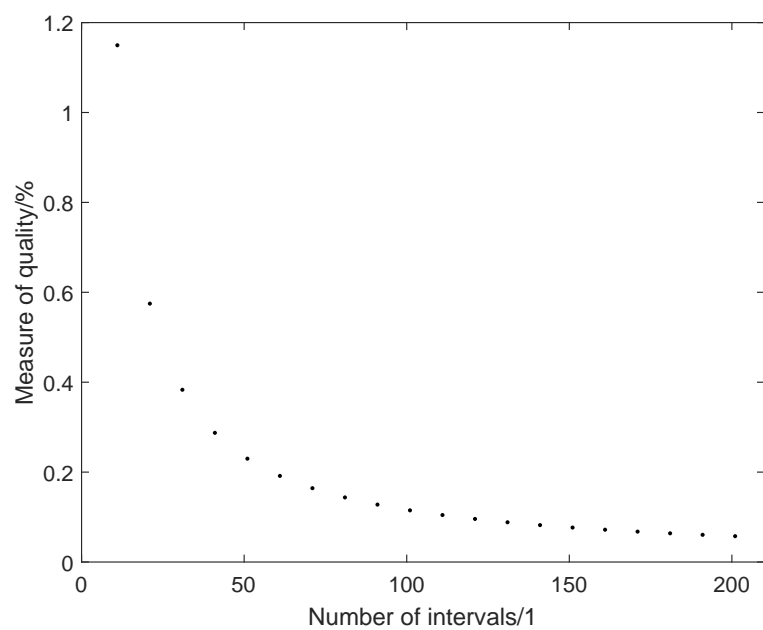


Figure 2. Measure \hat{Q} of the quality of the approximation to the areal material ratio function for test data set 1 for different numbers n_I of intervals.

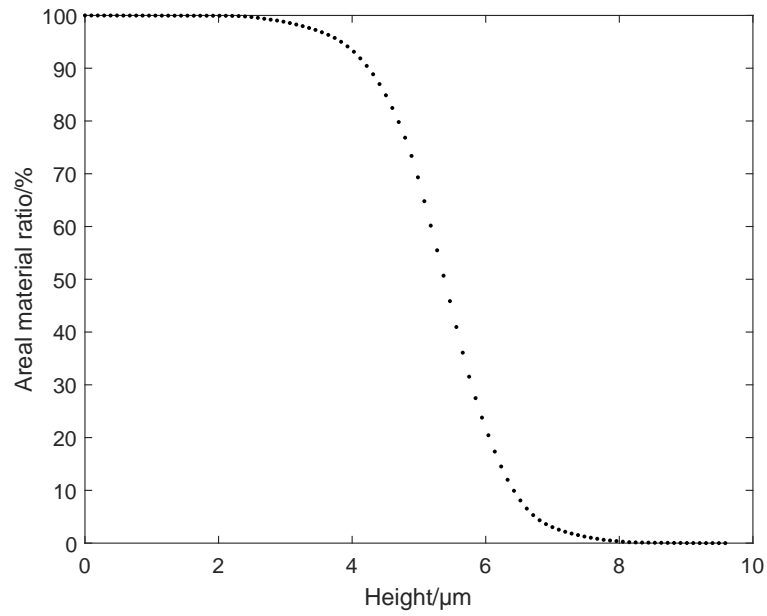


Figure 3. Approximations to the values of the areal material ratio function calculated with $n_I = 101$ for test data set 1.

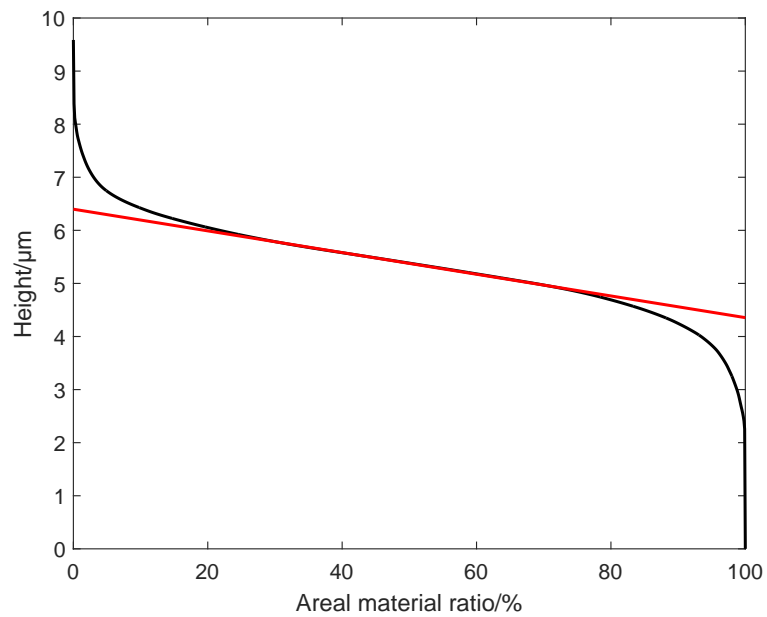


Figure 4. Approximations to the inverse areal material ratio function and equivalent straight line for test data set 1.

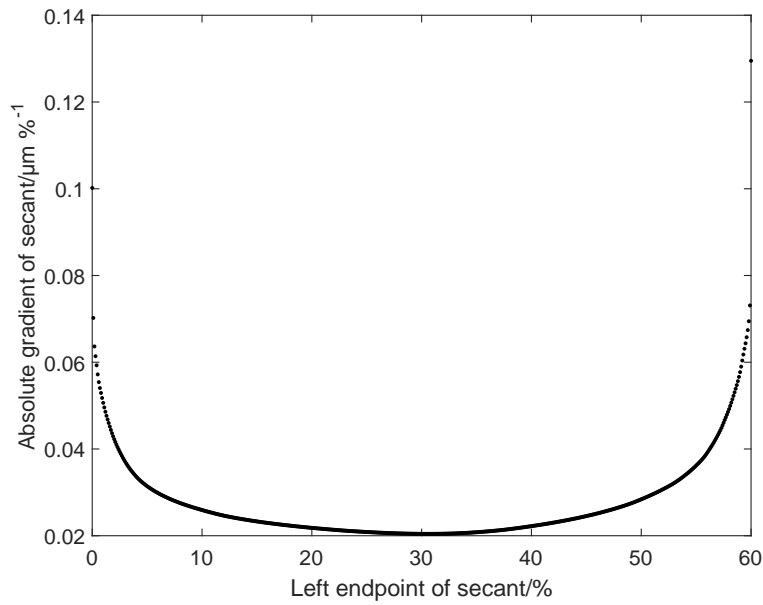


Figure 5. Absolute gradients of secants calculated for $n_p = 1\,001$ for test data set 1. The minimum value of absolute gradient corresponds to $p_L = 30.6\%$.

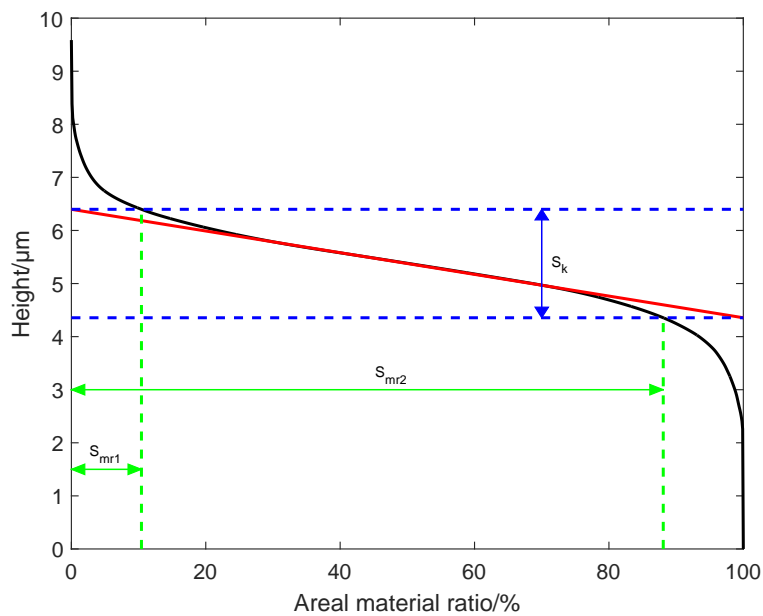


Figure 6. Relationships of S_k , S_{mr1} and S_{mr2} to the approximation of the inverse areal material ratio function for test data set 1.

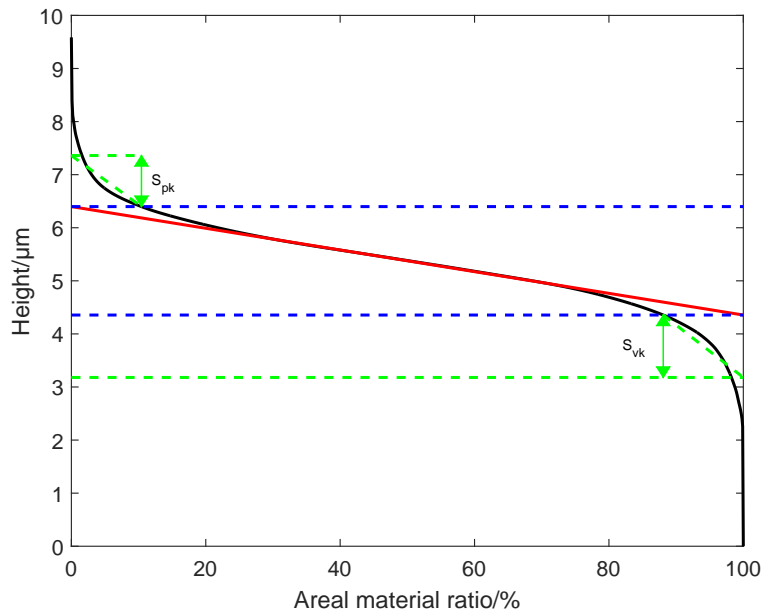


Figure 7. Relationships of S_{pk} and S_{vk} to the approximation of the inverse areal material ratio function for test data set 1.

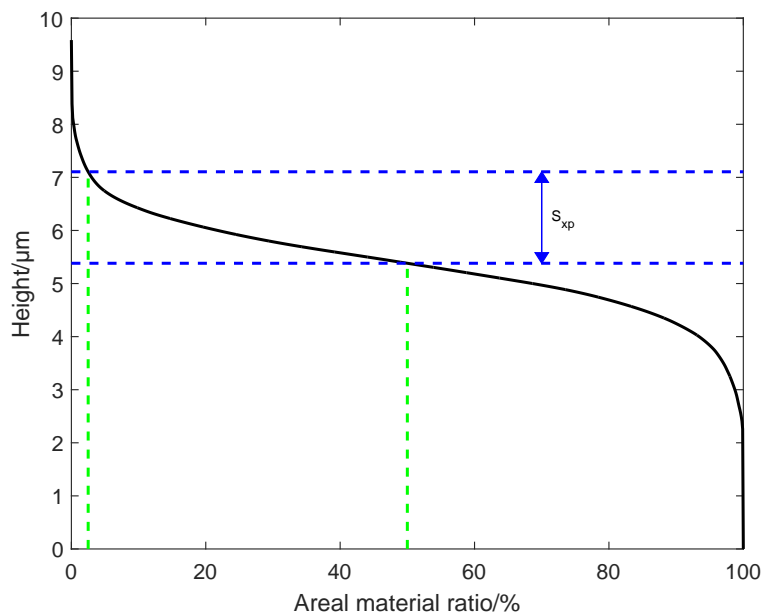


Figure 8. Relationship of S_{xp} to the approximation of the inverse areal material ratio function for test data set 1.

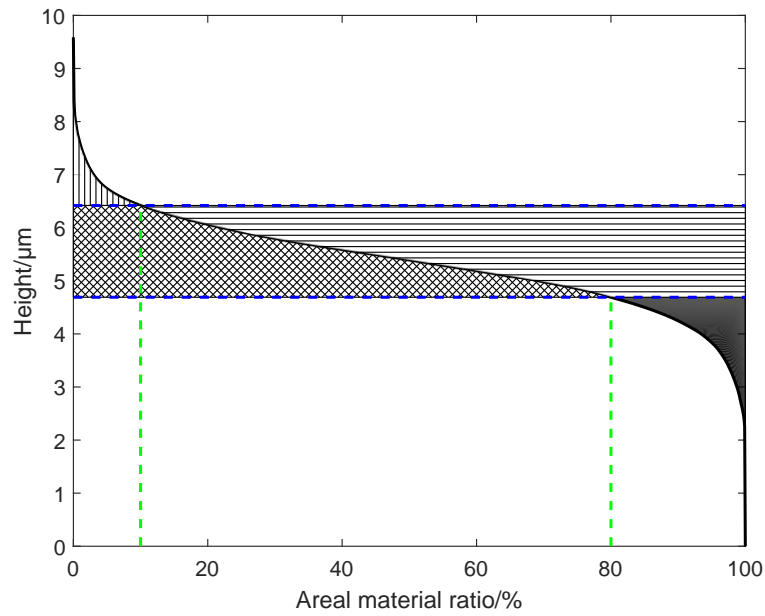


Figure 9. Relationships of the areas used in the evaluation of V_{mp} (vertically hatched region), V_{mc} (cross-hatched region), V_{vv} (solidly filled region) and V_{vc} (horizontally hatched region) to the approximation of the inverse areal material ratio function for test data set 1.

Table 1. Values of function parameters and coefficient of variation for data set 1 for S-F surface (top) and S-L surface (bottom).

	Software				CV/%
	A	B	C	NPL	
$S_k/\mu\text{m}$	2.042	2.055	2.035	2.043	0.4
$S_{mr1}/\%$	10.652	10.4	10.45	10.432	1.1
$S_{mr2}/\%$	88.002	88.1	88.13	88.086	0.1
$S_{pk}/\mu\text{m}$	0.929	0.981	0.958	0.957	2.2
$S_{vk}/\mu\text{m}$	1.209	1.214	1.165	1.170	2.1
$S_{xp}/\mu\text{m}$	1.731	1.781	–	1.723	1.8
$V_{mp}/\text{ml m}^{-2}$	0.047	0.048	0.048	0.048	1.0
$V_{mc}/\text{ml m}^{-2}$	0.740	0.741	0.736	0.739	0.3
$V_{vv}/\text{ml m}^{-2}$	0.124	0.123	0.121	0.122	1.1
$V_{vc}/\text{ml m}^{-2}$	0.998	0.990	0.987	0.990	0.5

	Software				CV/%
	A	B	C	NPL	
$S_k/\mu\text{m}$	1.775	1.934	1.647	1.699	7.1
$S_{mr1}/\%$	9.287	9	8.99	9.014	1.6
$S_{mr2}/\%$	85.187	87.2	85.98	86.056	1.0
$S_{pk}/\mu\text{m}$	0.738	0.946	0.632	0.644	19.6
$S_{vk}/\mu\text{m}$	1.204	1.149	1.054	1.083	6.0
$S_{xp}/\mu\text{m}$	1.430	1.636	–	1.287	12.1
$V_{mp}/\text{ml m}^{-2}$	0.038	0.046	0.032	0.033	17.2
$V_{mc}/\text{ml m}^{-2}$	0.679	0.707	0.618	0.634	6.2
$V_{vv}/\text{ml m}^{-2}$	0.127	0.118	0.112	0.116	5.4
$V_{vc}/\text{ml m}^{-2}$	0.844	0.896	0.765	0.786	7.2

Table 2. As for table 1 for data set 2.

	Software				CV/%
	A	B	C	NPL	
$S_k/\mu\text{m}$	0.648	0.646	0.631	0.636	1.3
$S_{mr1}/\%$	5.841	5.9	6.19	6.046	2.6
$S_{mr2}/\%$	88.191	88.3	87.97	88.039	0.2
$S_{pk}/\mu\text{m}$	0.134	0.131	0.127	0.130	2.2
$S_{vk}/\mu\text{m}$	0.302	0.301	0.297	0.297	0.9
$S_{xp}/\mu\text{m}$	0.399	0.407	–	0.390	2.1
$V_{mp}/\text{ml m}^{-2}$	0.008	0.008	0.008	0.008	0.0
$V_{mc}/\text{ml m}^{-2}$	0.222	0.222	0.216	0.218	1.4
$V_{vv}/\text{ml m}^{-2}$	0.034	0.034	0.034	0.034	0.0
$V_{vc}/\text{ml m}^{-2}$	0.270	0.267	0.26	0.263	1.7

	Software				CV/%
	A	B	C	NPL	
$S_k/\mu\text{m}$	0.643	0.632	0.566	0.611	5.6
$S_{mr1}/\%$	3.653	3.8	3.35	4.013	7.5
$S_{mr2}/\%$	87.306	89.3	87.34	88.119	1.1
$S_{pk}/\mu\text{m}$	0.060	0.073	0.076	0.085	14.1
$S_{vk}/\mu\text{m}$	0.254	0.246	0.236	0.233	4.0
$S_{xp}/\mu\text{m}$	0.338	0.353	–	0.334	2.9
$V_{mp}/\text{ml m}^{-2}$	0.004	0.006	0.005	0.005	16.3
$V_{mc}/\text{ml m}^{-2}$	0.224	0.209	0.196	0.207	5.5
$V_{vv}/\text{ml m}^{-2}$	0.032	0.030	0.029	0.030	4.2
$V_{vc}/\text{ml m}^{-2}$	0.266	0.249	0.227	0.247	6.5

Table 3. As for table 1 for data set 3.

	Software				CV/%
	A	B	C	NPL	
$S_k/\mu\text{m}$	0.085	0.086	0.084	0.085	1.0
$S_{mr1}/\%$	7.564	7.5	7.51	7.485	0.5
$S_{mr2}/\%$	85.914	85.9	86.03	86.114	0.1
$S_{pk}/\mu\text{m}$	0.024	0.026	0.025	0.025	3.3
$S_{vk}/\mu\text{m}$	0.048	0.048	0.047	0.047	1.2
$S_{xp}/\mu\text{m}$	0.058	0.059	–	0.057	1.7
$V_{mp}/\text{ml m}^{-2}$	0.001	0.001	0.001	0.001	0.0
$V_{mc}/\text{ml m}^{-2}$	0.032	0.032	0.032	0.032	0.0
$V_{vv}/\text{ml m}^{-2}$	0.005	0.005	0.005	0.005	0.0
$V_{vc}/\text{ml m}^{-2}$	0.039	0.038	0.037	0.038	2.1
	Software				CV/%
	A	B	C	NPL	
$S_k/\mu\text{m}$	0.068	0.07	0.06	0.064	6.8
$S_{mr1}/\%$	9.035	8.2	9.06	8.928	4.6
$S_{mr2}/\%$	87.785	87.5	88.05	88.201	0.4
$S_{pk}/\mu\text{m}$	0.022	0.022	0.021	0.022	2.3
$S_{vk}/\mu\text{m}$	0.043	0.039	0.035	0.036	9.4
$S_{xp}/\mu\text{m}$	0.049	0.049	–	0.046	3.6
$V_{mp}/\text{ml m}^{-2}$	0.001	0.001	0.001	0.001	0.0
$V_{mc}/\text{ml m}^{-2}$	0.024	0.025	0.021	0.023	7.3
$V_{vv}/\text{ml m}^{-2}$	0.004	0.004	0.004	0.004	0.0
$V_{vc}/\text{ml m}^{-2}$	0.031	0.032	0.027	0.030	7.2

Table 4. As for table 1 for data set 4.

	Software				CV/%
	A	B	C	NPL	
$S_k/\mu\text{m}$	1.303	1.307	1.307	1.307	0.2
$S_{mr1}/\%$	11.489	11.5	11.47	11.457	0.2
$S_{mr2}/\%$	88.939	89.1	89.18	89.126	0.1
$S_{pk}/\mu\text{m}$	0.355	0.36	0.363	0.360	0.9
$S_{vk}/\mu\text{m}$	0.291	0.286	0.283	0.285	1.2
$S_{xp}/\mu\text{m}$	0.926	0.953	–	0.926	1.7
$V_{mp}/\text{ml m}^{-2}$	0.016	0.016	0.016	0.016	0.0
$V_{mc}/\text{ml m}^{-2}$	0.488	0.485	0.488	0.489	0.4
$V_{vv}/\text{ml m}^{-2}$	0.047	0.047	0.046	0.046	1.2
$V_{vc}/\text{ml m}^{-2}$	0.658	0.657	0.656	0.656	0.1

	Software				CV/%
	A	B	C	NPL	
$S_k/\mu\text{m}$	0.752	0.56	0.25	0.253	54.3
$S_{mr1}/\%$	8.479	13.3	8.17	8.288	26.1
$S_{mr2}/\%$	88.904	89.8	89.15	89.089	0.4
$S_{pk}/\mu\text{m}$	0.105	0.248	0.031	0.032	98.1
$S_{vk}/\mu\text{m}$	0.239	0.107	0.082	0.082	59.0
$S_{xp}/\mu\text{m}$	0.472	0.509	–	0.155	51.4
$V_{mp}/\text{ml m}^{-2}$	0.006	0.011	0.002	0.002	81.4
$V_{mc}/\text{ml m}^{-2}$	0.263	0.209	0.086	0.087	55.2
$V_{vv}/\text{ml m}^{-2}$	0.032	0.019	0.011	0.011	54.3
$V_{vc}/\text{ml m}^{-2}$	0.347	0.300	0.113	0.115	56.0

Appendix A. Analytical approach to area evaluation

The approximate approach to area evaluation described in section 4 requires, for a specified height c , the classification of the rectangular regions $A_{i,j}$ into groups G_1 , G_2 and G_3 . Groups G_1 and G_3 may be used to determine analytically the required area.

Consider the region $A_{i,j}$ in group G_3 , defined by the surface heights $f_{i,j}$, $f_{i,j+1}$, $f_{i+1,j+1}$ and $f_{i+1,j}$ at its bottom left, top left, top right and bottom right vertices, respectively. It is assumed that $f_{i,j}$ and $f_{i,j+1}$ satisfy

$$f_{i,j} < c < f_{i,j+1}. \quad (\text{A.1})$$

With one exception (discussed later in this appendix), to treat a region for which constraint (A.1) is not satisfied, the region can be transformed to one for which the constraint does apply by effecting one of the following transformations:

- A clockwise rotation of 90° .
- A clockwise rotation of 180° .
- A clockwise rotation of 270° .
- A reflection about the vertical axis of reflection of the region.
- A reflection about the vertical axis of reflection of the region, followed by a clockwise rotation of 90° .

The required area is invariant to such a transformation.

Under the assumption $f_{i,j} < c < f_{i,j+1}$, the contour $f(x,y) = c$ may take one of seven forms, depending on the surface heights at the vertices and their relation to c . Identification of the form of the contour allows an appropriate approach to be used to evaluate the area $|A_{i,j}(c)|$ of the part of the region $A_{i,j}$ with height greater than or equal to c . Figure A1 shows an example of each of the seven forms of contour. Below, for each form, the relationship between surface heights and c , a description of the contour, and the approach to evaluate of $|A_{i,j}(c)|$, is provided. For simplicity, area evaluation is implemented for a unit square, i.e., a square having sides of length 1. For a rectangle having sides of length Δ_x and Δ_y , the area obtained for the unit square must be multiplied by $\Delta_x\Delta_y$.

- Form 1: If $f_{i+1,j+1} = f_{i+1,j} = c$, the contour comprises a single horizontal line running from the left edge to the right edge, and the part of the region with height greater than or equal to c is a rectangle. $|A_{i,j}(c)|$ is given by

$$|A_{i,j}(c)| = 1 - Y_1,$$

where

$$Y_1 = \frac{c - f_{i,j}}{f_{i,j+1} - f_{i,j}}.$$

- Form 2: If $f_{i+1,j+1} \leq c$ and $f_{i+1,j} < c$, or $f_{i+1,j+1} < c$ and $f_{i+1,j} \leq c$, the contour comprises a curve running from the left edge to the upper edge. The contour $f(X, Y) = c$ defined in expression (1) can be rearranged to express Y in terms of X :

$$Y(X) = (R_3 + R_4X)/(R_1 + R_2X),$$

or X in terms of Y :

$$X(Y) = -(R_3 - R_1Y)/(R_4 - R_2Y),$$

where

$$R_1 = f_{i,j+1} - f_{i,j}, \quad R_2 = f_{i,j} + f_{i+1,j+1} - f_{i,j+1} - f_{i+1,j}, \quad R_3 = c - f_{i,j} \quad \text{and} \quad R_4 = f_{i,j} - f_{i+1,j}.$$

$|A_{i,j}(c)|$ is given by

$$|A_{i,j}(c)| = X_U - \int_{X_L}^{X_U} Y(X) dX,$$

where $X_L = 0$ and $X_U = -(R_3 - R_1)/(R_4 - R_2)$, and using the result (omitting the constant of integration)

$$\int Y(X) dX = \frac{(R_3R_2 - R_4R_1)}{R_2^2} \log(R_1 + R_2X) + \frac{R_4X}{R_2}.$$

A special case arises when $R_2 = 0$. In this case, the contour $f(X, Y) = c$ can be rearranged to express Y in terms of X :

$$Y(X) = R_3/R_1 + R_4X/R_1,$$

or X in terms of Y :

$$X(Y) = -R_3/R_4 + R_1Y/R_4,$$

i.e., the contour comprises a straight line, and the part of the region with height greater than or equal to c is a triangle. $|A_{i,j}(c)|$ is given by

$$|A_{i,j}(c)| = X_1(1 - Y_1)/2,$$

where

$$X_1 = \frac{c - f_{i,j+1}}{f_{i+1,j+1} - f_{i,j+1}} \quad \text{and} \quad Y_1 = \frac{c - f_{i,j}}{f_{i,j+1} - f_{i,j}}.$$

- Form 3: If $f_{i+1,j+1} > c$ and $f_{i+1,j} \leq c$, the contour comprises a curve running from the left edge to the right edge. $|A_{i,j}(c)|$ is given by

$$|A_{i,j}(c)| = 1 - \int_{X_L}^{X_U} Y(X) dX,$$

where $X_L = 0$ and $X_U = 1$ and $\int Y(X) dX$ is as defined for form 2. A special case arises when $R_2 = 0$. In this case, the part of the region with height greater than or equal to c is a trapezoid. $|A_{i,j}(c)|$ is given by

$$|A_{i,j}(c)| = 1 - (Y_1 + Y_2)/2,$$

where

$$Y_1 = \frac{c - f_{i,j}}{f_{i,j+1} - f_{i,j}} \text{ and } Y_2 = \frac{c - f_{i+1,j}}{f_{i+1,j+1} - f_{i+1,j}}.$$

- Form 4: If $f_{i+1,j+1} \geq c$ and $f_{i+1,j} > c$, the contour comprises a curve running from the left edge to the lower edge. $|A_{i,j}(c)|$ is given by

$$|A_{i,j}(c)| = 1 - \int_{X_L}^{X_U} Y(X) dX,$$

where $X_L = 0$ and $X_U = -R_3/R_4$ and $\int Y(X) dX$ is as defined for form 2. A special case arises when $R_2 = 0$. In this case, the part of the region with height greater than or equal to c is a pentagon. $|A_{i,j}(c)|$ is given by

$$|A_{i,j}(c)| = 1 - (X_1 Y_1)/2,$$

where

$$X_1 = \frac{c - f_{i,j}}{f_{i+1,j} - f_{i,j}} \text{ and } Y_1 = \frac{c - f_{i,j}}{f_{i,j+1} - f_{i,j}}.$$

- Form 5: If $f_{i+1,j+1} < c$ and $f_{i+1,j} > c$ and

$$\frac{f_{i,j+1} - c}{c - f_{i,j}} - \frac{c - f_{i+1,j+1}}{f_{i+1,j} - c} = 0,$$

the contour comprises two lines, a horizontal line running from the left edge to the right edge, and a vertical line running from the lower edge to the upper edge. The part of the region with height greater than or equal to c is made up of two rectangles. $|A_{i,j}(c)|$ is given by

$$|A_{i,j}(c)| = X_1(1 - Y_1) + (1 - X_1)Y_1 = X_1 + Y_1 - 2X_1Y_1,$$

where

$$X_1 = \frac{c - f_{i,j}}{f_{i+1,j} - f_{i,j}}, \quad Y_1 = \frac{c - f_{i,j}}{f_{i,j+1} - f_{i,j}}.$$

- Form 6: If $f_{i+1,j+1} < c$ and $f_{i+1,j} > c$ and

$$\frac{f_{i,j+1} - c}{c - f_{i,j}} - \frac{c - f_{i+1,j+1}}{f_{i+1,j} - c} < 0,$$

the contour comprises two curves, one running from the left edge to the upper edge, the other running from the lower edge to the right edge. The part of the region with height greater than or equal to c is made up of two patches. Figure A2 shows how $|A_{i,j}(c)|$ can be evaluated for the form 6 contour of figure A1. The areas of the top left and bottom right patches are to be evaluated and summed. For the top left patch, the rectangle to the left of the dotted line (top row, left figure) is extracted (top row, right figure) and the shaded area is evaluated using the approach for a form 2 contour. For the bottom right patch, the region is rotated through 180° , the rectangle to the left of the dotted line (bottom row, left figure) is extracted (bottom row, right figure) and the shaded area is evaluated using the approach for a form 2 contour.

- Form 7: If $f_{i+1,j+1} < c$ and $f_{i+1,j} > c$ and

$$\frac{f_{i,j+1} - c}{c - f_{i,j}} - \frac{c - f_{i+1,j+1}}{f_{i+1,j} - c} > 0,$$

the contour comprises two curves, one running from the left edge to the lower edge, the other running from the upper edge to the right edge. The part of the region with height greater than or equal to c is a single patch. Figure A3 shows how $|A_{i,j}(c)|$ can be evaluated for the form 7 contour of figure A1. The areas of the bottom left and top right patches (shaded white) are to be evaluated and summed, and their sum subtracted from 1. For the bottom left patch, the rectangle enclosed by the dotted lines (top row, left figure) is extracted (top row, right figure), and the shaded area is evaluated using the approach for a form 3 contour and subtracted from the area of the extracted rectangle. For the top right patch, the region is rotated through 180° , the rectangle enclosed by the dotted lines (bottom row, left figure) is extracted (bottom row, right figure), and the shaded area is evaluated using the approach for a form 3 contour.

An eighth form of contour arises on considering the region for which

$$f_{i,j} < c, \quad f_{i,j+1} = c, \quad f_{i+1,j+1} > c \quad \text{and} \quad f_{i+1,j} = c. \quad (\text{A.2})$$

The surface heights do not satisfy constraint (A.1) and therefore this region must be considered separately. Note that a region for which

$$f_{i,j} > c, \quad f_{i,j+1} = c, \quad f_{i+1,j+1} < c \quad \text{and} \quad f_{i+1,j} = c,$$

can be transformed to one for which constraint (A.2) does apply by effecting a clockwise rotation of 180° . Figure A1 shows an example of a form 8 contour.

- Form 8: If $f_{i,j} < c$, $f_{i,j+1} = c$, $f_{i+1,j+1} > c$ and $f_{i+1,j} = c$, the contour comprises a curve running from the top left vertex to the bottom right vertex. $|A_{i,j}(c)|$ is given by

$$|A_{i,j}(c)| = 1 - \int_{X_L}^{X_U} Y(X) dX,$$

where $X_L = 0$ and $X_U = 1$ and $\int Y(X) dX$ is as defined for form 2. A special case arises when $R_2 = 0$. In this case, the part of the region with height greater than or equal to c is an isosceles right-angled triangle and $|A_{i,j}(c)| = 0.5$.

The area $|A(c)|$ of the region $A(c)$ corresponding to the part of the surface with height greater than or equal to c is given by

$$|A(c)| = \Delta_x \Delta_y \left(n_0 + \sum_{(i,j) \in \mathcal{G}_3} |A_{i,j}(c)| \right).$$

A measure of the quality of the approximation $\widehat{S}_{\text{mr}}(c)$ to the areal material ratio function $S_{\text{mr}}(c)$ is

$$\max_{k=1, \dots, n_c} \left\{ 100 \frac{|A(c_k)| - |\widehat{A}(c_k)|}{|A|} \% \right\}.$$

Figure A4 shows the above measure, calculated for $n_I = 101$, plotted against n_I for test data set 1.

Implementation of the analytical approach to area evaluation also allows parameters $S_{\text{mr}1}$ (section 5.5) and $S_{\text{mr}2}$ (section 5.6) to be evaluated analytically.

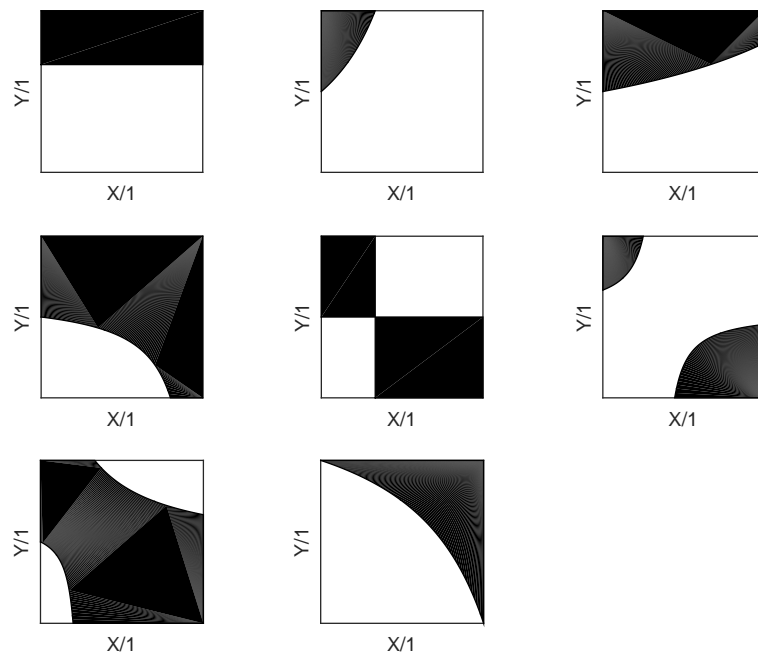


Figure A1. Examples of contours of forms 1–3 (top row from left to right), 4–6 (middle row from left to right) and 7–8 (bottom row). Shaded portions indicate the areas to be evaluated.

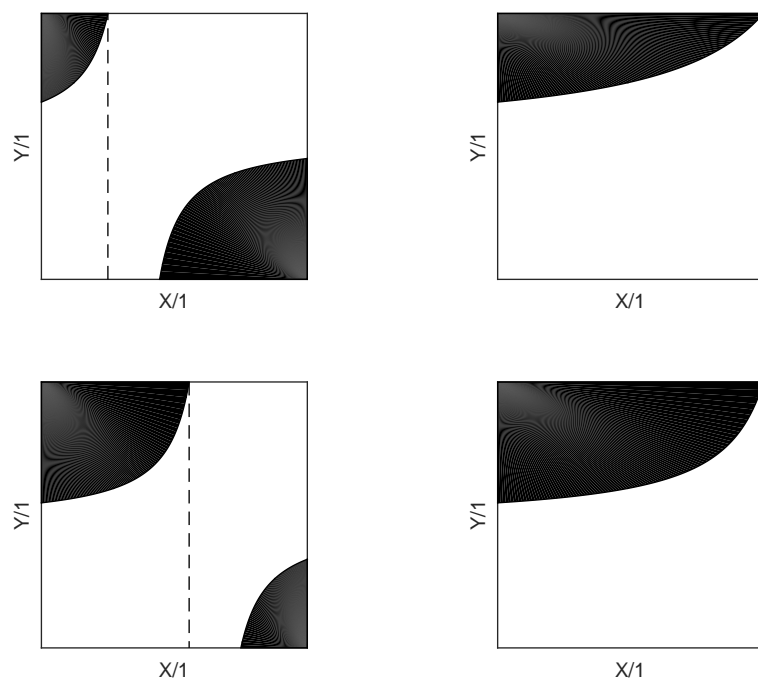


Figure A2. Area evaluation for form 6 contours.

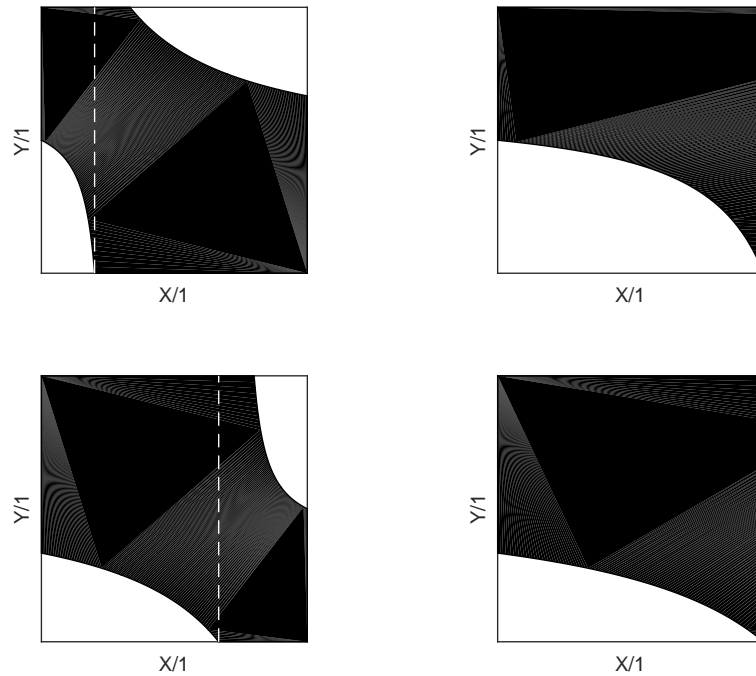


Figure A3. Area evaluation for form 7 contours.

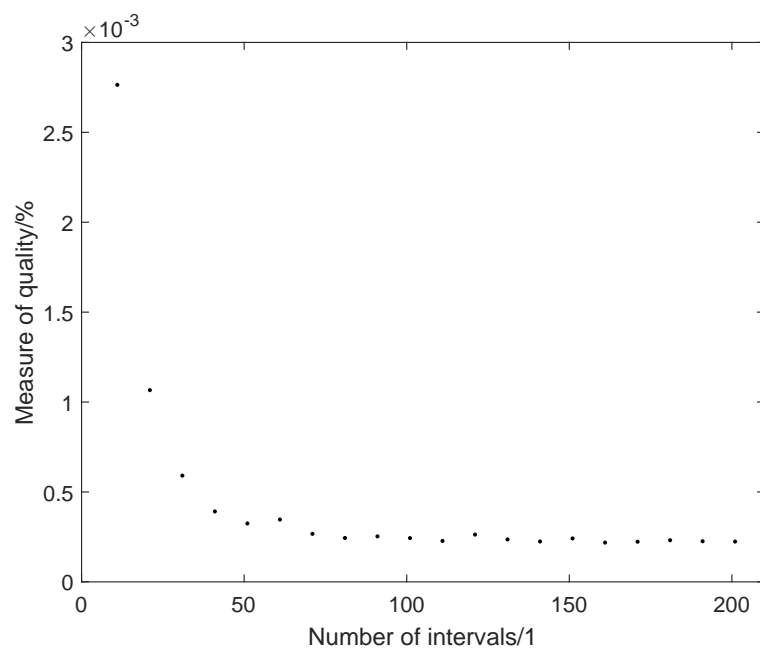


Figure A4. Measure of the quality of approximation to the areal material ratio function for test data set 1.

References

- [1] Leach R K 2009 *Fundamental Principles of Engineering Nanometrology* Elsevier, Amsterdam
- [2] ISO 25178-2:2012 *Geometrical product specification (GPS) — Surface texture: Areal — Part 2: Terms, definitions and surface texture parameters* International Organization for Standardization
- [3] Blunt L, Jiang X, Leach R K, Harris P M and Scott P 2008 The development of user-friendly software measurement standards for surface topography software assessment *Wear* **264**, 389–393
- [4] ISO 5436-2:2002 *Geometrical Product Specification (GPS) — Surface texture: Profile method — Measurement standards — Software measurement standards* International Organization for Standardization
- [5] ISO 4287:2000 *Geometrical Product Specification (GPS) — Surface texture: Profile method — Terms, definitions and surface texture parameters* International Organization for Standardization
- [6] Bui S H, Renegar T B, Vorburger T V, Raja J and Malburg M C 2004 Internet-based surface metrology algorithm testing system *Wear* **257**, 1213–1218
- [7] Bui S H and Vorburger T V 2007 Surface metrology algorithm testing system *Precision Engineering* **31**, 218–225
- [8] Jung L, Spranger B, Krüger-Sehm R and Krystek M 2004 Reference software for roughness analysis — features and results *Proc. XI Int. Colloq. Surfaces, Chemnitz, Germany, Feb.*, 164-170
- [9] Harris P M, Smith I M, Leach R K, Giusca C, Jiang X and Scott P 2011 Software measurement standards for areal surface texture parameters: part 1 – algorithms *Meas. Sci. Technol.* **23** doi:10.1088/0957-0233/23/10/105008
- [10] Harris P M, Smith I M, Chaoyi W, Leach R K and Giusca C 2011 Software measurement standards for areal surface texture parameters: part 2 – comparison of software *Meas. Sci. Technol.* **23** doi:10.1088/0957-0233/23/10/105009
- [11] Software for calculating areal surface texture function and feature parameters — Available to download from filexch.npl.co.uk/cgi-bin/download.pl?id=syxxziteueunfhmfvgbyz between 27 March and 26 May 2017, subsequently from NPL website (updated link to be provided in final version of this paper).
- [12] Leach R K ed. 2013 *Characterisation of Areal Surface Texture* Springer, Berlin
- [13] ISO/DIS 25178-72 *Geometrical Product Specification (GPS) — Surface texture: Areal — Part 72: XML file format X3P* International Organization for Standardization
- [14] Wiora G and Seewig J 2010 Free Interchange of 3D Measuring Data: X3P — a Flexible, System-independent Open Source Data Format *Inspect* **11** pp 16-17
- [15] Press W H, Flannery B P, Teukolsky S A and Vetterling W T 1992 *Numerical Recipes in FORTRAN: The Art of Scientific Computing* 2nd ed. Cambridge, England: Cambridge University Press
- [16] MATLAB ©1994–2017 The MathWorks, Inc.
- [17] Brent R 1973 *Algorithms for Minimization Without Derivatives* Prentice-Hall.
- [18] Forsythe G E, Malcolm M A and Moler C B 1976 *Computer Methods for Mathematical Computations* Prentice-Hall.
- [19] ISO 25178-3:2012 *Geometrical product specification (GPS) — Surface texture: Areal — Part 3: Specification operators* International Organization for Standardization