Surface x-ray diffraction analysis using a genetic algorithm: the case of Sn/Cu(100)-$(3\sqrt{2} \times \sqrt{2})R45^\circ$

2009 J. Phys.: Condens. Matter 21 134011

(http://iopscience.iop.org/0953-8984/21/13/134011)

J. Phys.: Condens. Matter **21** (2009) 134011 (13pp) doi:10.1088/0953-8984/21/13/134011

Surface x-ray diffraction analysis using a genetic algorithm: the case of Sn/Cu(100)-$(3\sqrt{2} \times \sqrt{2})$R45°

J Martínez-Blanco[1], **V Joco**[1], **C Quirós**[2], **P Segovia**[1] **and E G Michel**[1]

[1] Departamento de Física de la Materia Condensada and Instituto Universitario de Ciencia de Materiales 'Nicolás Cabrera', Universidad Autónoma de Madrid, E-28049 Madrid, Spain
[2] Departamento de Física, Universidad de Oviedo–CINN, c/ Calvo Sotelo s/n, E-33007 Oviedo, Spain

E-mail: enrique.garcia.michel@uam.es

Received 3 November 2008, in final form 28 January 2009
Published 12 March 2009
Online at stacks.iop.org/JPhysCM/21/134011

Abstract
The application of genetic algorithms to the analysis of surface x-ray diffraction data is discussed and the implementation of a genetic algorithm of evolutionary type is described in detail. The structure of Sn/Cu(100)-$(3\sqrt{2} \times \sqrt{2})$R45° is determined on the basis of surface x-ray diffraction data analysed using this algorithm. The results are compared to previous findings using other techniques.

(Some figures in this article are in colour only in the electronic version)

1. Introduction

A precise knowledge of the crystalline structure of a surface is one of the most important aspects required to understand its properties. Besides the importance of the knowledge of the atomic distribution, it is a critical starting step for the understanding of the surface electronic structure and of many dynamical processes taking place at the surface. Due to this reason, many different techniques have been developed over the years to tackle the problem of surface structure determination [1]. SXRD is one of the most powerful techniques sensitive to long-range order. On the one hand, the x-ray scattering process is much simpler than electron scattering while, on the other hand, the success of XRD for the determination of bulk crystalline structures does not need to be explained. A generalized use of SXRD as a standard technique has been hindered by several difficulties. First, the acquisition of a complete dataset of SXRD data is more time consuming than for the bulk, due to both the lower signal and the fact that sample life is much more limited. Thus, the datasets are less accurate than in the bulk. Second, the analysis of SXRD data is, in general, made following a methodology developed over the years: the experimental dataset is compared to the prediction from a certain model and the model is modified until an acceptable fit is obtained. In most cases, this procedure is rather inefficient, as a good fit is reached only if the starting model is fairly close to the true structure, which is time-consuming and requires considerable doses of ability or information from other techniques. Indeed, if the surface structure is complex, with several different adsorption sites and a large unit cell, the process of fitting an SXRD dataset can be long and tedious. In the first part of this paper (section 2) we introduce the use of genetic algorithms for the analysis of SXRD data, taking into account its specificities, and compare their features with other methods commonly used. In the second part of the paper (section 3), we present the analysis of the surface structure of Sn/Cu(100)-$(3\sqrt{2} \times \sqrt{2})$R45° using these methods.

2. Application of genetic algorithms to the analysis of surface x-ray diffraction data

Determining a surface crystalline structure on the basis of a dataset acquired in an SXRD experiment is a typical example of an *inverse problem*, where the properties of a physical system must be deduced from their effects [2–4]. Often, we have access only to a limited number of these effects. In the case of x-ray diffraction, it is clear that the structure of the dispersing system (in our case a surface) could be exactly determined from an inverse Fourier transform, but the amplitude and phase of all structure factors for all dispersing angles need to be determined in the experiment. However, in a typical SXRD experiment we have access to the amplitude of only a few reflections of the infinity possible. Due to this limitation, we have to resort to an indirect procedure, even at the price of a certain ambiguity. This intrinsic limitation does not prevent us from obtaining a large amount of information and eventually a reliable structural model.

Once a dataset as large as possible has been measured in the experiment, the goal is to determine the surface structural model. An approach to this problem is the use of direct methods to solve the structure. We will not consider further this type of procedure and refer the reader to [5–8] for more information. A different approach consists in modifying the parameters of a certain structural model to fit the experimental data using a least squares procedure, in such a way that they represent the optimal set of atomic positions in agreement with the experimental data, within their accuracy. In this paper we describe a *differential evolution* method [9] to undertake the refinement of the atomic positions. This is a particular case of a *genetic algorithm* of *evolutionary* type. Genetic algorithms have been successfully used before in the refinement of a crystallographic model using dynamic LEED [10, 11], and also in several other problems related to the determination of crystalline structures [12–14]. We describe below in more detail how the particular genetic algorithm used works. Other algorithms frequently used to face this problem are the *Levenberg–Marquardt* method or the *simulated annealing* method [15].

Independently of the fitting method selected, the complexity of the problem, which involves the simultaneous optimization of many parameters, makes it necessary to restrict the number of free parameters in the starting model by using the maximum possible additional information. For instance, in general not all atomic positions are allowed within the unit cell, simply because of symmetry reasons. Patterson diagrams [3, 4, 16] are a powerful tool at this stage, as the Patterson diagram corresponding to a certain experimental dataset can be directly obtained from the in-plane part of the dataset. The Patterson function $P(x, y)$ is usually represented as a contour map or a colour scale in the real space unit cell, as shown in figure 1. As it is a correlation function, the most important information contained in it is how often a certain interatomic distance appears. A positive peak in point (x_0, y_0) means that two or more atoms of the structure are joined by the vector (x_0, y_0). Obviously the absolute maximum is always at (0, 0) (most frequent distance), as all atoms are at zero distance from themselves.

A structure containing n atoms has $\frac{1}{2}n(n-1)$ independent vectors in the Patterson diagram. This means that this strategy will already provide us with the interatomic distances in simple cases containing only a few atoms. In the case of more complex structures, it is useful both to discard possible models and also to inspire new structures to be introduced in the rigorous fitting process.

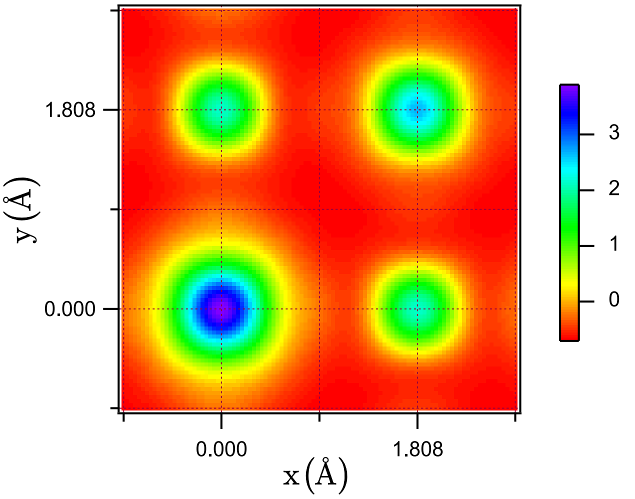


Figure 1. Patterson diagram of the Cu(100) surface calculated using in-plane structure factors for $l = 0.25$ in the surface unit cell area. The number of (h, k) pairs included in the calculation is 25.

2.1. Genetic algorithm: differential evolution

Once a reasonable starting model is defined, the difference between experimental data and the values calculated from the model is used to define an error function E. The model parameters are modified using any optimization process. The goal is to minimize the difference between both curves. The optimization process is repeated iteratively until the difference is small enough.

A first problem comes from the nonlinear nature of the error function. Besides the global minimum, there are many local minima for certain values of the parameters. Several methods have been used to tackle this problem [9]. From a historic point of view, the oldest strategy to overcome this difficulty was the *direct search* method, which can be described as a 'trial and error' procedure. The space of parameters is divided into a finite number of regions and the error function E is separately evaluated in each of them. A grid is constructed from these values, and the region corresponding to the smallest value of E is taken as the best fit. There are two elaborated variants of this method that use an algorithm to search the local minimum, namely the *Downhill Simplex* and the *Levenberg–Marquardt* method. In the *Downhill Simplex* the process starts after making an initial estimate for the values of the parameters. A geometric construction in the parameter space (called *simplex*) is displaced in the direction along which E decreases. The parameters providing the smallest E value in the environment of the initial estimate are taken as the best fit. The *Levenberg–Marquardt* method has been widely used. It also requires an initial estimate for the values of the parameters. The algorithm combines linearization and differential search to minimize E in the environment of the initial estimate. The parameters corresponding to the smallest E value in the environment of the initial estimate are taken as the best fit. A different approach is found in the *Monte Carlo method*, where the space of parameters is divided into small regions and some regions are randomly selected to evaluate the error function E in them. The algorithm is stopped after having checked a certain number of regions or when E becomes small enough. The region corresponding to the smallest E value corresponds to the best fit. In the *simulated annealing* method there is a non-zero probability of moving along a direction where E increases for each search step, so that the algorithm can escape from local minima. Finally, we mention also other more refined approaches (e.g. [17]).

Each of these strategies presents different problems when applied to SXRD. The number of parameters involved is often too large to use a *direct search* method, which becomes computationally unaffordable. On the other hand, *simplex* and *Levenberg–Marquardt* tend to be trapped in local minima, so that they are operational only if the initial estimate is very close to the final solution. In contrast the *Monte Carlo* and *simulated annealing* methods are not trapped in local minima, but they are rather inefficient to search the parameter space, as the process is made randomly, without considering the specific geometry of the error function.

We conclude than an efficient strategy to deal with this kind of problem should include a guided search with a fraction of random character. This is the case of *genetic algorithms*, which are inspired by the evolution of biological systems in nature [18] according to Darwin's theory of natural selection. Generation after generation, a population adapts itself to the environment by selecting the aptest individuals according to a certain criterion. The system evolves thanks to the recombination of the genes from the previous generation and random mutation in each generation. Mutations guarantee the diversity of the population, while gene recombination helps to concentrate the search in the most promising regions of the parameters space, as only the genes (parameters) which are closest to the experiment (these are the aptest individuals) are allowed to recombine for the next generation.

A detailed description of the fitting procedure used in this work can be found in [9]. It is a genetic algorithm of evolutionary class [19]. The main feature of this kind of algorithm is that they are intrinsically parallel, i.e. many regions of the parameter space are simultaneously searched. Let us assume that we have a set of N measurements of structure factors F_j^{obs} with $j = 1, 2, \ldots, N$. From our starting model, which contains n parameters to be fitted grouped together in the vector $\vec{p} = \{p_1, p_2, \ldots, p_n\}$, we can calculate the corresponding structure factors, $F_j^{\mathrm{calc}}(\vec{p})$. Experimental and calculated structure factors are compared using an error function $E(\vec{p})$. Guided by $E(\vec{p})$, the algorithm will try to optimize the vector of parameters $\vec{p}$ starting from an initial population of values generated randomly within a range for each parameter. Each individual of this population is called a *chromosome* and it is composed of n *genes*. The optimization is made by repeated cycles of mutation, recombination and selection of genes.

Figure 2 shows the detailed flux diagram for the differential evolution algorithm used. We start from an initial population of m vectors of parameters grouped in the matrix $P = [\vec{p}_0, \vec{p}_1, \ldots, \vec{p}_{m-1}]$. The size of the population m is chosen by the user. With a large population, the algorithm will inspect more options, but each iteration will be computationally more costly. The starting fitting parameters are introduced by the user and stored in vector $\vec{p}_0$, while the other $m - 1$ vectors are initialized assigning to each parameter a random value within its allowed range, which is also selected by the user. Once the initial population is defined, the error function E is evaluated for each chromosome (vector $\vec{p}_i$).

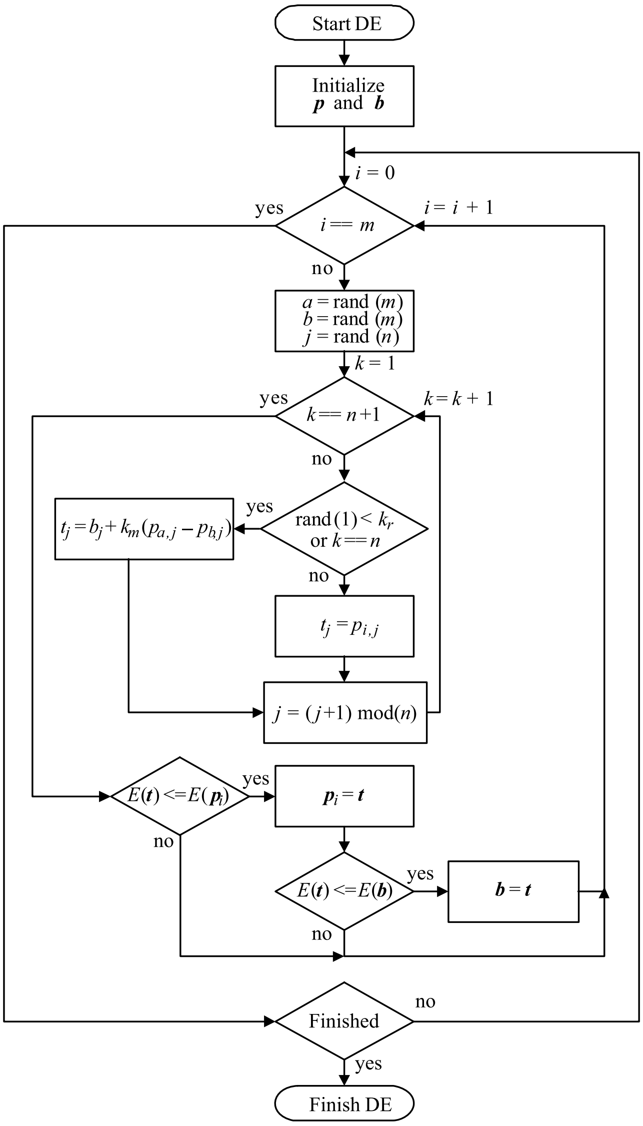


Figure 2. Flux diagram of the differential evolution algorithm. Reproduced with permission from [9]. Copyright 1999, the Royal Society.

After inspecting the result, the vector of parameters (chromosome) with minimum E is stored in vector $\vec{b} = \{b_1, b_2, \ldots, b_n\}$. This vector is updated as soon as an equal or better solution is found in the iteration process.

2.1.1. Mutation. In order to create a new generation of individuals, the algorithm selects two vectors randomly from the current population, $\vec{p}_a$ and $\vec{p}_b$, and its difference is used to mutate the best parameters vector found so far, according to

$$\vec{b}' = \vec{b} + k_m(\vec{p}_a - \vec{p}_b) \tag{1}$$

where k_m is the *mutation constant*, taking values in the 0–1 range. The constant should be estimated by the user in each case. A high value for k_m provides us with a large genetic diversity, so that there will be better chances to find a solution, but the convergence of the fitting process will be slower.

2.1.2. Recombination. As a next step the vector $\vec{t} = \{t_1, t_2, \ldots, t_n\}$ is constructed. This vector competes with $\vec{p}_0$, and it is obtained in the following way. Starting from the jth parameter (randomly selected), parameters t_j (modulus n) take values either from $\vec{b}'$ or from $\vec{p}_0$. A binomial distribution is used to decide which parameters are taken from $\vec{b}'$ and which from $\vec{p}_0$. Next, a random number between 0 and 1 is generated. This number is compared with the *recombination constant* k_r, which is chosen by the user. If the random number is less than or equal to k_r, t_j takes the value of the jth parameter of $\vec{b}'$. If the random number is larger than k_r, t_j takes the values of the jth parameter of $\vec{p}_0$. After repeating this process $n - 1$ times, finally the last parameter of $\vec{t}$ is taken from $\vec{b}'$, so that at least one parameter of $\vec{t}$ is different from the parameters of $\vec{p}_0$. If the value of any of the parameters t_j is outside the assigned ranges, which may happen depending on the value of $k_m(\vec{p}_a - \vec{p}_b)$ (see equation (1)), it would be replaced by a random value generated according to the expression

$$t_j = p_j^{\min} + \mathrm{rand}(p_j^{\max} - p_j^{\min})$$

where $p_j^{\min}$ and $p_j^{\max}$ are the extrema of the validity range of the jth parameter.

2.1.3. Selection. If vector $\vec{t}$ satisfies $E(\vec{t}) \leqslant E(\vec{p}_0)$, then $\vec{t}$ is selected to replace $\vec{p}_0$, otherwise, $\vec{p}_0$ survives until the next generation.

This procedure is repeated for all the individual of the population, i.e. for all the vectors of parameters of the matrix P, with a new vector $\vec{b}'$ calculated each time. Once the process ends, one generation is finished. The algorithm iterates as many generations as needed to get $E(\vec{b})$ small enough (ideally, equal to zero).

Finally, once the best set of parameters is obtained (defined as the set of parameters that best fits the experimental data), the accuracy of the result must be calculated. To this end we define and calculate the *local error* as the change Δp_j of parameter p_j that increases the error function $E(\vec{p})$ by a certain amount. In this work, this amount has been taken as 2.5% along both directions of the deviation. This value is reasonable for the usual size of the dataset in an SXRD experiment and the experimental accuracies reached, but it can be modified at will. If we calculate this way the local error for each parameter, it will be clear which parameters are well determined by the fitting process, and which parameters are not so well determined. An interesting application is to use this information to introduce new parameters in a certain structural model. For instance, if a certain atomic position is considered to be possible, a new parameter might be introduced in the model. Depending on how well the fitting process determined the new parameter, we can judge whether the new parameter is adequate or not.

2.2. Error function

There are several definitions for the error function. In general, each error function is better adapted to a certain type of problem. One common definition in surface crystallography is the *mean square error*, frequently called χ^2, defined as

$$\chi^2(\vec{p}_i) = \frac{1}{N-n}\sum_{j=1}^{N}\frac{(F_j^{\mathrm{obs}} - |F_j^{\mathrm{calc}}(\vec{p}_i)|)^2}{\sigma_j^2} \qquad (2)$$

where σ_j are the experimental errors associated with each structure factor. This definition has a statistical meaning when normalized to the difference between the number of experimental points and the number of free parameters, as in equation (2). In this case, as the difference between the experimental and calculated structure factors can be of the order of the experimental error, the right part of equation (2) is of the order of $N/(N-n)$. As N should be 5–10 times larger than n for a large enough experimental dataset, χ^2 approaches 1 from above for a fit whose quality is limited only by the experimental error. A value much larger than 1 means that the model is not adequate.

χ^2 values from different authors or corresponding to different experimental conditions must be compared with care, as χ^2 depends on the error bars of the measurements. It may happen for two datasets and their corresponding fitting models that the less accurate set has a smaller χ^2, giving the false impression of a better determined structure. In this case, a better model is characterized by smaller error bars in the parameters obtained from the fitting, because the error bars are larger for the less precise dataset. Furthermore, the experimental errors contain both the statistical dispersion of the data and systematic errors related to the measurement process, which are difficult to estimate. An overestimation of the errors may decrease the value of χ^2.

Different error functions free from these limitations (called *reliability factors* or *R factors*) are frequently used. One popular *R factor* is

$$R(\vec{p}_i) = \frac{\sum_{j=1}^{N}(F_j^{\mathrm{obs}} - |F_j^{\mathrm{calc}}(\vec{p}_i)|)^2}{\sum_{j=1}^{N}(F_j^{\mathrm{obs}})^2}. \qquad (3)$$

Its value should be smaller than 0.01 (1%) to accept a fit. As detailed below (see section 3.3), this definition is better adapted to our case. When the algorithm is guided by the R factor, it does not take into account the error bars (which in turn are frequently not realistic), and the optimization process ends with a fit whose χ^2 is, in general, better than the one reached when the algorithm is guided by the value of χ^2 itself.

2.3. Implementation of the algorithm: analysis software

One of the most popular codes for the fitting of surface x-ray diffraction data is ROD [15]. This is a free-access code and it is continuously improved within the project ANAROD[3] currently hosted at the European Synchrotron Radiation Facility (ESRF).

[3] URL: http://www.esrf.eu/computing/scientific/joint_projects/ANA-ROD/robots.htm

The algorithms implemented for data fitting using ROD are to the best of our knowledge only of two kinds: *Levenberg–Marquardt* and *simulated annealing*. Their features have been already described. On the other hand, this software presents a main limitation derived from the fact that its graphical interface is much below current standards. Both the data entry and the visualization of the results are made through a command console.

During the development of this work, we have constructed a user-friendlier interface, with the aim of implementing the genetic algorithm using the Igor Pro® environment. The code developed is called ORUGA (obtaining rods using genetic algorithm) and it is inspired by ROD code, so that all input files from ROD are compatible with ORUGA. All functions required to calculate the structure factors have been translated from the C programming language (which is the native language of ROD) into the Igor programming language. ORUGA software permits us to visualize in real time the effect on the fitting of any change in the parameters, which can be selected to remain fixed or to be fitted. The differential evolution algorithm itself is written as a procedure called GeneticOptimisation.ipf, developed by Nelson [20], which is published under a general public license of GNU (http://www.opensource.org/licenses/gpl-license.php).

3. Surface crystallography of Sn/Cu(100)-($3\sqrt{2} \times \sqrt{2}$)R45°

3.1. Previous structural studies

There are five different surface phases for submonolayers of Sn deposited on Cu(100) at room temperature [21–28]: $p(2 \times 2)$ at 0.21 ML; $p(2 \times 6)$ at 0.33 ML with two rotational domains; $\left(\begin{smallmatrix} -4 & 2 \\ 0 & 4 \end{smallmatrix}\right)$ (equivalent to $c(4 \times 8)$) at ∼0.40–0.45 ML; ($3\sqrt{2} \times \sqrt{2}$)R45° at 0.5 ML and $c(4 \times 4)$ at 0.65 ML.

A detailed analysis of the crystalline structure of the ($3\sqrt{2} \times \sqrt{2}$)R45° phase has been made using dynamic LEED by Pussi *et al* [28]. We summarize here its most important features and refer the reader to [28] for more details. The structure found is a substitutional surface alloy. In this model, Sn atoms in the last layer substitute Cu atoms occupying $c(2 \times 2)$ positions. The ($3\sqrt{2} \times \sqrt{2}$)R45° periodicity is due to formation of pairs of Sn atoms, which get closer to each other. The atomic row of Cu atoms among them is removed, so that the reconstruction is of the 'missing row' type. Additional support for this model has come from recent MEIS (medium energy ion scattering) results [29].

Sn/Cu(100) surface phases have received renewed attention since the discovery of a temperature-induced surface phase transition for the ($3\sqrt{2} \times \sqrt{2}$)R45° phase, which becomes ($\sqrt{2} \times \sqrt{2}$)R45° above 360 K [30, 31]. Similar phase transitions have been observed in the case of two surface phases of In/Cu(100) [32, 33] and for other surface phases of Sn/Cu(100) [24, 34]. The ground state of these surface phases has been interpreted as a surface charge density wave [30–33, 35, 36], but a different model has been proposed recently [37].

In this section we apply the genetic algorithm to the analysis of the surface structure of Sn/Cu(100)-($3\sqrt{2} \times \sqrt{2}$)R45° with a double objective. First, as this surface structure has been analysed in detail using dynamic LEED [28], the reliability of the genetic algorithm used can be directly tested. Second, this analysis is important to understand the properties of the surface phase transition from ($3\sqrt{2} \times \sqrt{2}$)R45° to ($\sqrt{2} \times \sqrt{2}$)R45° at 360 K [30], and in particular to describe the nature of the ($\sqrt{2} \times \sqrt{2}$)R45° high-temperature phase [38].

3.2. Experiment

The experiments were performed at the ID3 beamline of the ESRF. The sample was cleaned by repeated cycles of ion sputtering and annealing to 900 K in a base pressure better than 3×10^{-10} mbar. Surface cleanliness was checked by AES and the crystalline quality of the surface was measured from the terrace size, estimated from the width in reciprocal space of x-ray reflections [39]. From this analysis we estimate a mean terrace size for Cu(100) of ∼900 Å. Sn was deposited from a Knudsen cell heated up to 1100 K at rates in the range of 0.01–0.05 ML min^{-1}. During the deposition process it was possible to monitor the surface x-ray reflections from the sample, so that a very accurate coverage calibration was possible. A mean domain size of 480 Å was found for the ($3\sqrt{2} \times \sqrt{2}$)R45° phase.

3.3. Surface crystallography

3.3.1. Experimental data treatment. The experimental data were treated following standard procedures to obtain the normalized integrated intensity, taking into account the correction derived from the diffractometer geometry. This treatment was made using the code ANA from the ANAROD project (URL: http://www.esrf.eu/computing/scientific/joint_projects/ANA-ROD/robots.htm). The code AVE was used to obtain the non-equivalent reflections. AVE calculates an error bar σ_{hkl} for each structure factor, taking into account the statistical error σ_2 of each reflection and a quantity ϵ which is an average of all the agreement factors ϵ_{hkl}, each of them assigned to a set of equivalent reflections:

$$\sigma_{hkl} = \sqrt{\epsilon^2 F_{hkl}^2 + \sigma_2^2}. \qquad (4)$$

The error of the structure factor F_{hkl} takes into account the agreement factor between equivalent reflections and its own statistical error, which in general is very small. Systematic errors are estimated and included in the final error, as otherwise the total error calculated by AVE would be too small.

The full list of structure factors with their errors and the corresponding Miller indexes h, k and l are introduced in ORUGA to be used in the refinement of the structural model. The fitting process requires a dataset as large as possible and compatible and reliable structure factors.

Table 1. Estimate of the temperature factors at 300 K according to the Debye–Waller correction.

Atomic species	θ_D	B (Å^2)
Sn *in plane*	140	1.48
Sn *out of plane*	140	1.48
Cu (surf.) *in plane*	200	1.36
Cu (surf.) *out of plane*	200	1.36
Cu (bulk)	320	0.53

3.3.2. Temperature factors. It is well known that the diffraction intensity at finite temperatures is attenuated with respect to the value at zero temperature [2]. In order to make a comparison with experimental data possible and to take into account the effect of temperature, the theoretical atomic form factors have to be multiplied by an exponential factor (Debye–Waller factor). This effect becomes important for atomic form factors corresponding to a large momentum transfer.

ORUGA code allows the user to assign temperature factors associated to both the movement in plane and out of plane for all atoms in the proposed unit cell. The fitted parameter is directly $8\pi^2$ times the mean square displacement of the atom along the direction of the vibration at a certain temperature T: $8\pi^2\langle u_j^2\rangle \equiv B_j(T)$, in units of Å^2 [2–4]. A simple way to estimate initial values for the fit is through literature Debye temperatures θ_D for the bulk crystal or the surface. In the harmonic approximation of the dynamical theory of crystalline lattices and for θ_D much smaller than T, a second-order approximation for $B(T)$ gives a linear dependence with temperature [40, 41]:

$$B(T) \simeq \frac{24\pi^2\hbar^2}{Mk\theta_D^2}T = 11490\frac{T}{M\theta_D^2} \tag{5}$$

where the factor $24\pi^2\hbar^2/k$ (with k the Boltzmann constant) has been merged in the constant 11 490 in such a way that $B(T)$ is expressed in Å^2 if T and θ_D are introduced in Kelvin and M (the mass of the atom) in atomic mass units.

Taking into account that the data corresponding to the $(3\sqrt{2} \times \sqrt{2})$R45° phase were measured at room temperature ($T = 300$ K), and using literature values for θ_D (reference [42] for Cu and [28] for last-layer Sn), we can estimate the temperature factors for our model. Table 1 shows the estimated values for B, which are used as input for the fitting process. Note that in the case of bulk Cu the Debye temperature is larger than 300 K and thus the approximation made to calculate $B(T)$ would not be valid. However, the exact value is $B = 0.55$, almost identical to the approximate one.

3.3.3. Additional parameters of the algorithm. Besides the experimental dataset and a starting model, the differential evolution routine requires values for five additional parameters. The *number of generations* parameter fixes the maximum number of generations permitted. If the error function is improved from one generation to the next by a fraction smaller than the *tolerance* parameter, the fitting is finished. The *population size multiplier* parameter is an integer number that is multiplied by the number of free parameters to obtain the amount of possible values taken by a parameter within its validity range. For instance, if we have 20 free parameters, and a population size multiplier of 5 is used, the exploration will be made dividing the validity range of each parameter by $5 \times 20 = 100$. A too large value of this parameter means that a precision in the fitting beyond the accuracy of the experimental data is demanded[4]. The *mutation constant* and the *recombination constant* parameters take values between 0 and 1. As mentioned before, the *mutation constant* parameter is used to create genetic diversity, while the *recombination constant* parameter controls the weight to be given to the results from a generation to create the next one.

As the *mutation constant* and the *recombination constant* parameters are critical for the performance of the algorithm, they were estimated using a model of six atomic layers with 36 atoms in total, similar to the model used for the fit of the $(3\sqrt{2} \times \sqrt{2})$R45° phase and with the same number of fitted parameters. Using this model, the structure factors of a certain atomic configuration were calculated and these values were used as experimental data for a fitting series using different values of the *mutation constant* and the *recombination constant* parameters. The result can be observed in figure 3, where the temperature factors (related to the Debye–Waller factor) are taken as constants during this fit, and figure 4, where the fits are made including the temperature factors as free parameters.

These results suggest the following remarks, valid for a dataset of size and complexity similar to ours: the algorithm convergence is slower for a low recombination constant, while for a low mutation constant the algorithm converges to larger values of the error function. Thus, for a dataset of the complexity we are considering, it is convenient to use large constants of mutation and recombination. The fit of the trial model was driven first using χ^2, and afterwards using the R factor. Once the fit was finished, both error functions were calculated, as summarized in the table shown at the top of figures 3 and 4. From this study we conclude that, when the optimization is driven by the R factor, the process is more efficient and the final χ^2 value obtained is even better than when the fit is guided by the χ^2 itself. In view of the results shown in figure 4, the crystallographic models obtained below were fitted using the R factor as error function and a value of 0.75 for both the mutation and recombination constants.

3.3.4. Fit of the $(3\sqrt{2} \times \sqrt{2})$R45° structure. *Dataset*: As the Cu(100) substrate presents $p4mm$ symmetry, there are only four possible symmetry groups for the Sn reconstruction that are compatible with the observed LEED pattern [43]. From smaller to larger symmetry, these groups are: oblique $p1$; oblique $p2$; primitive rectangular pm; and primitive rectangular $p2mm$. The experimental dataset was averaged using the most restrictive group, $p2mm$, because the agreement factor in this case was low enough (see figure 5). This

[4] In principle, there is no direct relationship between the grid on which a parameter is varied and the total number of free parameters. However, for a certain number of free parameters (model complexity), the size of the experimental dataset determines the maximum possible accuracy that can be reached. In other words, there is no reason to probe a grid denser than the maximum sensitivity of the dataset.

Simulation WITHOUT temperature factors

		Error function: χ^2		Error function: R	
Mutation	Recomb.	χ^2	R (%)	χ^2	R (%)
0.25	0.25	1.34	3.6	2.32	2.2
0.5	0.5	0.83	4.2	0.96	0.8
0.75	0.75	1.15	2.7	0.98	0.9
0.25	0.75	2.76	23.4	2.59	2.2
0.75	0.25	1.0	3.3	1.13	1.0

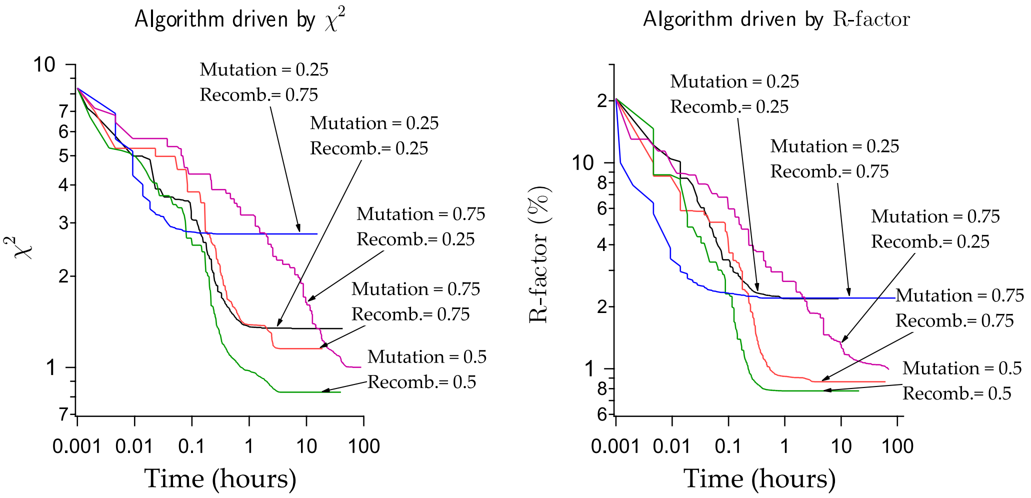


Figure 3. Results for the fitting simulation of a trial model without temperature factors. In a first series of fittings the algorithm was driven by χ^2. Later on, the same fittings were made using the R-factor. In both cases, the two error functions were calculated for the converged fitting.

Simulation WITH temperature factors

		Error function: χ^2		Error function: R	
Mutation	Recomb.	χ^2	R (%)	χ^2	R (%)
0.25	0.25	1.37	10.3	2.57	2.1
0.5	0.5	1.27	2.5	1.62	1.3
0.75	0.75	1.15	4.9	1.29	0.9
0.25	0.75	2.25	11.3	4.24	3.0
0.75	0.25	1.15	7.1	1.23	1.1

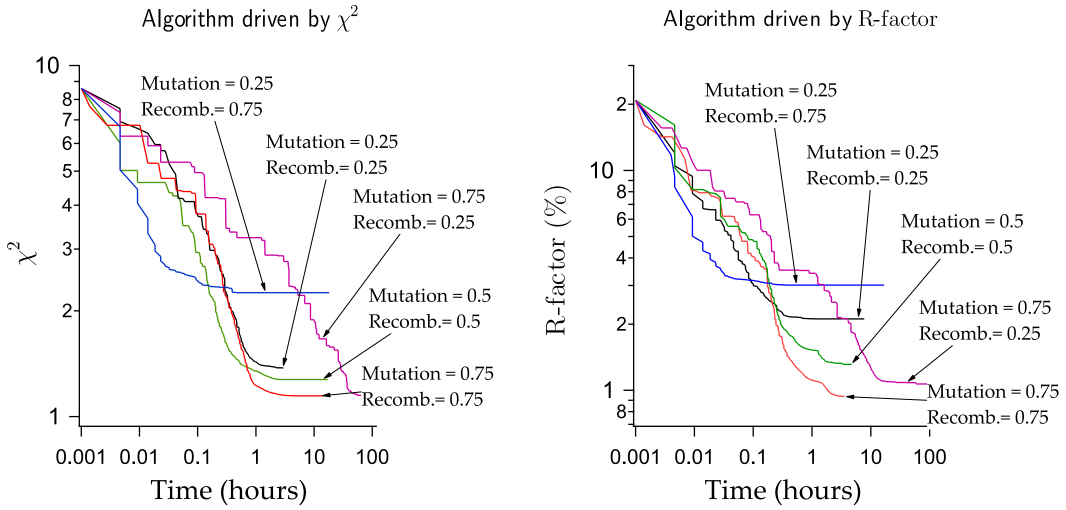
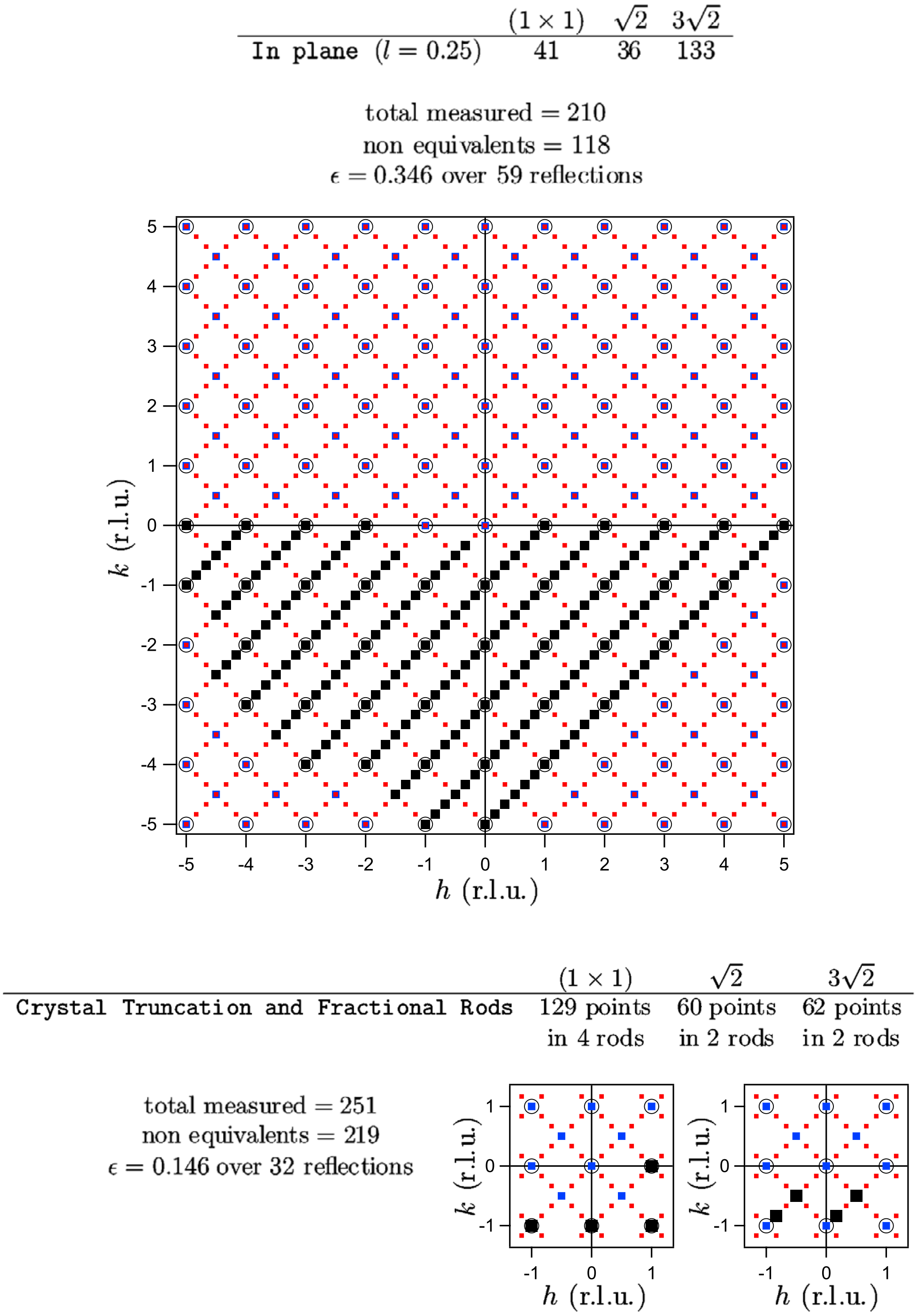


Figure 4. Results for the fitting simulation of a trial model with temperature factors. In a first series of fittings the algorithm was driven by χ^2. Later on, the same fittings were made using the R factor. In both cases, the two error functions were calculated for the converged fitting.



	(1×1)	$\sqrt{2}$	$3\sqrt{2}$
`In plane` $(l = 0.25)$	41	36	133

total measured = 210
non equivalents = 118
$\epsilon = 0.346$ over 59 reflections

	(1×1)	$\sqrt{2}$	$3\sqrt{2}$
`Crystal Truncation and Fractional Rods`	129 points in 4 rods	60 points in 2 rods	62 points in 2 rods

total measured = 251
non equivalents = 219
$\epsilon = 0.146$ over 32 reflections

Figure 5. Experimental points (filled squares) measured for the $(3\sqrt{2} \times \sqrt{2})$R45° phase. *Non-equivalent* points are obtained after considering the symmetry operations of group $p2mm$. The mean value of the agreement factor ϵ is also calculated and shown for points where more than one equivalent reflection was measured. The labels $3\sqrt{2}$ and $\sqrt{2}$ refer to points specific for the $(3\sqrt{2} \times \sqrt{2})$R45° and $(\sqrt{2} \times \sqrt{2})$R45° periodicities, respectively.

indicates that the dataset is compatible with this symmetry. Figure 5 shows all structure factors measured for the $(3\sqrt{2} \times \sqrt{2})$R45° phase, classified as in plane, fractional rods and crystal truncation rods.

In-plane structure factors were measured for a perpendicular momentum transfer of $l = 0.25$. The 210 measured experimental points were reduced to 118 non-equivalent structure factors using AVE code and applying the $p2mm$ symmetry group. The crystal truncation rods measured at $(h, k) = (1, 0)$ and $(0, -1)$ are equivalent, and thus there are three non-equivalent crystal truncation rods. Finally, there are two fractional rods specific of one of the surface reconstruction domains of $(3\sqrt{2} \times \sqrt{2})$R45° periodicity and two fractional rods (measured in $(\sqrt{2} \times \sqrt{2})$R45° positions) containing information from both domains.

Patterson diagram: Figure 6(a) shows the Patterson diagram obtained from the in-plane dataset.

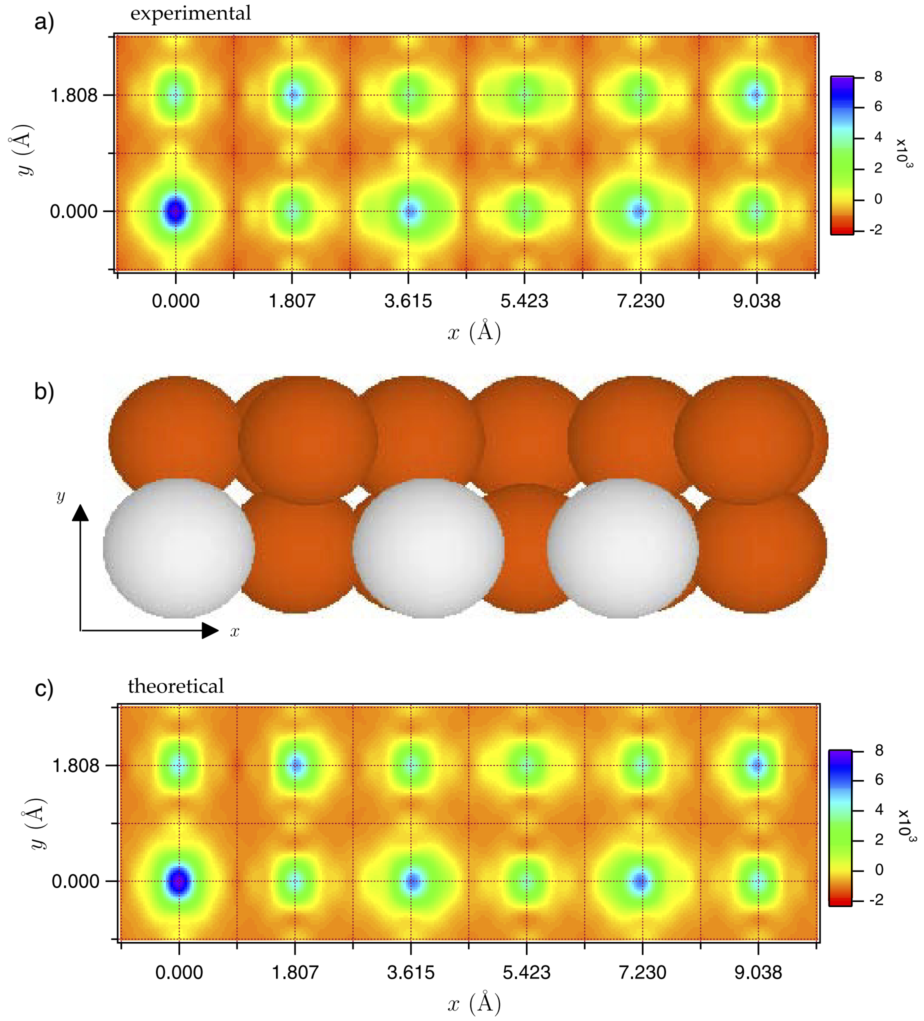


Figure 6. (a) Patterson diagram obtained from the in-plane structure factors set ($l = 0.25$) for the $(3\sqrt{2} \times \sqrt{2})R45°$ phase. (b) Top view of the surface unit cell obtained from the fit. (c) Theoretical Patterson diagram calculated using the atomic positions shown in panel (b).

A simple inspection of the Patterson diagram helps to limit the possible starting models for the fit. For instance, the overlayer model proposed in [22] can be discarded, as the vectors (3.66,0) and (7.18,0) observed in the Patterson diagram (corresponding to the maxima found near the nominal positions (3.6150) and (7.2300), respectively), are not found in the unit cell proposed in that model. Other models proposed in the literature [27, 28] are in principle compatible with our experimental Patterson diagram, and thus the starting crystallographic model will be constructed taking into account their features.

Structural model: As a first step, a model identical to the overlayer model of [27] was constructed, but introducing occupation parameters associated with the six atoms in the last two atomic layers. The occupation parameters take values between 0 and 1, so that a value equal to 0 is equivalent to saying that the corresponding atom is missing. By doing this, we consider at the same time the models shown in figures 1(B) and (C) and figure 2 from [28]. Furthermore, the possible lack of any Cu atom in the last atomic layer is also considered. The rest of the parameters used in the model are explained in figure 7, where each atom has a parameter for its vertical displacement. Atoms marked with arrows also have a parameter related to horizontal displacement along the direction permitted by the symmetry group $p2mm$.

The model described above was taken as the starting point for the ORUGA code, used to fit the experimental data. In-plane data are mostly sensitive to the lateral atomic displacements and atomic positions in the unit cell, as shown upon constructing the Patterson diagram. Thus, as a first step, only the in-plane dataset was considered, in an effort to determine the occupation parameters of the atomic sites in the last two atomic layers. All structural parameters of the model were left free, including the occupation parameters. The differential evolution algorithm required only 60 generations (~1 min) to converge all occupation parameters to a value very close to 1, with the exception of the last-layer Cu atom located at the centre of the unit cell, which was very close to 0 (0.016).

Thus, after this first fit we conclude that the lack of the last-layer intermediate Cu atom is strongly determined by the experimental in-plane structure factors. On the basis of this information, a final structural model was constructed and

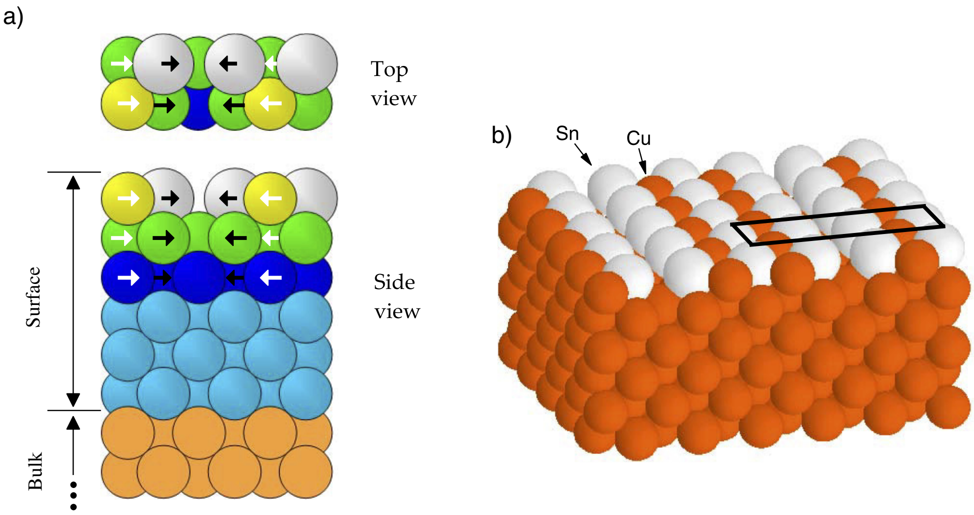


Figure 7. (a) Top and side view of the unit cell introduced in the fitting code, after discarding the presence of the last-layer intermediate Cu atom. The unit cell contains 6 layers. All atoms have a parameter related to their vertical displacement. Atoms marked with arrows also have a parameter related to their horizontal displacement along the direction permitted by the symmetry group $p2mm$. (b) 3D perspective of the unit cell repeated several times. A rectangle highlights the surface unit cell.

refined using the whole dataset (including in-plane data), but keeping the occupation factors constant. Obviously, we have now only two Cu atoms in the last layer, and thus the model is of *alloy* type, with Sn and Cu atoms in the last layer, and in agreement with the model proposed in [28]. In summary, the model has 35 atoms in the unit cell and 42 free parameters. As the dataset comprises a total number of 337 non-equivalent structure factors (see figure 5), we have a ratio of approx. ∼8 structure factors/parameter). The parameters used are:

- 3 non-structural, related to scale factors and the surface roughness factor β [44].
- 29 atomic displacements, out of which 6 correspond to displacements along the x axis of atoms from the three last layers, and 23 to displacements along the z axis.
- 10 temperature parameters (5 in the plane and 5 out of the plane). Each of the three non-equivalent atoms in the last layer has one in-plane and one out-of-plane independent temperature parameters. The same temperature parameter was assigned to the atoms in the last but one layer. Finally, a last temperature parameter was assigned to the atoms in the other four layers of the model.

Refinement of the structural model: At this point, the refinement of the atomic positions and temperature factors of the structural model was made in a few steps, using the whole set of experimental data (integer and fractional rods and in-plane data). The sequence was as follows:

(1) The temperature factors associated with the atoms in the surface unit cell were kept fixed at the values estimated in table 1. These estimates give an idea of the order of magnitude. It is not critical at this step if the temperature factors are not accurate, as the Debye–Waller correction is small, and we are rather interested in the right atomic positions to search later on around them for the final solution. ORUGA code permits us to associate an isotropic temperature factor to the bulk unit cell. At variance with the surface temperature factors, this parameter cannot be fitted. A value of 0.53 Å^2 was chosen, also according to the estimate in table 1, and this value was maintained in the rest of the analysis. The rest of the structural parameters were left free, with starting atomic displacements equal to zero, i.e. we start from the nominal atomic positions. The size of the vertical allowed variation range was chosen equal to 0.2 Å. For horizontal movements, ranges of 0.5 Å were chosen for last-layer atoms, 0.15 Å for the last but one layer and 0.075 Å for the last but two layer. These ranges are large enough to contain positions compatible with the interatomic distances found in the experimental Patterson diagram.

(2) During this fit, the lateral atomic displacements rapidly converged towards values at the centre of the unit cell, while last-layer Sn and Cu atoms' vertical positions tended to higher values (towards the vacuum side), especially in the case of Sn and most evidently in the unpaired Sn atom. This configuration is required to reproduce the profile found for the crystal truncation rods, which contain mainly information on the registry of the surface with respect to deeper layers. The maximum allowed variation range had to be corrected several times to allow for larger values of the vertical displacement. In all cases the algorithm was restarted from the same initial positions, and converged into a reproducible result, which is a robustness indication for the algorithm and also of a good ability to reach the minimum. The R-factor value at this point was 0.8%.

(3) Finally, starting from the previous result, all parameters were left free, including the temperature parameters. Each of the ten temperature parameters was allowed to vary between 0 and 4 Å^2. After including this temperature correction the atomic positions were slightly modified and a final value of the R factor of 0.36% was reached.

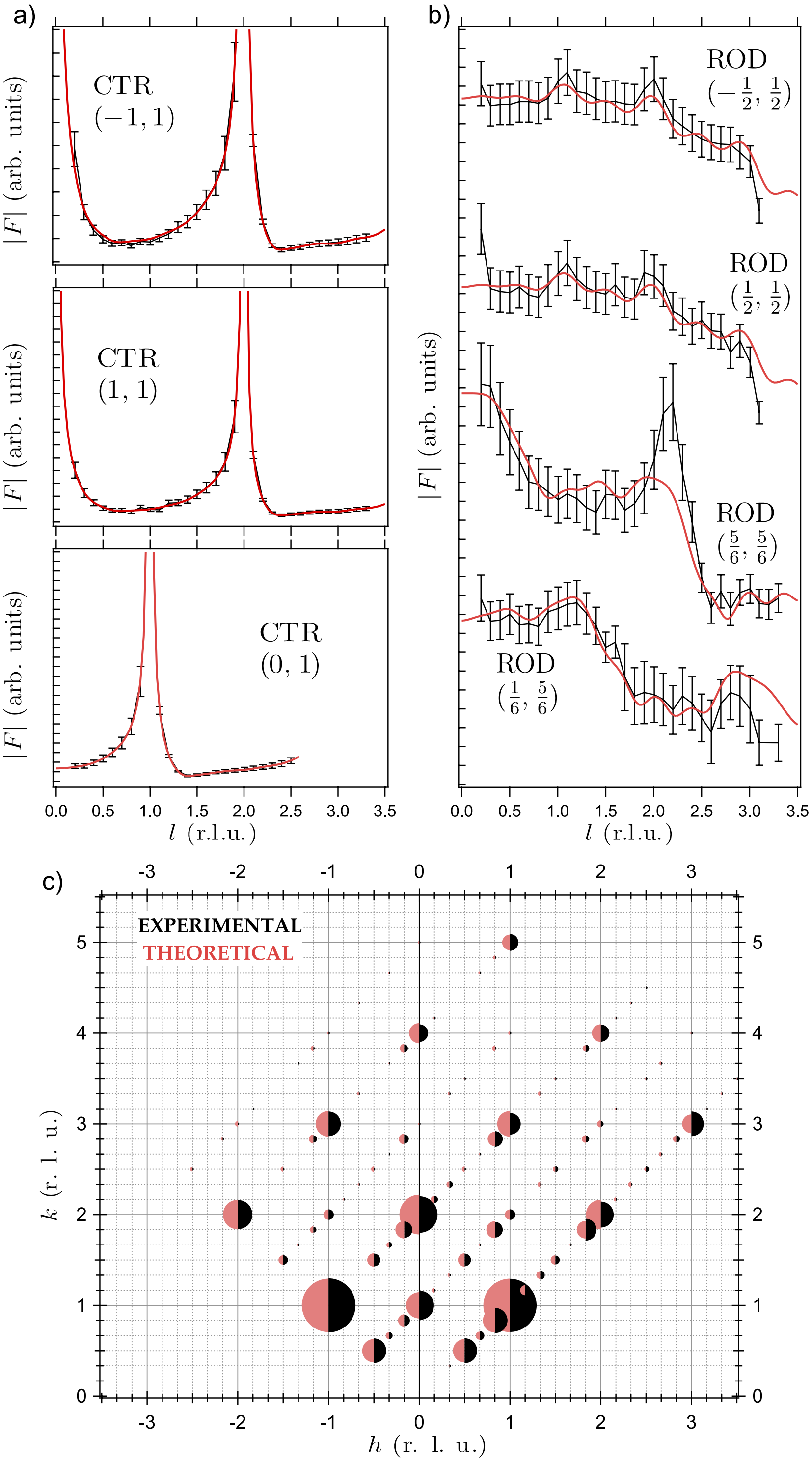
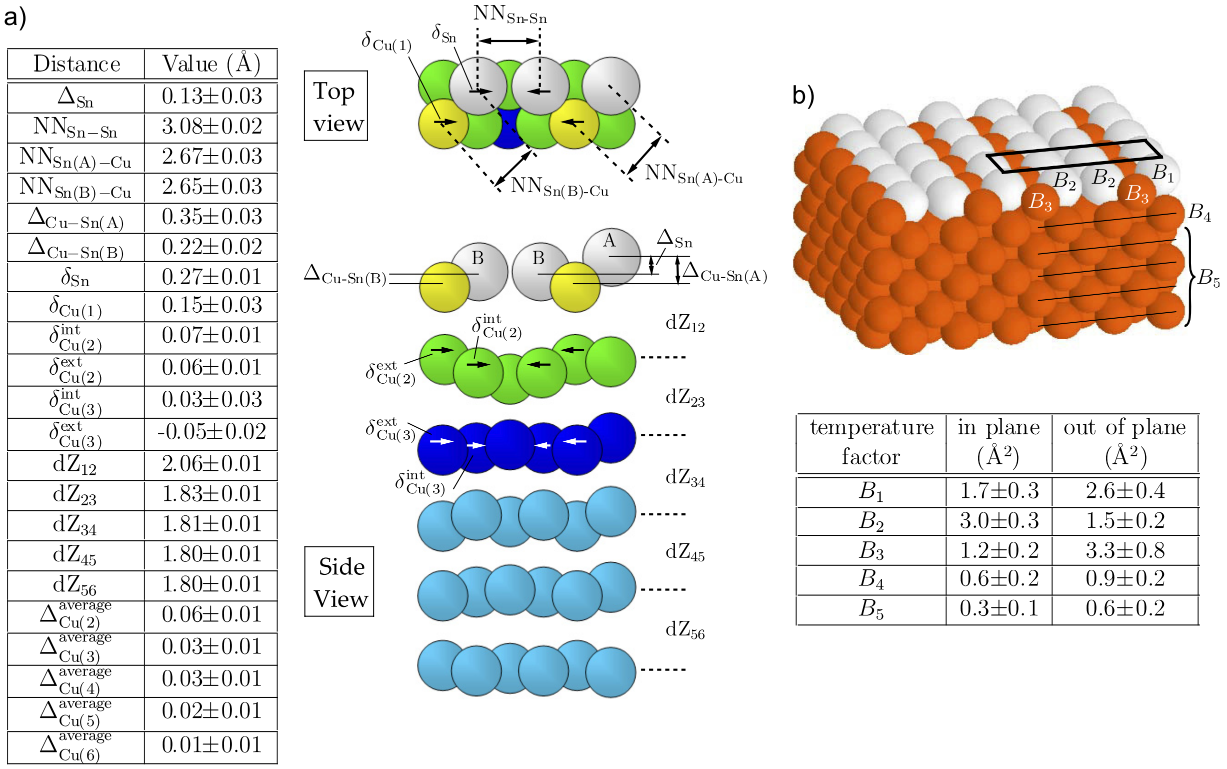


Figure 8. (a) Non-equivalent crystal truncation rods (CTR), measured for the $(3\sqrt{2} \times \sqrt{2})$R45° phase. (b) Fractional rods. Experimental points (joined with black lines) are shown with their error bars. The red (light gray) curve corresponds to the calculation made on the basis of the fitted structural model. (c) In-plane non-equivalent points (for $l = 0.25$). The radius of each half-circle is proportional to the measured structure factor (black) and to the structure factor calculated with the fitted structural model (red/light gray).

Figure 8 shows the full dataset used for the fitting, and the comparison of the structure factors calculated according to the optimized fitted model. The agreement is excellent, especially for the integer rods (CTRs) and the in-plane data. The fractional rod measured at (5/6, 5/6) contains four experimental points which are less well reproduced by the fitted model. This small discrepancy could not be solved, but it represents a minor fraction of the full set of structure factors.

In figure 6(c) we show the Patterson diagram constructed using the structure factors calculated for the optimized model.



Distance	Value (Å)
Δ_{Sn}	0.13±0.03
NN_{Sn-Sn}	3.08±0.02
$NN_{Sn(A)-Cu}$	2.67±0.03
$NN_{Sn(B)-Cu}$	2.65±0.03
$\Delta_{Cu-Sn(A)}$	0.35±0.03
$\Delta_{Cu-Sn(B)}$	0.22±0.02
δ_{Sn}	0.27±0.01
$\delta_{Cu(1)}$	0.15±0.03
$\delta^{int}_{Cu(2)}$	0.07±0.01
$\delta^{ext}_{Cu(2)}$	0.06±0.01
$\delta^{int}_{Cu(3)}$	0.03±0.03
$\delta^{ext}_{Cu(3)}$	-0.05±0.02
dZ_{12}	2.06±0.01
dZ_{23}	1.83±0.01
dZ_{34}	1.81±0.01
dZ_{45}	1.80±0.01
dZ_{56}	1.80±0.01
$\Delta^{average}_{Cu(2)}$	0.06±0.01
$\Delta^{average}_{Cu(3)}$	0.03±0.01
$\Delta^{average}_{Cu(4)}$	0.03±0.01
$\Delta^{average}_{Cu(5)}$	0.02±0.01
$\Delta^{average}_{Cu(6)}$	0.01±0.01

temperature factor	in plane (Å^2)	out of plane (Å^2)
B_1	1.7±0.3	2.6±0.4
B_2	3.0±0.3	1.5±0.2
B_3	1.2±0.2	3.3±0.8
B_4	0.6±0.2	0.9±0.2
B_5	0.3±0.1	0.6±0.2

Figure 9. (a) Main distances in Å obtained from the fit of the $(3\sqrt{2} \times \sqrt{2})$R45° phase. The atomic positions are represented on the right in an exaggerated scale to better show the relative positions within each layer. Distances between layers are referred to the centre of mass of each layer. (b) Three-dimensional perspective view of the unit cell repeated ten times, using the real positions obtained from the fit. A rectangle marks the surface unit cell. Different temperature factors (one in-plane and another out-of-plane) are assigned to the three last-layer non-equivalent atoms. Their values can be seen in the table below. A single temperature factor was assigned to all last but one atomic layers (B_4) and a last factor for the other four layers (B_5).

Also in this case the agreement with the experimental Patterson diagram (panel (a) in this figure) is satisfactory.

Description of the structural model: Figure 9 shows the final configuration of the structural model, including the values of the main atomic distances calculated from the fitted parameters. The most important features of the model are the following:

- There is a significant modification of the atomic positions in the two last atomic layers, as expected from the large difference in atomic radii between Cu and Sn (Sn is 10% larger in the bulk). Lateral relaxations and the last-layer corrugation compensate the strain induced by the difference in atomic radii.
- The distance between the last two Cu layers is 7% larger than the nominal bulk value. This value is larger than the 4% found by dynamic LEED in [28].
- Every third Sn atomic row lies 0.13 Å above the other two, which are in turn paired, reducing their distance by 0.54 Å. Even after the pairing, the minimum distance between Sn atoms is 10% larger than the interatomic distance in bulk Sn (2.81 Å).
- The minimum Sn–Cu interatomic distance found is

$$\sqrt{(NN_{Sn(B)-Cu})^2 + (\Delta_{Cu-Sn(B)})^2} = 2.66\ \text{Å}$$

which corresponds to an effective Sn radius of 1.37 Å. This value is 3% smaller than the effective bulk Sn radius (1.41 Å). The value obtained in [28] from dynamic LEED was 7%.

- The lack of the central last-layer Cu atom produced a lateral relaxation of Sn atoms of 0.27 Åand, to a lesser extent, for the other Cu atoms as well (0.15 Å). This relaxation towards the centre of the unit cell is also observed in the last but one and last but two layers, although with smaller values.
- The distances between layers (dZ_{ij}) shown in the table of figure 9(a) are calculated between the centres of mass of each layer. As the bulk nominal value is 1.807 Å, there is a relevant expansion in the last layer (2.06 Å with respect to the centre of mass of the last layer and 1.94 Å with respect to the centre of mass of the Cu atoms in the last layer); and in the last but one layer (1.83 Å). The modulation of the atomic positions within each layer (Δ^{mean}_{Cu}) is also calculated with respect to the centres of mass, and becomes smaller for deeper layers, as expected.
- The total temperature factor (sum of the in-plane and out-of-plane factors) decreases for deeper layers. Its value is approximately 4.5 Å^2 for atoms in the last layer, 1.5 Å^2 in the last but one layer and 0.9 Å^2 in the following layers, before reaching the bulk, which has an isotropic temperature factor of 0.53 Å^2. The paired Sn atoms

have an in-plane temperature factor twice as large as out-of-plane, while the opposite is true for last-layer Cu atoms. The highest Sn atom also has a higher out-of-plane temperature factor.

4. Conclusions

If we compare the structural parameters found with the results from [28] using dynamic LEED, we find a good agreement (and sometimes an almost exact coincidence) with most of the distances obtained in this work from SXRD and using the differential evolution algorithm for the optimization of the fit. As there are no precedents in the use of this algorithm for the crystallographic analysis using SXRD, these findings support the validity of the method, which is found to be a promising and powerful tool in the analysis of this kind of problem. It is able to extract crystallographic information in a fast and reliable way, in spite of the large size of the unit cell and the significant number of parameters fitted. A general comparison with other methods is difficult, as only the Levenberg–Marquardt and the *simulated annealing* method [15] are actually implemented in a fitting code of widespread use. While these methods are useful for the refinement of a model, the genetic algorithm appears as a much powerful and fast method to discriminate between possible models and to find the best structure for a certain model, i.e. in what concerns the uniqueness and reliability of the model found, mainly due to the much broader range of structures probed in a systematic way.

Acknowledgments

We thank X Torrelles and J A Martin-Gago for fruitful discussions. Financial support by the Spanish MICINN (FIS2005-00747 and FIS2007-29085-E) and by CAM (S-0505/PPQ/0316) is gratefully acknowledged. We thank the ESRF staff for help and support. CQ acknowledges support from the Spanish Government and European Social Fund under 'Ramón y Cajal' Programme.

References

[1] Unertl W N (ed) 1996 *Handbook of Surface Science vol 1: Physical Structure* (Amsterdam: Elsevier)

[2] Warren B E 1969 *X-Ray Diffraction* (Reading, MA: Addison-Wesley)

[3] Feidenhans'l R 1989 *Surf. Sci. Rep.* **10** 105

[4] Robinson I K and Tweet D J 1992 *Rep. Prog. Phys.* **55** 599

[5] Torrelles X, Rius J, Boscherini F, Heun S, Mueller B H, Ferrer S, Alvarez J and Miratvilles C 1998 *Phys. Rev.* B **57** 4281

[6] Torrelles X, Miratvilles C and Ferrer S 1999 *Surf. Sci.* **423** 338

[7] Marks L D, Erdman N and Subramanian A 2001 *J. Phys.: Condens. Matter* **13** 10677

[8] Saldin D K and Shneerson V L 2008 *J. Phys.: Condens. Matter* **20** 304208

[9] Wormington M, Panaccione C, Matney K M and Bowen D K 1999 *Phil. Trans. R. Soc.* A **357** 2827

[10] Van Hove M A and Döll R 1996 *Surf. Sci.* **355** 393

[11] Mitsui T, Hongo S and Urano T 2001 *Surf. Sci.* **482** 1451

[12] Shenoy V B, Wang C Z, Chuang F C, Ciobanu C V and Ho K M 2004 *Surf. Sci.* **573** L375

[13] Briggs R M and Ciobanu C V 2007 *Phys. Rev.* B **75** 195415

[14] Trimarchi G and Zunger A 2007 *Phys. Rev.* B **75** 104113

[15] Vlieg E 2000 *J. Appl. Crystallogr.* **33** 401

[16] Patterson A L 1934 *Phys. Rev.* **46** 372

[17] Blanco-Rey M and de Andrés P 2006 *Surf. Sci.* **600** L91

[18] Holland J H 1975 *Adaptation in Natural and Artificial Systems, an Introductory Analysis with Application to Biology, Control and Artificial Intelligence* (Ann Arbor, MI: The University of Michigan Press)

[19] Storn R and Price K V *Technical Report TR-95-012, ICSI 1995* (URL: ftp://ftp.icsi.berkeley.edu/pub/techreports/1995/tr-95-012.pdf)

[20] Nelson A 2006 *J. Appl. Crystallogr.* **39** 273

[21] Rhead G E, Argile C and Barthes M G 1981 *Surf. Interface Anal.* **3** 165

[22] Argile C and Rhead G E 1982 *Thin Solid Films* **87** 265

[23] Argile C and Rhead G E 1983 *Surf. Sci.* **135** 18

[24] Martínez-Blanco J, Joco V, Segovia P, Balasubramanian T and Michel E G 2006 *Appl. Surf. Sci.* **252** 5331

[25] Nara Y, Yaji K, Iimori T, Nakatsuji K and Komori F 2007 *Surf. Sci.* **601** 5170

[26] Lallo J, Goncharova L V, Hinch B J, Rangan S, Bartynski R A and Strongin D R 2008 *Surf. Sci.* **602** 2348

[27] McLoughlin E, Cafolla A A, AlShamaileh E and Barnes C J 2001 *Surf. Sci.* **482–485** 1431

[28] Pussi K, AlShamaileh E, McLoughlin E, Cafolla A A and Lindroos M 2004 *Surf. Sci.* **549** 24

[29] Brown M G, Walker M, McConville C F, Noakes T C Q, Bailey P and Draxler M 2006 *24th European Conf. in Surface Sci.* unpublished

[30] Martínez-Blanco J, Joco V, Ascolani H, Tejeda A, Quirós C, Panaccione G, Balasubramanian T, Segovia P and Michel E G 2005 *Phys. Rev.* B **72** 041401(R)

[31] Martínez-Blanco J, Joco V, Fujii J, Segovia P and Michel E G 2008 *Phys. Rev.* B **77** 195418

[32] Nakagawa T, Boishin G I, Fujioka H, Yeom H W, Matsuda I, Takagi N, Nishijima M and Aruga T 2001 *Phys. Rev. Lett.* **86** 854

[33] Nakagawa T, Okuyama H, Nishijima M, Aruga T, Yeom H W, Rotenberg E, Krenzer B and Kevan S D 2003 *Phys. Rev.* B **67** 241401

[34] Martínez-Blanco J, Joco V, Fujii J, Segovia P and Michel E G 2009 *J. Phys.: Condens. Matter* **21** 055001

[35] Aruga T 2002 *J. Phys.: Condens. Matter* **14** 8393

[36] Aruga T 2006 *Surf. Sci. Rep.* **61** 283

[37] Yaji K, Nara Y, Nakatsuji K, Iimori T, Yagyu K, Nakayama R, Nemoto N and Komori F 2008 *Phys. Rev.* B **78** 035427

[38] Martínez-Blanco J *et al* 2009 *Phys. Rev. Lett.* submitted

[39] Dutta P and Sinha S K 1981 *Phys. Rev. Lett.* **47** 50

[40] Debye P 1914 *Ann. Phys.* **43** 49

[41] Waller I 1923 *Z. Phys.* A **17** 398

[42] Reid R J 1972 *Surf. Sci.* **29** 623

[43] Henry N F M and Lonsdale K 1965 *International Tables for X-Ray Crystallography* vol I–IV (Birmingham: Kynoch)

[44] Robinson I K 1986 *Phys. Rev.* B **33** 3830