

PAPER • OPEN ACCESS

A tracking algorithm for Monte Carlo simulation of surface roughness in EPMA measurements

To cite this article: D Sánchez-Gonzalo *et al* 2018 *IOP Conf. Ser.: Mater. Sci. Eng.* **304** 012015

View the [article online](#) for updates and enhancements.

You may also like

- [Effect of Ni on the kinetics of intermetallic compounds evolution on the Sn-0.7Cu-10Bi-xNi/Co interface during various reflow](#)
He Gao, Fuxiang Wei, Caixia Lin et al.
- [Optimization of Te Solution Chemistry in the Electrochemical Atomic Layer Deposition Growth of CdTe](#)
Brian Perdue, Justin Czerniawski, Julie Anthony et al.
- [Analysis of Partial Crystallization in Yb³⁺ Doped Aluminophosphosilicate Fiber Preforms Prepared with Organic Chelate Precursor Doping Technique](#)
Rong Luo, Weinan Li, Chaoqi Hou et al.



ECS
The
Electrochemical
Society
Advancing solid state &
electrochemical science & technology

DISCOVER
how sustainability
intersects with
electrochemistry & solid
state science research

A tracking algorithm for Monte Carlo simulation of surface roughness in EPMA measurements

D Sánchez-Gonzalo¹, X Llovet², R Graciani¹ and F Salvat¹

¹ Facultat de Física (FQA and ICC). Universitat de Barcelona. Diagonal 647, ES-08028 Barcelona, Spain

² Scientific and Technological Centres. Universitat de Barcelona. Lluís Solé i Sabarís, 1-3. ES-08028 Barcelona, Spain

E-mail: francesc.salvat@ub.edu

Abstract.

We present an algorithm for Monte Carlo simulation of particle trajectories through rough interfaces. The microscopic topography of the rough surface is assumed to be described by an altitude map in a dense rectangular grid. The tracking algorithm has been implemented in a Fortran subroutine package, which is coupled to the code system PENELOPE/PENEPMA to perform Monte Carlo simulations of X-ray emission from samples with rough surfaces irradiated by electron beams. To validate the numerical procedure, electron probe microanalysis (EPMA) simulations of an ideal sample with periodic surface roughness have been performed by using 1) the PENGEOM geometry package of the PENELOPE code system and 2) the new tracking algorithm. The results from the two simulations are found to be equivalent, that is, their differences are generally less than the associated statistical uncertainties. Results from simulations of samples with realistic rough surfaces are also presented.

1. Introduction

Conventional quantification procedures for electron probe microanalysis (EPMA) assume flat bulk specimens. However, in some cases specimens must be analysed without any surface preparation either because they cannot be destroyed (e.g., cultural heritage objects) or because the information of interest is in the surface itself (e.g., corrosion studies). The analysis of rough targets is challenging because of the dependence of the measured line intensities on the topography of the sample surface [1,2]. The same type of difficulty is found in X-ray fluorescence (XRF) [3–5], and in the transport of light through scintillating crystals and optical systems with rough interfaces [6].

The effect of surface roughness on radiation transport processes, can be studied by means of Monte Carlo simulation [7, 8]. In EPMA and electron microscopy, the area of the sample that is effectively probed has a limited extension, and the relevant portion of the surface can be described numerically, e.g., by using information from stereoscopic or confocal microscopy. Generally, though, the extension of a rough surface is too large to allow a microscopic description, and one must assume that the surface roughness is “homogeneous”, in the sense that the statistical roughness parameters do not change appreciably with position. Assuming that curvature is not too large, the rough surface can be described as the combination of a



mathematical (smooth) surface and a numerical model representing a finite portion of a nearly planar rough surface with characteristics similar to those of the real surface.

In this study we describe tracking strategies to account for the effect of roughness of the sample surface (and of interfaces between different materials) within a conventional Monte Carlo transport code for material structures defined by mathematical surfaces. In the simulations presented below we use the Monte Carlo code system PENELOPE [9], which describes the material system where radiation propagates by using constructive quadric geometry [10], that is, as a set of homogeneous bodies limited by quadric surfaces.

A rough surface is assumed to be defined within a rectangular cuboid, which will be referred to as “the box”, by means of an altitude map on a dense two-dimensional rectangular grid. A continuous representation of the rough surface is obtained by triangularisation of the altitude map. For the simulation of radiation transport through rough surfaces we have adopted a simple scheme in which the neighbourhood of the mathematical surface is replaced by the rough surface box. During simulation, when a particle reaches a mathematical surface that corresponds to a rough interface, an external subprogramme is called to perform the tracking through the associated surface box. Control is returned to the main programme when the incident particle and any secondary particles generated during the process have left the box. This scheme allows keeping the structure of the conventional simulation code practically unaltered.

Simulations of EPMA measurements, that is, of X-ray emission from targets bombarded by electron beams, can be performed with the Fortran main programme PENEPMA [11], which uses PENELOPE as the simulation engine and the quadric geometry package PENGEOM [10]. To allow consideration of targets with a rough surface, we have written a new programme, named EPMAR, which generates the same information as PENEPMA and uses the present tracking algorithm to describe the transport of particles through the rough irradiated surface. To verify the correctness of the proposed tracking algorithm, we consider homogeneous targets with an artificial rough surface consisting of a periodic array of square pyramids. This artificial sample structure can be described by both PENEPMA and EPMAR. The results from the two codes are found to agree within statistical uncertainties. We present also results from EPMAR simulations with a realistic rough surface.

2. Tracking of particles near a rough surface

When the particle being transported approaches a mathematical surface representing a rough interface, the main programme switches to tracking within the corresponding surface box. To simplify the geometrical operations, the motion of particles within the box is referred to a right-handed reference frame with the origin at one vertex of the box and the axes parallel to the edges, so that the box lies in the first octant. The plane $z = 0$ is regarded as the base of the box, and the surface is described by giving its altitude $z(x, y)$ at the points of a regular Cartesian grid with spacings Δ_x, Δ_y . The definition of the surface is provided by the user in an ASCII file with three columns, which specify the coordinates x_i ($i = 1 : N_x$) and y_j ($j = 1 : N_y$) of each grid point, and the surface altitude $z_{i,j}$, respectively. It is assumed that the x coordinate varies first in the file, that is, that the triad $x_i, y_j, z_{i,j}$ is given in the line number $i + (j - 1)N_x$ of the file. Since

$$x_i = (i - 1)\Delta_x, \quad y_j = (j - 1)\Delta_y, \quad (1)$$

the arrays x_i and y_j do not need to be stored in memory. To avoid inconsistencies resulting from numerical roundoff, our subroutines define a box of height $\Delta_z = 1.1S$, where S is the difference in altitudes of the highest and lowest point of the original surface, and the surface is shifted so that the altitudes of the lowest and highest points are $0.05\Delta_z$ and $\Delta_z - 0.05S$, respectively. That is, the shifted values $z_{i,j}$ are all such that $0.05S \leq z_{i,j} \leq \Delta_z - 0.05S$. A continuous surface is obtained by introducing a triangular mesh. The grid points of each cell $(x_i, x_{i+1}) \times (y_j, y_{j+1})$ define two possible pairs of triangles (see fig. 1). The surface generally looks smoother when

using the pair that intersect along the diagonal of the cell that has the largest slope (case b in fig. 1) and, consequently, this is the triangularisation scheme adopted in our algorithm. Thus, each cell defines a column consisting of two sections, separated by the surface triangles. Note that we are assuming that the volumes below and above the surface are filled with two different materials.

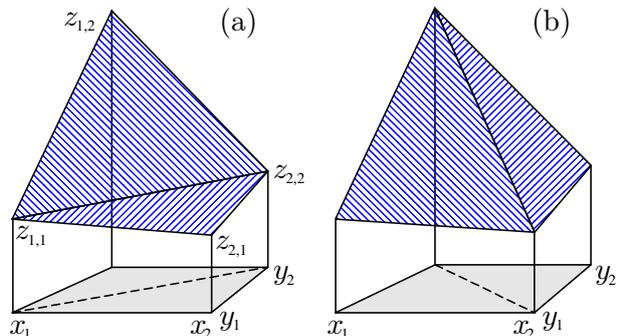


Figure 1. Possible triangularisations of a cell $(x_1, x_2) \times (y_1, y_2)$, characterised by the slopes of the diagonals of the cell.

In Monte Carlo simulations the trajectory of a particle is described as a sequence of connected linear steps, or “jumps”, of length Δs determined by the physics subroutines. Each step corresponds to a segment of a ray defined by the equation

$$\mathbf{r}(s) = \mathbf{r}(0) + s\hat{\mathbf{d}}. \quad (2)$$

where the vectors $\mathbf{r}(0) = (x(0), y(0), z(0))$ and $\hat{\mathbf{d}} = (u, v, w)$ are, respectively, the position and the direction of motion at the start of the step. Steps represent free flights of the particle, which end either with an interaction or when the particle arrives at an interface, i.e., a surface separating two different materials.

A conventional Monte Carlo code generates random particle histories sequentially, by sampling the length Δs of the steps, and the energy loss and the deflection angles caused by the interactions. The probability distribution functions of these variables are determined by the adopted interaction model (see, e.g., Ref. [9]). When a particle arrives at a mathematical interface (a quadric, in the case of PENELOPE/PENGEOM), the simulation code halts the particle and resumes simulation in the material behind the interface.

In the case of rough interfaces, the Monte Carlo code can still follow particles up to a distance $\simeq \Delta_z$ from the mathematical surface and then switch the simulation to the surface box. The tracking of particles through the box is performed by using the strategy and the formulas presented in the Appendix.

2.1. Samples with a rough external surface

A situation of interest in EPMA is that of a sample with a rough external surface that is bombarded by the electron beam. Because the probed volume has microscopic lateral dimensions, the relevant portion of the sample surface can be described numerically. We have written a dedicated programme, named EPMAR, which simulates EPMA measurements on targets having the structure sketched in fig. 2, with the rough surface box placed on top of a base sample. The latter is described by means of the PENGEOM subroutines, which allow defining embedded particles, lamellae, etc. Notice that the plane $z = 0$ is assumed to separate the box and the base sample. For simplicity, we also assume that the box and the base sample have the same lateral extensions, i.e., that both are limited by the planes $x = 0$, $x = \Delta_x$, $y = 0$ and $y = \Delta_y$. EPMAR was obtained by modifying the programme PENEPMA [11] to include the option of a rough external surface; the two codes operate similarly and generate the same information.

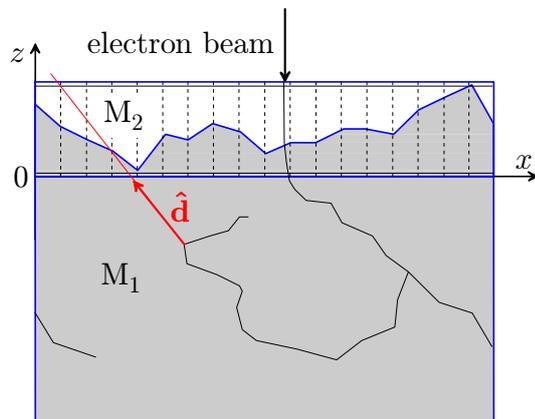


Figure 2. Schematics of the simulation of an EPMA measurement on a sample with a rough surface. The programme EPMAR allows using periodic boundary conditions in the x - y plane.

The simulation of particle histories within the box is performed by calling the PENELOPE subroutines and using the tracking strategy described in the Appendix. When a particle leaves the box through one of the lateral faces, we make it re-enter the opposite face thus imposing periodic boundary conditions in the x and y directions. Each particle is followed until it leaves the box through the lower or the upper plane, or it is absorbed within the material. Because secondary particles may occasionally be produced, they also have to be followed until they leave the box or are absorbed. When the simulation of the shower in the box is completed, the tracking of particles is resumed in the base sample using the PENGEOM subroutines. The process is repeated each time a particle arrives to the rough interface.

In practical EPMA of e.g., geological samples with lamellae, the analysed volume may present complicated structures, which approximately recur with a certain periodicity. The definition of the geometry of these samples with PENGEOM is very laborious. To simplify the work, the programme EPMAR offers the option of applying periodic boundary conditions in the x - y plane also to the base sample. In addition, although PENGEOM uses mathematical surfaces to limit the bodies in the base sample, the method described in Section 2.2 can be used to add roughness to the “internal” surfaces.

2.2. Internal rough interfaces

To obtain a general tracking algorithm we also consider geometries with macroscopic rough interfaces, which are each described by a box representing a finite sample portion of the rough surface. As indicated above, we assume that the main programme tracks particles in the conventional way, i.e., by using only mathematical surfaces, and surface roughness is taken into account only when a particle gets close to the mathematical surface. For the sake of programming simplicity, we let the main programme track a particle until it reaches the mathematical surface, which is then replaced by the surface box. The box is placed “parallel” to the mathematical surface in a position such that the particle crosses the rough surface with randomly selected x and y coordinates (in the box frame). The simulation of the particle is then continued within the box by using the tracking method described in the Appendix.

Let $\mathbf{r}_s = (x_s, y_s, z_s)$ and $\hat{\mathbf{d}}_s = (u_s, v_s, w_s)$ be the position and direction of motion of the particle when it arrives at the mathematical surface. The latter is then replaced by the box oriented so that its z axis is parallel or anti-parallel to the normal $\hat{\mathbf{n}} = (n_x, n_y, n_z)$ to the mathematical surface, making sure that the materials in the box are coincident with those in the PENGEOM geometry (see fig. 3). Since the tracking of particles within the box is performed with reference to the box frame, the position and direction vectors of the particle have to be transformed from the laboratory frame (i.e., from the reference frame adopted in the main programme) to the box frame. Consistently with the assumed “homogeneity” of the rough surface, the particle starts

its motion in the box from a random position $\mathbf{r} = (x, y, z)$ where the coordinates x and y are sampled uniformly and z is set equal to the altitude of the triangularised surface at (x, y) plus or minus a small amount, $\epsilon \simeq 10^{-8}$ cm, to ensure that the particle has just passed the surface. The direction vector $\hat{\mathbf{d}} = (u, v, w)$ in the box frame is obtained by rotating the laboratory vector $\hat{\mathbf{d}}_s$,

$$\hat{\mathbf{d}} = \mathcal{R}_B \hat{\mathbf{d}}_s, \quad (3)$$

where the rotation \mathcal{R}_B is required to transform the unit vector $\hat{\mathbf{z}} = (0, 0, 1)$ into the surface normal vector $\hat{\mathbf{n}}$. We use the following

$$\mathcal{R}_B = \mathcal{R}(\phi\hat{\mathbf{z}})\mathcal{R}(\theta\hat{\mathbf{y}}) \quad (4)$$

where θ and ϕ are the polar and azimuthal angles of the normal vector $\hat{\mathbf{n}}$, and

$$\mathcal{R}(\phi\hat{\mathbf{z}}) = \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathcal{R}(\theta\hat{\mathbf{y}}) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}, \quad (5)$$

are rotations of angles ϕ and θ around the z and y axes of the laboratory frame, respectively. That is,

$$\mathcal{R}_B = \begin{pmatrix} \cos \phi \cos \theta & -\sin \phi \cos \theta & \sin \theta \\ \sin \phi \cos \theta & \cos \phi \cos \theta & \sin \theta \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}, \quad (6)$$

with

$$\cos \theta = n_x, \quad \sin \theta = \sqrt{n_x^2 + n_y^2}, \quad \cos \phi = \frac{n_x}{\sqrt{n_x^2 + n_y^2}}, \quad \sin \phi = \frac{n_y}{\sqrt{n_x^2 + n_y^2}}. \quad (7)$$

When $n_x^2 + n_y^2 \ll 1$, so that $n_z \simeq \pm 1$, we have

$$\mathcal{R}_B = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad (8)$$

if $n_z \simeq -1$ and $\mathcal{R}_B = \mathcal{I}$, the identity matrix, if $n_z \simeq +1$.

The evolution of particle histories within the box is followed by calling the PENELOPE subroutines and using the tracking strategy described in the Appendix. When a particle leaves the box through one of the lateral faces, we make it re-enter the opposite face thus imposing periodic boundary conditions in the x and y directions. Each particle is followed until it leaves the box through the lower or the upper plane, or it is absorbed in the material. The handling of secondary particles is somewhat tricky, because they cannot be sent directly to the secondary stack of PENELOPE (see Ref. [9]), where all particles are referred to the laboratory frame. Secondary particles generated within the box are followed until they leave the box or are absorbed. Only when the simulation of the shower in the box is completed, the particles that have left through the upper and lower faces of the box are transferred to the secondary stack. Their position coordinates and directions of motion in the laboratory frame are obtained by applying the inverse of the rotation, eq. (4), to the vectors \mathbf{r} and $\hat{\mathbf{d}}$ in the box frame, followed by a translation \mathbf{r}_s of the position vectors, $\mathbf{r} \rightarrow \mathbf{r} + \mathbf{r}_s$.

After completing the simulation of the shower in the box, the main programme resumes normal simulation through the PENGEOG geometry. The process is repeated each time a particle arrives at a rough interface.

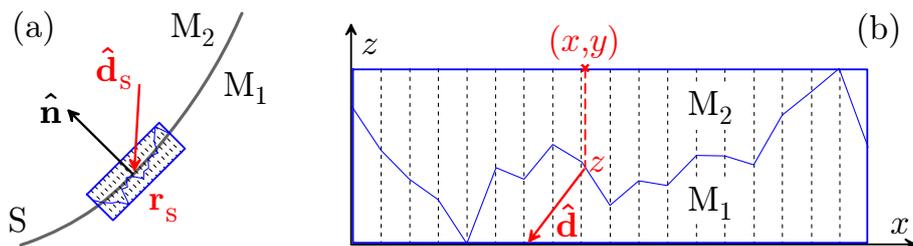


Figure 3. Geometrical transformation from the laboratory frame to the box frame of a rough “internal” surface. The particle reaches the mathematical surface at \mathbf{r}_s moving in the direction $\hat{\mathbf{d}}_s$. The simulation within the rough-surface box starts with the particle at a random position just below the triangularised surface and moving in the direction $\hat{\mathbf{d}} = \mathcal{R}_B \hat{\mathbf{d}}_s$, see eq. (3).

3. Validation and examples

To validate the present simulation scheme, as well as to reveal the effect of surface roughness, we have performed simulations of EPMA measurements on Fe samples with a rough external surface having a simple structure, namely a periodic array of square pyramids with $2\ \mu\text{m}$ base edges and lateral faces forming an angle α with the normal to the surface (fig. 4). This structure can be described by the programme PENEPMA, which uses the PENGEOM geometry subroutines, and by the EPMAR code, which implements the present tracking algorithm for rough surfaces. Notice that the adopted strategy for surface triangularisation (the triangles in each column share the diagonal with the largest inclination of the surface) reproduces the surface geometry exactly.

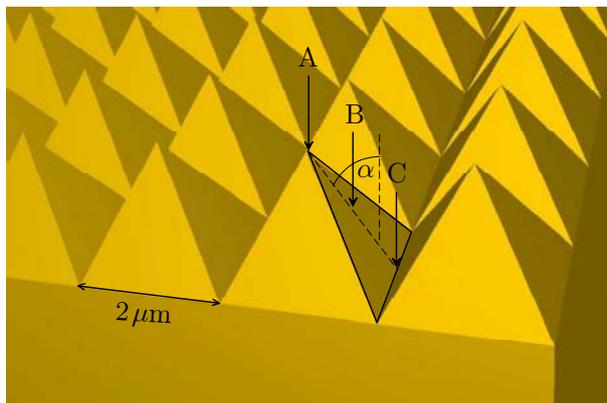


Figure 4. Geometry of the sample with a periodic “rough” surface used to verify the numerical tracking algorithm.

Simulations of identical arrangements were performed with the codes PENEPMA and EPMAR. Line intensities were tallied by using a detector that counts X-rays emitted from the sample in directions with polar and azimuthal angles in the intervals $(35^\circ, 55^\circ)$ and $(0^\circ, 90^\circ)$, respectively [see Ref. [11] for details]. A pencil beam of 20 keV electrons (with null lateral extent) impinged normally on the surface of the sample, at three different positions of a pyramid near the centre of the structure (to minimise edge effects). The numerical results from the two codes are equivalent, that is, their differences are less than, or of the order of their estimated statistical uncertainties. With the considered sample geometry, EPMAR is slightly more efficient than PENEPMA, and hence the numerical effort devoted to tracking particles through the rough surface box does not slow down the programme significantly.

Results from simulations for samples with pyramids of two different heights, corresponding to angles α of 30° and 60° are shown in table 1, together with results from the ideal target with a smooth surface ($\alpha = 90^\circ$). The latter case was also simulated with PENEPMA assuming a flat surface of the sample; the results agree with those of EPMAR to within statistical uncertainties. The effect of the surface roughness on the measured intensities is usually manifest.

Table 1. Intensities (number of X-rays per unit solid angle and per incident electron) of the indicated Fe X-ray lines for a 20 keV electron beam at the various impact positions in the pyramid (as indicated in fig. 4). The quoted relative uncertainties correspond to a 3σ confidence interval.

Line	α	A	B	C
K-L2	30	4.14E-5±2.1E-7	4.89E-5±2.4E-7	5.21E-5±2.6E-7
K-L2	60	4.21E-5±2.1E-7	4.81E-5±2.4E-7	5.14E-5±2.6E-7
K-L2	90	4.92E-5±2.4E-7	4.92E-5±2.4E-7	4.92E-5±2.4E-7
K-L3	30	8.13E-5±2.9E-7	9.56E-5±3.4E-7	1.02E-4±4.0E-7
K-L3	60	8.23E-5±3.0E-7	9.42E-5±3.4E-7	1.01E-4±3.6E-7
K-L3	90	9.61E-5±3.4E-7	9.61E-5±3.4E-7	9.61E-5±3.4E-7

Simulations have also been performed for a Fe sample with a realistic rough surface, a small portion of which is displayed in fig. 5. The sample was irradiated with a 20 keV electron beam, impinging normally on the surface box, at the positions indicated in fig. 5. As in the previous example, line intensities were tallied by considering a detector that counts X-rays emitted in directions with polar and azimuthal angles in the intervals $(35^\circ, 55^\circ)$ and $(0^\circ, 90^\circ)$, respectively. Simulations were performed with the programme EPMAR; each run involved a number of simulated showers such that the statistical relative uncertainty of the Fe K-L2 line intensity was less than 1 %. Figure 5 shows the surface profile along the line scanned by the electron beam, and the calculated intensity (number of X-rays per unit solid angle and per incident electron) of the Fe K-L2 line at the various beam positions.

These examples indicate that line intensities from EPMA measurements in bulk samples with a rough surface tend to be larger at the valleys of the surface and smaller at the peaks.

4. Concluding remarks

The tracking algorithm described in this communication provides a practical solution to account for surface roughness in conventional Monte Carlo transport codes using geometries defined by mathematical surfaces. In our implementation, much care has been exercised to protect the programme against numerical round-off errors. The use of a “movable” surface box allows describing the topography of rough surfaces on the scale of the nanometre.

The programme EPMAR performs Monte Carlo simulations of EPMA measurements on samples with rough surfaces. It not only accounts for the roughness of the irradiated surface of the sample, but also and permits considering “internal” rough surfaces to define material structures (inclusions or lamellae) within the base sample. The programme has been validated by comparison with results from PENEPMA in the case artificial surfaces with periodic roughness.

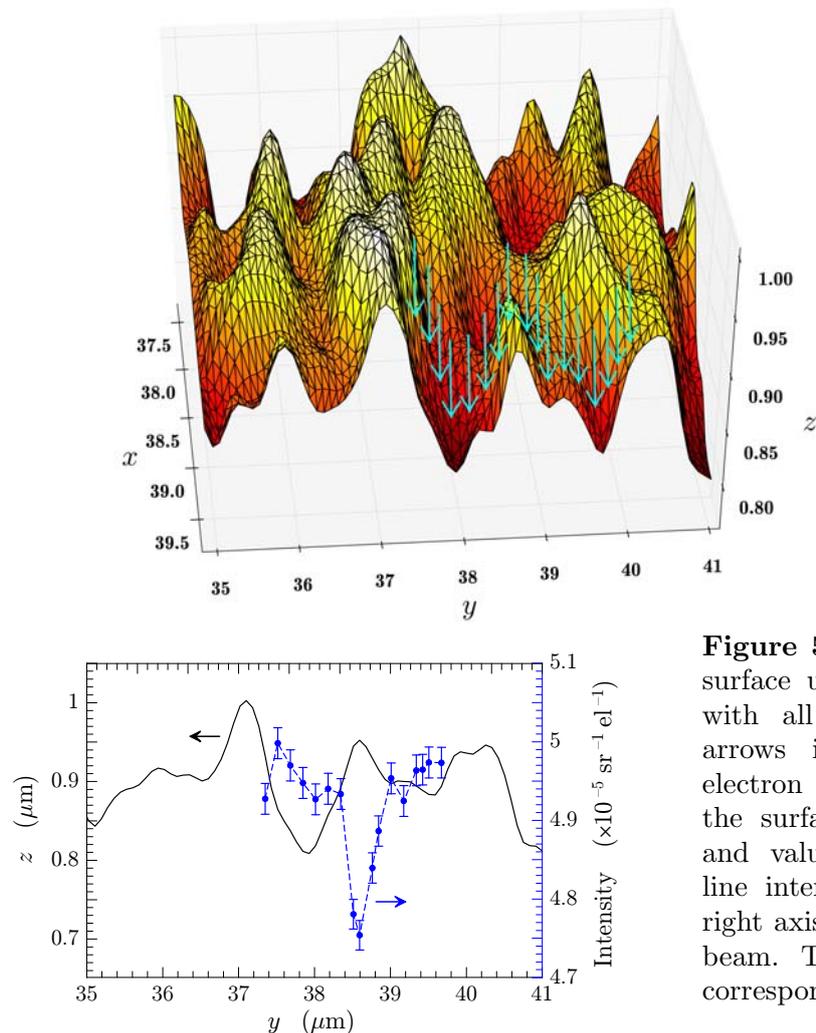


Figure 5. Top: A portion of the rough surface used in the EPMA simulations, with all lengths in micrometres; the arrows indicate the positions of the electron beam. Bottom: Line scan of the surface (solid black line, left axis) and values of the simulated Fe K-L2 line intensities (circles with error bars, right axis) at the various positions of the beam. The quoted relative uncertainties correspond to a 3σ confidence interval.

Acknowledgements

Financial support from the Spanish Ministerio de Economía y Competitividad and ERDF (projects nos. FPA2013-44549-P, FPA2014-57896-C4-2-R, and FPA2016-77689-C2-2-R) and from the Generalitat de Catalunya (grants 2014 SGR 846 and 2014 SGR 769) is gratefully acknowledged. DS-G wishes to thank the Industrial Doctorate Plan of the Generalitat de Catalunya for mobility support (Ref. 2016DI014).

References

- [1] Yamada A, Fons P, Matsubara K, Iwata K, Sakurai K and Niki S 2003 *Thin Solid Films* **431–432** 277–283
- [2] Newbury D E 2004 *Scanning* **26** 103–114
- [3] Kuhn W K and Andermann G 2001 *Appl. Surf. Sci.* **185** 84–91
- [4] Brunetti A and Golosio B 2014 *Spectrochim. Acta B* **94–95** 58–62
- [5] Naito M, Hasebe N, Kusano H, Nagaoka H, Kuwako M, Oyama Y, Shibamura E, Amano Y, Ohta T, Kim K J and Lopes J A M 2015 *Nucl. Instrum. Meth. A* **788** 182–187
- [6] Kilimchuk I V, Tarasov V A and Vlasova I D 2010 *Radiat. Meas.* **45** 383–385
- [7] Gauvin R and Lifshin 2000 *Microchim. Acta* **132** 201–204.
- [8] Sorbier L, Rosenberg E and Merlet C 2001 *Advanced Monte Carlo for radiation physics, particle transport simulation and applications*. (Kling A, Barao F, Nakagawa M, Tavora L and Vaz P; Eds.) (Berlin: Springer-Verlag) 389–394

- [9] Salvat F 2015 *PENELOPE-2014: a code system for Monte Carlo simulation of electron and photon transport*. (Issy-les-Moulineaux, France: OECD/NEA Data Bank)
- [10] Almansa J, Salvat-Pujol F, Díaz-Londoño G, Carnicer A, Lallena A M and Salvat F 2016 *Comput. Phys. Commun.* **199** 102–113
- [11] Llovet X and Salvat F 2017 *Microsc. Microanal.* **23** 634–646

Appendix. Tracking particles in the surface box

As indicated above, to describe the motion of particles within a surface box we use a reference frame with the origin at the box corner and axes parallel to the box sides. The tracking is simplified by considering the box as the set of elementary volumes defined by the “vertical” planes $x = x_i$, $y = y_j$ that pass through the grid points, eq. (1), and the triangularised surface. Thus, each cell $(x_i, x_{i+1}) \times (y_j, y_{j+1})$ determines a column volume with two sections of materials M_1 and M_2 separated by two surface triangles.

Given the initial position $\mathbf{r}(0)$ and direction of motion $\hat{\mathbf{d}}$ of a particle, the indices of the column that contains the particle are

$$i = [1 + x(0)/\Delta_x] \quad \text{and} \quad j = [1 + y(0)/\Delta_y], \quad (\text{A.1})$$

where $[x]$ denotes the integer part of x . The altitudes of the surface at the four vertices of the cell determine the two surface triangles (which share the diagonal of the cell corresponding to the largest slope). The material where the particle moves or, equivalently, the position of the particle relative to the surface (below or above), is determined by the altitude of the triangularised surface at the point $x(0), y(0)$, which can be readily calculated by simple linear interpolation of the three vertices of the corresponding triangle.

When the particle tries to jump a distance Δs , decided by the physics subroutines, we must determine whether the flight ends within the current material, the particle changes material by crossing the surface, or it leaves the box. To determine the actual move we consider the intersections of the ray $\mathbf{r}(0) + s\hat{\mathbf{d}}$ with the column planes, the two triangles, and the upper and lower sides of the box; evidently, only the intersections that occur ahead of the particle (i.e., at positive s) are relevant. When the first crossing is with one of the vertical planes, we move the particle to the corresponding column and repeat the process. If the first intersection is with one of the triangles, the particle is halted at the surface and the simulation is resumed in the new material. If the particle leaves the box, control is returned to the steering main program to continue tracking any secondary particles generated within the box, or to proceed with the simulation of particles outside the box. Evidently, the operation that is more laborious is the calculation of intersections with triangles, which is performed as follows.

A triangle is defined by the coordinates of its three vertexes, $\mathbf{V}^{(0)}$, $\mathbf{V}^{(1)}$, and $\mathbf{V}^{(2)}$. The points \mathbf{R} of the plane of the triangle are conveniently represented by using the so-called barycentric coordinates (η, μ) defined by

$$\mathbf{R} = \mathbf{V}^{(0)} + \eta(\mathbf{V}^{(1)} - \mathbf{V}^{(0)}) + \mu(\mathbf{V}^{(2)} - \mathbf{V}^{(0)}). \quad (\text{A.2})$$

Notice that the point \mathbf{R} is inside the triangle only when $\eta \geq 0$, $\mu \geq 0$, and $\eta + \mu \leq 1$. The intersection of the ray and the plane of the triangle is determined by the vector equation

$$\mathbf{r}(0) + s\hat{\mathbf{d}} = \mathbf{V}^{(0)} + \eta(\mathbf{V}^{(1)} - \mathbf{V}^{(0)}) + \mu(\mathbf{V}^{(2)} - \mathbf{V}^{(0)}). \quad (\text{A.3})$$

Rearranging terms we have

$$-s\hat{\mathbf{d}} + \eta(\mathbf{V}^{(1)} - \mathbf{V}^{(0)}) + \mu(\mathbf{V}^{(2)} - \mathbf{V}^{(0)}) = \mathbf{r}(0) - \mathbf{V}^{(0)} \quad (\text{A.4})$$

or, in matrix form,

$$\begin{pmatrix} -u & V_x^{(1)} - V_x^{(0)} & V_x^{(2)} - V_x^{(0)} \\ -v & V_y^{(1)} - V_y^{(0)} & V_y^{(2)} - V_y^{(0)} \\ -w & V_z^{(1)} - V_z^{(0)} & V_z^{(2)} - V_z^{(0)} \end{pmatrix} \begin{pmatrix} s \\ \eta \\ \mu \end{pmatrix} = \begin{pmatrix} x(0) - V_x^{(0)} \\ y(0) - V_y^{(0)} \\ z(0) - V_z^{(0)} \end{pmatrix}. \quad (\text{A.5})$$

The solution can be obtained by using Cramer's rule. It is worth noting that the determinant of the system,

$$\begin{aligned} D &\equiv \begin{vmatrix} -u & V_x^{(1)} - V_x^{(0)} & V_x^{(2)} - V_x^{(0)} \\ -v & V_y^{(1)} - V_y^{(0)} & V_y^{(2)} - V_y^{(0)} \\ -w & V_z^{(1)} - V_z^{(0)} & V_z^{(2)} - V_z^{(0)} \end{vmatrix} \\ &= -\hat{\mathbf{d}} \cdot \left[\left(\mathbf{V}^{(1)} - \mathbf{V}^{(0)} \right) \times \left(\mathbf{V}^{(2)} - \mathbf{V}^{(0)} \right) \right], \end{aligned} \quad (\text{A.6})$$

represents the volume of the parallelepiped formed by the sides of the triangle and the direction of motion of the particle. The vector

$$\mathbf{N} = \left(\mathbf{V}^{(1)} - \mathbf{V}^{(0)} \right) \times \left(\mathbf{V}^{(2)} - \mathbf{V}^{(0)} \right)$$

is perpendicular to the triangle. Evidently, the system of equations eq. (A.5) has a solution only when $D \neq 0$ or, equivalently, when the direction $\hat{\mathbf{d}}$ of the ray is not parallel to the triangle ($\hat{\mathbf{d}} \cdot \mathbf{N} \neq 0$).

To avoid unnecessary calculations, it is convenient to calculate first the distance s to the plane of the triangle, which is given by

$$s = \frac{1}{D} \begin{vmatrix} x(0) - V_x^{(0)} & V_x^{(1)} - V_x^{(0)} & V_x^{(2)} - V_x^{(0)} \\ y(0) - V_y^{(0)} & V_y^{(1)} - V_y^{(0)} & V_y^{(2)} - V_y^{(0)} \\ z(0) - V_z^{(0)} & V_z^{(1)} - V_z^{(0)} & V_z^{(2)} - V_z^{(0)} \end{vmatrix} = \frac{1}{D} \left(\mathbf{r}(0) - \mathbf{V}^{(0)} \right) \cdot \mathbf{N}. \quad (\text{A.7})$$

The particle may cross the triangle only when s is positive and less than the step length Δs .