

Development of mathematical reference standards for the validation of surface texture parameter calculation software

Luke Todhunter¹, Richard Leach¹, Simon Lawes¹, François Blateyron² and Peter Harris³

¹Manufacturing Metrology Team, University of Nottingham, NG8 1BB, UK

²Digital Surf, Besançon, France

³National Physical Laboratory, Teddington, TW11 0LW, UK

E-mail: Luke.Todhunter@nottingham.ac.uk

Abstract. A framework for the validation of surface texture parameter calculation software is proposed. The framework utilises mathematically defined surfaces and mathematically calculated surface texture parameter values to produce a reference against which third-party software can be compared; in principle, free from the approximations that are intrinsic when applying numerical methods and algorithms to discrete data in order to realise continuous definitions. This paper provides a proof-of-concept of the new framework using a simple two-term cosine surface. The required steps to enable meaningful comparison and subsequent validation of surface texture parameter calculation software are showcased.

1. Introduction

Measurement of surface texture has become increasingly important in assessing manufactured part properties, such as friction and wear, and modern areal surface texture measurement techniques are popular among both research and industry [1, 2]. Characterising surface texture is performed using surface texture parameters: numerical descriptors of the topography of the surface that are used to assess whether a manufactured part surface is fit for purpose [3, 4]. Typically, surface texture parameters are calculated using software. These software packages are often commercial in nature and prioritise speed and user experience, potentially compromising the accuracy of calculated parameter values. The task of validating these commercial software packages falls to national measurement institutes (NMIs), which are independent bodies responsible for providing traceability of measurement results to reference standards. The basis of the validation is the provision by NMIs of reference data and reference software, jointly referred to as software measurement standards [5-9].

Because surface datasets obtained via measurement are discrete in nature, both commercial and reference software typically perform calculations on the datasets which are not equivalent to the continuous definitions given in ISO 25178-2 [4]. In addition, these calculations involve numerical methods undertaken in finite precision arithmetic that lead to numerical errors.. This approach to characterising surfaces leads to parameter value uncertainties, independent of measurement noise, which are not currently quantified: a deficiency in the approach to validation currently offered by NMIs. As a result of these issues, surface texture parameter values calculated by reference software have associated uncertainties, and parameter values provided by different NMI software can differ even for the same surface dataset [10].

2. A mathematical approach

The proposed approach involves creating a series of mathematically defined continuous surface functions for which continuous definitions of surface texture parameters can be realised. Accurate surface texture parameters are calculated for these mathematical surfaces, free from numerical approximations, directly using the continuous parameter definitions found in ISO 25178-2.

The next step is to obtain parameter values for the same surface using the surface texture parameter calculation software under validation. The majority of software requires discrete datasets, and so a discrete dataset representation of the continuous mathematical surface is required. The surface texture parameter calculation software can then be tested by performing a comparison between the software obtained values and the mathematical values. Such a comparison must account for effects such as discretisation error, as the software will have been given a discrete representation of the continuous surface.

By avoiding numerical methods and approximations wherever possible, the authors aim to produce a fully traceable framework for validating surface texture parameter calculation software that delivers consistent and reproducible results. The following sections of this paper serve to deliver an example of the framework in action as a proof-of-concept. Subsequent work by the authors will seek to expand upon this further.

3. Proof-of-concept: A simple cosine surface example

In this section, the proposed approach to surface texture parameter calculation software validation will be demonstrated in a step-by-step fashion with a two dimensional surface comprising two cosine terms.

3.1. Defining a continuous mathematical surface

A function $z(x, y)$ must be defined such that for any position on the surface (given as x and y coordinates), the function describes a height value z . A useful method for defining mathematical surfaces is to utilise a Fourier series, using the idea that any signal can be reproduced, to a stated approximation with sufficiently many terms, as a summation of multiple cosine terms of the form

$$z(x, y) = \sum_l \sum_m A_{l,m} \cos \left(2\pi \left(\frac{l(x + \phi_x)}{N_x} + \frac{m(y + \phi_y)}{N_y} \right) \right) \quad (1)$$

where $A_{l,m}$ is the amplitude, N_x and N_y are the periods for the x and y components of the cosine, l and m are integers and ϕ_x and ϕ_y are phase terms. Combining large numbers of these terms enables the creation of complex, well-defined continuous surfaces with a simple basic structure. An example surface created using this method is one with two cosine terms, given as

$$z(x, y) = \cos \pi x + \cos \pi y \quad (2)$$

where x and y define the lateral position on the surface. This example is given in metre-scale for simplicity. For more realistic surfaces, micrometre or millimetre scales are typically used.

3.2. Surface texture parameter calculation

The vast majority of surface texture parameters in ISO 25178-2 are expressed as continuous functions. For the simple cosine surface defined in equation (2), this leads to a definition for S_{ku} , for example, of

$$S_{ku} = \left(\frac{1}{|A|} \iint_A (\cos \pi x + \cos \pi y)^2 dx dy \right)^{-2} \left(\frac{1}{|A|} \iint_A (\cos \pi x + \cos \pi y)^4 dx dy \right). \quad (3)$$

where A is the domain area over which the integration is performed and $|A|$ is the physical size of the area under evaluation. Evaluating this equation can be performed either manually or using a symbolic mathematics software package. With an evaluation area of 100 m², from -5 m to 5 m in the x and y directions, this leads to an S_{ku} parameter value of 9/4.

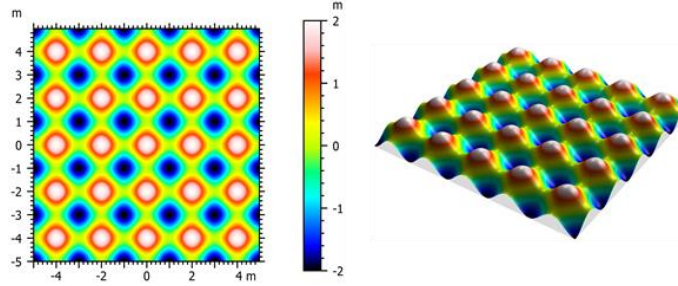


Figure 1. Discrete dataset representation of the cosine surface defined in equation (2). *Left:* Top-down colour-map of the dataset. *Right:* 3D view of the dataset.

3.3. Discrete dataset representation

Creating a discrete dataset from the mathematical surface can be achieved by calculating height values $z(x, y)$ at uniformly separated x, y positions within a defined area and storing the values in a grid array. Creating the discrete dataset by evaluating the mathematical surface at each point ensures numerical agreement between the mathematical surface and the discrete surface, up to the finite precision of the data file. The data must be formatted for incorporation into a file type suitable for input into the required software. An example is the .SDF file type, a standard file format defined in ISO 25178-71 [5]. Figure 1 shows an .SDF 500×500 pixel discrete dataset sampled from the surface defined in equation (2).

3.4. Software validation

Using the discrete dataset, it is now possible to obtain surface texture parameter values from the software under test and compare them to the mathematical values. The differences between the software and mathematical parameter values for the surface defined in equation (2), and the discrete representation shown in figure 1, are shown in figure 2.

As the dataset only contains a finite number of points, it cannot contain all the information of the continuous surface. This discretisation error must be accounted for to ensure the differences between methods are not due to the loss of information in the discrete dataset. Theoretically, the effect of discretisation error can be removed if the software is given a dataset with an infinite number of points. This infinite dataset would contain the same amount of information as the continuous surface. In practice, the construction of such a dataset is not possible. Instead, this method can be simulated by calculating parameter values for many representations of the surface, each with increasing pixel density. The obtained parameter values can then be extrapolated to find the asymptote. An example of this is shown in figure 3, in which the S_a parameter for the surface defined in equation (2) is calculated for a range of dataset densities and extrapolated further by fitting the relative differences between parameter values using a power law equation. Using the convergent values in subsequent comparisons with the mathematically obtained parameter values enables much more meaningful comparisons to be made. It is important to note that the final convergent value depends critically on the extrapolation method used, and further work is required to ensure that extrapolation is realistic and robust.

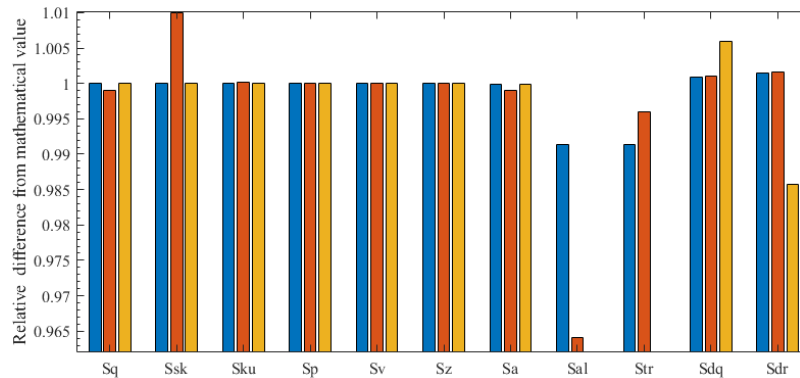


Figure 2. Relative surface texture parameter values for three software packages (blue, orange and yellow), divided by the mathematical value to enable comparisons between parameter values.

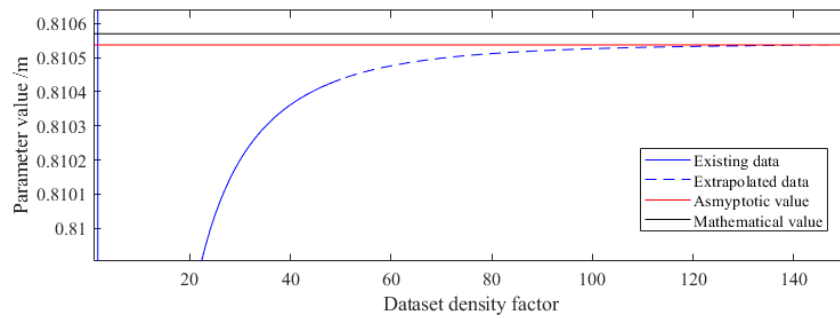


Figure 3. Extrapolation of S_a surface texture parameter values from software, obtained for a range of dataset densities from 10×10 to 500×500 pixels, and extrapolated to find an asymptote.

4. Conclusion

The state-of-the-art in software measurement standards for surface texture parameters contain deficiencies due to applying numerical methods to discrete representations of surfaces when calculating continuously defined parameters. The proposed methods provide a way of validating software by relying on choosing mathematically defined surface functions for which it is possible to realise the continuous definitions of surface texture parameters and applying those definitions with small numerical error. Sampling these surfaces to produce discrete datasets enables comparison with numerical algorithms used in surface texture parameter calculation software and allows validation of the software to occur.

4.1. Future work

Extending the concept introduced in this paper to more complex surfaces will better represent surfaces obtained from real-world measurements. Additionally, work needs to be done to calculate a wide range of surface texture parameter values for these complex surfaces, as simple symbolic mathematics software methods are limited in their solving power, even on high specification machines.

A direct comparison between the mathematically and software obtained values is limited in its usefulness. Future work aims to develop performance metrics that assess the deviation from the mathematical value in combination with the measurement error of a real input surface, to determine whether the software is a significant factor in the overall uncertainty.

Acknowledgements

We would like to thank the Engineering and Physical Sciences Research Council (EPSRC Grants EP/M008983/1 and EP/1017933/1).

References

- [1] Leach R 2014 *Fundamental Principles of Engineering Nanometrology* (Binghamton: Elsevier Science)
- [2] Whitehouse D J 2010 *Handbook of Surface and Nanometrology* (Boca Raton: CRC Press)
- [3] Leach R 2013 *Characterisation of Areal Surface Texture* (Berlin: Springer)
- [4] ISO 25178-2 2012 *Geometrical product specifications (GPS) - Surface texture: Areal Part 2: Terms, definitions and surface texture parameters* (Geneva: International Organisation for Standardisation)
- [5] ISO 25178-71 2012 *Geometrical product specifications (GPS) - Surface texture: Areal - Part 71: Software measurement standards* (Geneva: International Organisation for Standardisation)
- [6] Harris P M, Smith I M, Leach R K, Giusca C, Jiang X and Scott P 2012 Software measurement standards for areal surface texture parameters: part 1—algorithms *Meas. Sci. Technol.* **23** 105008
- [7] Blunt L, Jiang X, Leach R, Harris P and Scott P 2008 The development of user-friendly software measurement standards for surface topography software assessment *Wear* **264** 389–93
- [8] Bui S H and Vorburger T V 2007 Surface metrology algorithm testing system *Precis. Eng.* **31** 218–25
- [9] Jung L, Krüger-Sehm R, Spranger B and Koenders L 2004 Reference software for roughness analysis - features and results *XI International Colloquium on Surfaces Chemnitz, Germany* 164–169
- [10] Todhunter L D, Leach R K, Lawes S D A and Blateyron F 2017 An analysis of type F2 software measurement standards for profile surface texture parameters *Meas. Sci. Technol.* **28** 065017