Theory and implementation behind: Universal surface creation - smallest unitcell

Bjarke Brink Buus, Jakob Howalt & Thomas Bligaard

April 13, 2015

1 Construction of surface slabs

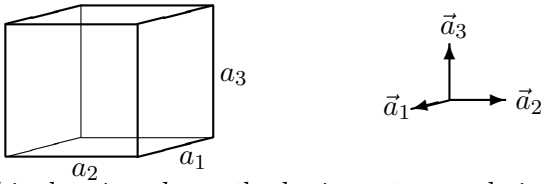
The aim for this part of the project is to create all possible surfaces with a given Miller Indice and the conventional bulk structure. In this section, the theory behind the construction of surface slabs will be outlined and from this fundament an implementation has been developed in Python. This implementation, will be able to create any surface for any type of structure including following common bulk structures - the simple cubic unit cell, the body centered cubic unit cell, the face centered cubic unit cell and the hexagonal close packed unit cell.

1.1 Theory

By introducing both the real and the reciprocal lattice spaces most pieces of the puzzle of creating any surface is derived. In addition some integer mathmatics will be used.

1.1.1 Real lattice space

First, we will start by defining the system in real space. We have three basis vectors that span the crystal lattice in the conventional unit cell $(\vec{a}_1, \vec{a}_2, \vec{a}_3)$. These three vectors do not have to be orthogonal and the procedure will therefore also work for hcp structures. Additionally, the lengths of the vectors will not in all cases be the same, so the theoretical approach to this problem, will involve three independent lengths. For most bulk structures there will only be one or two different lattice constants determining the unit cell due to symmetry, for instance the L10 and L12 alloys as mentioned in section XXX method XXX. The unit cell can be seen in drawing 1 with the lengths and directions.



Drawing 1: *This drawing shows the basis vectors and sizes for the system.*

A surface is defined by its Miller Indice (h,k,l), where h, k and l all are integers, which in real space can be described by the crystal planes that are parallel to the plane that intersects the basis vectors $(\vec{a}_1, \vec{a}_2, \vec{a}_3)$ at

$$\frac{1}{h}\vec{a}_1,\ \frac{1}{k}\vec{a}_2,\ \frac{1}{l}\vec{a}_3.$$

The Miller Indices are used for all four types of structures sc, bcc, fcc and hcp. In case of one or more of the Miller Indice (h,k,l) is zero, the plane does not intersect with the corresponding axis. For instance, if k is equal to zero, the normal vector to the plane, defined by (h,0,l), will be orthogonal to $\vec{a}_2$. A (0,0,0) Miller Indice is unphysical and hence will not be included in the theory section, however a part of the implementation code will notify the user that the chosen surface is not possible to create.

1.1.2 Reciprocal lattice space

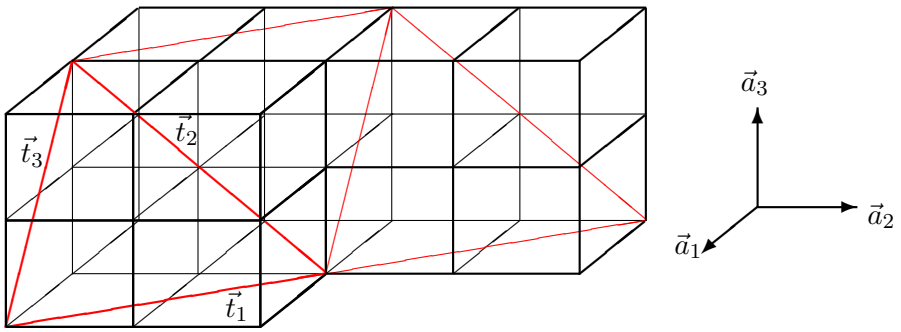
For the full understanding of the lattice construction, it is very usefull to introduce the reciprocal vector space. The basis vectors in the reciprocal lattice space are given by,

$$\vec{b}_1 = \frac{\vec{a}_2 \times \vec{a}_3}{\vec{a}_1 \cdot (\vec{a}_2 \times \vec{a}_3)},\ \vec{b}_2 = \frac{\vec{a}_3 \times \vec{a}_1}{\vec{a}_2 \cdot (\vec{a}_3 \times \vec{a}_1)},\ \vec{b}_3 = \frac{\vec{a}_1 \times \vec{a}_2}{\vec{a}_3 \cdot (\vec{a}_1 \times \vec{a}_2)} \tag{1}$$

$$\vec{a}_i \cdot \vec{b}_j = \delta_{ij} \tag{2}$$

When introducing the reciprocal lattice vectors, the normal vector for a given surface plane with the Miller Indices (hkl) is given by

$$\vec{n} = h\vec{b}_1 + k\vec{b}_2 + l\vec{b}_3 \tag{3}$$



Drawing 2: *This drawing shows a surface with Miller Indices (2,1,1). The repetition of a lattice point is shown, and the vectors spanning this surface can be found.*

Furthermore, a desired surface with a normal vector $\vec{n}$, drawing 2 shows that for a set of non-zero Miller indices, three vectors, will be noted $\vec{t}_{1,2,3}$, in the plane can easily be found. Two vectors, linearly independent ofcourse, created by a linear combination of the vectors $\vec{t}_{1,2,3}$, can span the desired surface. These vectors are given by

$$\vec{t_1} = -\frac{1}{h}\vec{a}_1 + \frac{1}{k}\vec{a}_2, \vec{t_2} = \frac{1}{k}\vec{a}_2 - \frac{1}{l}\vec{a}_3, \vec{t_3} = \frac{1}{h}\vec{a}_1 - \frac{1}{l}\vec{a}_3$$

but it should be noted that the anti-parallel versions of $\vec{t}_{1,2,3}$ also can be used. Since h, k and l all are integers, we can multiply with the product hkl and divide with the indice, which will be common for each of the lattice vectors, $\vec{a}_i$, with respect to both of the lattice vectors $\vec{a}_i$, so a new set of vectors end up being,

$$\vec{t_1} = k\vec{a}_1 - h\vec{a}_2,\ \vec{t_2} = l\vec{a}_1 - h\vec{a}_3,\ \vec{t_3} = l\vec{a}_2 - k\vec{a}_3 \tag{4}$$

For the special cases of two of the Miller Indices being zero, it is a very straightforward, to see that the appropiate vectors to span the normal vector, will be the corresponding basis vectors in real space. If for instance, h and k both are zero, it will result in a choice of $\vec{v}_1 = \vec{a}_1$ and $\vec{v}_2 = \vec{a}_2$.

1.1.3 Determination of the two surface vectors

Having introduced the real space and the reciprocal space, most of the theory is available and hence the determination of the two vectors that span the surface with respect to a given Miller Indice is possible.

The simple lattice points $r_{i,j,m}$ are placed at $\vec{r}_{i,j,m} = i\vec{a}_1 + j\vec{a}_2 + k\vec{a}_3$ where i, j, m all are integers. Because of the arrangement of the lattice points, not all surface planes will go through these points. The dot product between the normal vector $\vec{n}$ and the lattice points $\vec{r}_{i,j,m}$ gives,

$$hi + kj + lm = d$$

and since all constants are integers, d must also be an integer and therefore the values of d has been quantized. This equation is a 'Linear Diophantine Equation' and the smallest d for which there exist an non-zero solution (i, j, m) when (h, k, l) are non-zero is, accordingly to 'Bezouts Identity', when d is the smallest common divisor of h, k, and l. If one or two of the Miller Indices is zero, the identity is true, but only when choosing the largest common divisor for the non-zero parts of (h, k, l). If a Miller indice has a common divisor $e > 1$, the non-zero components of the Miller Indice can then be reduced with $\frac{1}{e}(h, k, l)$, and still define the same surface.
A solution will therefore exist for

$$hi + kj + lm = d \tag{5}$$

and because of the reduction of the normal vector, the value for d will be ± 1. The two surface vectors, must obey the fact that they are orthogonal with respect to the normal vector $\vec{n}$,

$$\vec{v}_{1,2} \cdot \vec{n} = 0. \tag{6}$$

Because of this, the cross product between the two surface vectors $\vec{v}_1$ and $\vec{v}_2$ must give a constant times the normal vector $\vec{n}$. The constant must be as small as possible, still non zero, because the area spanned by $\vec{v}_1$ and $\vec{v}_2$ is equal to the length of the cross product. And since the new normal vector is the smallest possible, the constant must be ± 1.
Consider a choice of the two surface vectors, $\vec{t}_1$ and $\vec{t}_3$. They will both fullfil equation 6, however the crossproduct between $\vec{t}_1$ and $\vec{t}_3$ will be,

$$\vec{t}_1 \times \vec{t}_3 = \begin{pmatrix} k \\ -h \\ 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ l \\ -k \end{pmatrix} = \begin{pmatrix} kh \\ k^2 \\ kl \end{pmatrix} = k \begin{pmatrix} h \\ k \\ l \end{pmatrix}.$$

Unless the size of k is ± 1, the size of the surface area spanned by $\vec{t}_1$ and $\vec{t}_3$ will be too big. It is therefore crucial to introduce a completely new linear combination of the three vectors $\vec{t}_1$, $\vec{t}_2$ and $\vec{t}_3$. A linear combination of $\vec{t}_1$

and $\vec{t}_2$ fullfill the same requirements when p and q are integers in

$$\left(p \begin{pmatrix} k \\ -h \\ 0 \end{pmatrix} + q \begin{pmatrix} l \\ 0 \\ -h \end{pmatrix} \right) \times \begin{pmatrix} 0 \\ l \\ -k \end{pmatrix} \Rightarrow (pk + ql) \begin{pmatrix} h \\ k \\ l \end{pmatrix} = \begin{pmatrix} h \\ k \\ l \end{pmatrix} \quad (7)$$

This leaves a more simple equation to solve,

$$(pk + ql) = 1 \quad (8)$$

The solution to this equation can be found using the Extended Euclidean Algorithm to determine the unknowns integers, p and q. The two new vectors, which span the surface are described

$$\vec{v}_1 = p \begin{pmatrix} k\vec{a}_1 \\ -h\vec{a}_2 \\ 0 \end{pmatrix} + q \begin{pmatrix} l\vec{a}_1 \\ 0 \\ -h\vec{a}_3 \end{pmatrix}, \ \vec{v}_2 = \begin{pmatrix} 0 \\ l\vec{a}_2 \\ -k\vec{a}_3 \end{pmatrix} \quad (9)$$

However, there are infinite possible solutions for p and q but some of the solutions are better than others, in relation to visualizing the surface. Therefore another criteria is implemented. The closer to being orthogonal the surface vectors are, the easier it becomes to apply adsorbates onto the the surface. The procedure for this will be explained in section 1.2, but the theory will be explained here. The solution for p and q can be chosen to accomodate this with respect to an integer, c, by

$$\vec{v}_1 = (p + cl) \begin{pmatrix} k\vec{a}_1 \\ -h\vec{a}_2 \\ 0 \end{pmatrix} + (q - ck) \begin{pmatrix} l\vec{a}_1 \\ 0 \\ -h\vec{a}_3 \end{pmatrix} \quad (10)$$

This change of vector $\vec{v}_1$, does not change the crossproduct between $\vec{v}_1$ and $\vec{v}_2$, as shown in equation 7, because the cross product between the changes and $\vec{v}_2$ is zero. This is shown below

$$\begin{aligned} \left(cl \begin{pmatrix} k\vec{a}_1 \\ -h\vec{a}_2 \\ 0 \end{pmatrix} - ck \begin{pmatrix} l\vec{a}_1 \\ 0 \\ -h\vec{a}_3 \end{pmatrix} \right) \times \begin{pmatrix} 0 \\ l\vec{a}_2 \\ -k\vec{a}_3 \end{pmatrix} &= \\ \left(ckl \begin{pmatrix} h \\ k \\ l \end{pmatrix} - ckl \begin{pmatrix} h \\ k \\ l \end{pmatrix} \right) &= 0 \end{aligned} \quad (11)$$

This change of $\vec{v}_1$ results in an algorithm, which will be presented later, to determine the most appropriate choice of vectors.

1.1.4 Finding the 3^{rd} vector for the new unit cell

After determining the two vectors spanning the surface $(\vec{v}_1, \vec{v}_2)$, the third basis vector($\vec{v}_3$) of the surface slab can be found. This vector does not need to be orthogonal to the two surface vectors. The vector will go from one lattice point to its repeated lattice point another place in the structure. This means that the same contraints apply to this vector as for $\vec{v}_1$ and $\vec{v}_2$, but some additional constraints will be added. The vector will have to be an integer linear combination of the three original lattice vectors $(\vec{a}_1, \vec{a}_2, \vec{a}_3)$ and have the coordinates $(i_3\vec{a}_1, j_3\vec{a}_2, m_3\vec{a}_3)$. In addition $\vec{v}_3$ cannot be orthogonal to the surface normal, so $\vec{v}_3 \cdot \vec{n} \neq 0$.
To find the integers i_3, j_3 and m_3 by calculating the dot product using normalvector of the surface from equation 3, and the definitions of the reciprocal vectors $(\vec{b}_1, \vec{b}_2, \vec{b}_3)$:

$$\begin{aligned} \vec{n} \cdot \vec{v}_3 &= (h\vec{b}_1 + k\vec{b}_2 + l\vec{b}_3) \cdot (i_3\vec{a}_1 + j_3\vec{a}_2 + m_3\vec{a}_3) \\ &= hi_3 + kj_3 + lm_3 = d, \end{aligned} \tag{12}$$

where d must be a non-zero integer because all of h, k, l, i_3, j_3 and m_3 are integers. It will now be shown that $d = 1$.
Defining the volume of the conventional unit cell to be V spanned by the conventional basis $\vec{a}_1, \vec{a}_2$ and $\vec{a}_3$.

$$V = (\vec{a}_1 \times \vec{a}_2) \cdot \vec{a}_3 \tag{13}$$

and rewriting of the three reciprocal vectors to the following form

$$V\vec{b}_1 = \vec{a}_2 \times \vec{a}_3,\ V\vec{b}_2 = \vec{a}_3 \times \vec{a}_1,\ V\vec{b}_3 = \vec{a}_1 \times \vec{a}_2. \tag{14}$$

will ease the calculations, and with all the pieces set, a determination of the value of d is possible. The volume of the cell spanned by $(\vec{v}_1, \vec{v}_2, \vec{v}_3)$ is presented below, where the constants $(i,j,m)_{1,2}$ refer to the constants defined by the formalism used for $\vec{v}_1$, equation 10, and for $\vec{v}_2$, equation 9.

$$\begin{aligned} V = \ & \vec{v}_3 \cdot (\vec{v}_1 \times \vec{v}_2) = \vec{v}_3 \cdot \{(i_1\vec{a}_1 + j_1\vec{a}_2 + m_1\vec{a}_3) \times (i_2\vec{a}_1 + j_2\vec{a}_2 + m_2\vec{a}_3)\} \\ = \ & \vec{v}_3 \cdot \{i_1j_2\vec{a}_1 \times \vec{a}_2 + i_1m_2\vec{a}_1 \times \vec{a}_3 + j_1i_2\vec{a}_2 \times \vec{a}_1 + j_1m_2\vec{a}_2 \times \vec{a}_3\} \\ & + \vec{v}_3 \cdot \{m_1i_2\vec{a}_3 \times \vec{a}_1 + m_1j_2\vec{a}_3 \times \vec{a}_2\} \\ = \ & V\vec{v}_3 \cdot \left\{i_1j_2\vec{b}_3 - i_1m_2\vec{b}_2 - j_1i_2\vec{b}_3 + j_1m_2\vec{b}_1 + m_1i_2\vec{b}_2 - m_1j_2\vec{b}_1\right\} \\ = \ & V\vec{v}_3 \cdot \left\{ \begin{pmatrix} i_1 \\ j_1 \\ m_1 \end{pmatrix} \times \begin{pmatrix} i_2 \\ j_2 \\ m_2 \end{pmatrix} \right\} \cdot \begin{pmatrix} \vec{b}_1 \\ \vec{b}_2 \\ \vec{b}_3 \end{pmatrix} \end{aligned} \tag{15}$$

The cross product between (i_1, j_1, m_1) and (i_2, j_2, m_2) has been found previously in equation 7 using the Extended Euclidian Algorithm as (h, k, l). Inserting this into the equation

$$V = V\vec{v}_3 \cdot (h\vec{b}_1 + k\vec{b}_2 + l\vec{b}_3) = Vd. \tag{16}$$

d is therefore equal to 1. The knowledge of d in equation 12 leads to a new equation to solve, to determine the third vector $\vec{v}_3$.

$$\begin{aligned} \vec{n} \cdot \vec{v}_3 &= (h\vec{b}_1 + k\vec{b}_2 + l\vec{b}_3) \cdot (i_3\vec{a}_1, j_3\vec{a}_2, m_3\vec{a}_3) \\ &= hi_3 + kj_3 + lm_3 = 1 \end{aligned} \tag{17}$$

This equation can be solved using the Extended Euclidean Algorithm for three variables and therefore the third vector $\vec{v}_3$ is determined. With these three vectors, $(\vec{v}_1, \vec{v}_2, \vec{v}_3)$, a basis for the new unit cell is created. The implementation of this will be described in the following section, along with some ways to get around the numerical issues in Python.

1.2 Implementation in Python

Based on the theory derived above an arbitrary surface can be created using the procedure found on the DTU niflheim cluster. To create a surface using the procedure described in this section a conventional bulk cell of the surface material is needed along with the Miller indices and the depth of the slab.
The implementation in Python using ASE to setup the atoms consists of three parts. First, a new basis is derived from the Miller indices with two of the basis vectors lying in the surface plane. Secondly, the atoms in the conventional bulk cell are expressed in the terms of the new basis in a slab with the selected depth. Finally, the unit cell of the slab is modified so the third cell vector points perpendicular to the surface and all atoms are moved into the unit cell.

1.2.1 Surface basis

For any surface type described by a Miller indice (h, k, l) the surface basis $(\vec{v}_1, \vec{v}_2, \vec{v}_3)$ is found relative to the conventional bulk unit cell. $\vec{v}_1$ and $\vec{v}_2$ are chosen to be in the surface plane.
In the special case where only one of the Miller indices is non-zero $\vec{v}_1$ and $\vec{v}_2$ are simply the unit vectors in the directions where the Miller indices are zero, respectively and $\vec{v}_3$ is the direction where the Miller indice is non-zero.

For all other situations $\vec{v}_1$ and $\vec{v}_2$ are found by solving the linear equation 8 using the Extended Euclidean Algorithm - in the script defined as `ext_gcd()`. This yields an infinite set of solutions all of which can be used. However, the optimal structure is found when the angle between the two base vectors are as close to 90^o as possible, as the structure will be as compact as possible and specific sites are easier to identify. This solution is found by minimizing the scalar product of the two base vectors by $c \in \mathbf{Z}$.

$$\left| \left((p+cl) \begin{pmatrix} k\vec{a}_1 \\ -h\vec{a}_2 \\ 0 \end{pmatrix} + (q-ck) \begin{pmatrix} l\vec{a}_1 \\ 0 \\ -h\vec{a}_3 \end{pmatrix} \right) \cdot \begin{pmatrix} 0 \\ l\vec{a}_2 \\ -k\vec{a}_3 \end{pmatrix} \right|_{min(c)}$$

This can be expressed as $|k_1 + ck_2|_{min(c)}$ and the solution is found when c is equal to the fraction $-\frac{k_1}{k_2}$ rounded to the nearest integer. Because of numerical errors a tolerance is used. In python this is expressed as follows.

```
p,q = ext_gcd(k,l)
k1 = dot( p*(k*a1-h*a2)+q*(l*a1-h*a3) , l*a2-k*a3)
k2 = dot( l*(k*a1-h*a2)-k*(l*a1-h*a3) , l*a2-k*a3)
if abs(k2)>tol:
 c = -int(round(k1/k2))
 p,q = p+c*l, q-c*k
v1 = p*array((k,-h,0))+q*array((l,0,-h))
v2 = reduce(array((0,l,-k)))
a,b = ext_gcd(p*k+q*l,h)
v3 = array((b,a*p,a*q))
```

The last four lines define the base vectors for the surface using the Extended Euclidean Algorithm for two variables to find $\vec{v}_1$ and $\vec{v}_2$ and three variables to find $\vec{v}_3$.

1.2.2 Atom positions

When the basis have been found the atoms in the conventional cell are base-changed to the new basis using

```
for i in range(len(bulk)):
 newpos = linalg.solve(basis.T,bulk.get_scaled_positions()[i])
 scaled += [newpos-floor(newpos+tol)]
```

and then moved so the scaled positions in the new basis are within the box spanned by the basis. The tolerance is needed so atoms positioned exactly

on the boundary are treated consistently despite numerical errors. The cell in the new basis is then repeated in the $\vec{v}_3$ direction to create the required slap depth.
For many applications it is useful to have the z-direction pointing perpendicular to the surface to enable electrostatic decoupling and to make vacuum height and adsorbate distance well defined. The next step in the procedure is therefore to align the z-direction with the cross product of $\vec{v}_1$ and $\vec{v}_2$ with a length so the cell volume is preserved. The final step before the slab is created is then to move the atoms so the scaled coordinates are between 0 and 1 in the $\vec{v}_1$ and $\vec{v}_2$ directions making it obvious how the atoms are located relative to each other when the structure is visualized.