ANDRZEJ DANILUK 5 JULY 2020

rheed++ User guide

1. The growth++ program

The *growth++* program is distributed in the form of .cpp file and one header file. The growt++.cpp contains all functionalities of the program. The header file contains:

- tm.h – aliases and templates used in the code.

The *growth++.cpp* is written in C++11, therefore one can use GNU g++ compiler or any C++11 compliant compiler to build the executable program. For example, under Linux use the g++ compiler: g++ -std=c++11 growth++.cpp -o growth++, and run the program: ./growth++

1.1. Input data

Input data are read from the *inputGrowthData.dat* text file where the user provides information on the parameters of the growth simulation, i.e. growth model:

0 - diffusive growth,

1 - distributed growth-variant 1,

2 - distributed growth-variant 2,

3 - distributed growth-variant 3,

the number of growing layers (*numLayers*), the upper limit of growth time interval (*tMax*), the number of integration intervals (*numIntervals*), the values of A_n (or k_n) parameters, and corresponding values of the gR_n (growth rate for each layer, See sections 1.3-1.4).

Sample contents of the input file *inputGrowthData.dat*, for variant 1 of the distributed growth model:

```
1                    growth model
------------
12                   number of growing layers
9                    upper limit of growth time interval
1000                 number of integration intervals
An/kn--gRn--
0.9     1
0.6     1
0.9     0.8
0.9     0.8
0.8     0.8
0.7     0.8
0.6     0.8
0.6     0.8
0.6     0.8
0.6     0.8
0.6     0.8
0.6     0.8
```

1.2. Output data

Output data are written on a disc in the current directory in the form of *coverage.dat* text file. The data in this file are saved in two columns: the first column stores the values of growth time, the second – the values of layer coverages.

Sample contents of the output file *coverage.dat*:

```
12          number of growing layers
608         number of returned values
0           0
0.0194295   0.0194256
0.0389367   0.0389164          growth time [t/tau]
0.0583636   0.0583085
0.0775918   0.0774789
0.097405    0.0972027
0.117954    0.117621           layer coverage
0.137916    0.137415
0.157058    0.156351
0.178043    0.177055
```

1.3. The diffusive growth

The process of the diffusive growth is described by the following set of coupled differential equations:

$$\frac{d\theta_n}{dt} = gR_n(\theta_{n-1} - \theta_n) + k_n(\theta_{n+1} - \theta_{n+2})(\theta_{n-1} - \theta_n) - k_n(\theta_n - \theta_{n+1})(\theta_{n-2} - \theta_{n-1}), \quad (1)$$

subject to the initial conditions: $\theta_0(t) = 1$, $\theta_n(0) = 0$, and $\theta_\infty(t) = 0$.

In these equations θ_n is the coverage ratio of the *n*th surface layer. k_n is the parameter that determines the coverage profile of the growing film surface. Large values of k_n give rise to an effective mass transport from higher layers to lower layers. For large values of k_n the diffusive growth model can be used for modelling the growth of the Frank–Van der Merwe (FV) type. The growth rate gR_n is $1/\tau_n$, where τ_n is the deposition time of n*th* layer.

1.4. The distributed growth

In this model, the outer edges of the films are taken into consideration, and the atoms which reach the area of the layers will be divided according to the number of possible places of interaction. The process of the distributed growth is described by the following set of coupled differential equations:

$$\frac{d\theta_n}{dt} = gR_n[(\theta_{n-1} - \theta_n) + \alpha_n(\theta_n - \theta_{n+1}) - \alpha_{n-1}(\theta_{n-1} - \theta_n)], \quad (2)$$

subject to the initial conditions: $\theta_0(t) = 1$, $\theta_n(0) = 0$ for $n \geq 1$, and $\theta_\infty(t) = 0$.

The parameter α_n is defined by:

$$\alpha_n = A_n \frac{d_n(\theta_n)}{d_n(\theta_n) + d_{n+1}(\theta_{n+1})}. \quad (3)$$

$d_n(\theta_n)$ is the perimeter of the *n*th film, and A_n is the phenomenological parameter that measures the net rate of transfer from one layer to the next. For A_n close to 1.0 this model gives perfect layer-by-layer growth (FV), and for A_n close to 0 it gives non-diffusive growth (growth of the Volmer–Webber type).

1.4.1. Variant 1

In this model we can choose the following dependence:

$$d_n(\theta_n) = \theta_n \sqrt{(1-\theta_n)}, \quad (4)$$

which corresponds to a growth mode in which both the number and size of the nucleation sites change during film growth.

1.4.2. Variant 2

In this model we can choose the following dependence:

$$d_n(\theta_n) = \theta_n (1-\theta_n), \quad (5)$$

which corresponds to a growth mode that depends solely on the number of filled and empty places on the surface.

1.4.3. Variant 3

Assuming that in the case of the surface which is less than half filled, there is a definite number of islands and each of them has the same perimeter, and also that in the case of the surface which is more than half filled there is a definite number of empty sites with the same perimeter, the following can be written:

$$d_n(\theta_n) = \theta_n^{p_1} \quad \text{for} \quad \theta_n \langle \theta_c, \quad (6a)$$

$$d_n(\theta_n) = (1-\theta_n)^{p_2} \quad \text{for} \quad \theta_n \rangle \theta_c, \quad (6b)$$

with:

$$\theta_c^{p_1} = (1-\theta_c)^{p_2}, \quad (6c)$$

and $p_1 = p_2 = 0.5$, $\theta_c = 0.5$.

In this model one can notice a considerable flexibility, which is dependent upon the way in which the perimeter of the film is linked to the coverage of this particular film. For more details about growth modes See Refs [10, 11].

2. The rheed++ program

The *rheed++* program is distributed in the form of .cpp file and three header files. The *rheed++.cpp* contains all functionalities of the program. The header files contain:

- dtp.h - Doyle and Turner constants parameters of the analytic representation of the electron scattering factors [13],
- uc.h - values of universal physical constants and constants for the crystals under investigation,
- tm.h – aliases and templates used in the code.

The *rheed++.cpp* is written in C++11, therefore one can use GNU g++ compiler or any C++11 compliant compiler to build the executable program. For example, under Linux use the g++ compiler: g++ -std=c++11 rheed++.cpp -o rheed++, and run the program ./rheed++

2.1. Input data

The presented algorithm allows the user significant freedom in implementation, so (for simplicity of code analysis) the parameters for the crystals under investigation are defined in *uc.h* and *dtp.h* header files.

Input data are read from the *inputRheedData.dat* text file where the user provides information on the parameters of the RHEED simulation, i.e.: values of the parameters α and β that describe the imaginary parts of the potential, temperature of the crystal [K], energy of the incident electron beam [eV], and incident angle of the electron beam [deg].

Sample contents of the input file *inputRheedData.dat*:

```
0.1 ——α
0.5 ——β  values of the imaginary parts of the potential
900 ———— temperature of the crystal [K]
10E+3 ——— energy of the incident electron beam [eV]
1.238 ———— incident angle of the electron beam [deg]
```

2.2. Output data

The execution generates three text files with results of the calculations. The *potential.dat* file stores values of the real part of the crystalline potential for the structures under investigation. The data are saved in two columns: the first column stores the values of z-axis (expressed in [Å]), the second—the values of the crystal potential (expressed in [eV] units). The amplitude of the RHEED specular beam intensity oscillations is saved in the file whose name contains simulation parameters. Data in this file are saved in two columns: the first column stores the growth time (expressed in [t/τ] units), the second one - the specular reflected beam amplitude. The normalized data are saved in the file with prefix *norm-*.

3. How to run the simulation for the first time

1. Prepare the contents of the *inputGrowthData.dat*

2. Prepare the contents of the *inputRheedData.dat*

3. Compile and run the *growth ++* program

4. Compile and run the *rheed ++* program