UDKM1DSIM - A Simulation Toolkit for 1D Ultrafast Dynamics in Condensed Matter

D. Schicka, A. Bojahr, M. Herzoga,b, R. Shaydukc, C. von Korff Schmisingd, M. Bargheera,c

aInstitut für Physik & Astronomie, Universität Potsdam, Karl-Liebknecht-Straße 24-25, 14476 Potsdam, Germany
bAbteilung Physikalische Chemie, Fritz-Haber-Institut der Max-Planck-Gesellschaft, Faradayweg 4-6, 14195 Berlin, Germany
cHelmholtz-Zentrum Berlin für Materialien und Energie GmbH, Wilhelm-Conrad-Röntgen Campus, BESSY II, Albert-Einstein-Straße 15, 12489 Berlin, Germany
dInstitut für Optik und Atomare Physik, Technische Universität Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany

Abstract

The udkm1DSIM toolbox is a collection of MATLAB (MathWorks Inc.) classes and routines to simulate the structural dynamics and the according X-ray diffraction response in one-dimensional crystalline sample structures upon an arbitrary time-dependent external stimulus, e.g. an ultrashort laser pulse. The toolbox provides the capabilities to define arbitrary layered structures on the atomic level including a rich database of corresponding element-specific physical properties. The excitation of ultrafast dynamics is represented by an N-temperature model which is commonly applied for ultrafast optical excitations. Structural dynamics due to thermal stress are calculated by a linear-chain model of masses and springs. The resulting X-ray diffraction response is computed by dynamical X-ray theory. The udkm1DSIM toolbox is highly modular and allows for introducing user-defined results at any step in the simulation procedure.

Keywords: ultrafast dynamics, heat diffusion, N-temperature model, coherent phonons, incoherent phonons, thermoelasticity, dynamical X-ray theory

Program Summary

Manuscript Title: udkm1DSIM - A Simulation Toolkit for 1D Ultrafast Dynamics in Condensed Matter
Authors: D. Schick, A. Bojahr, M. Herzog, R. Shayduk, C. von Korff Schmising, M. Bargheer
Program Title: udkm1DSIM
Journal Reference: 
Catalogue identifier: 
Licensing provisions: BSD
Programming language: MATLAB (MathWorks Inc.)
Computer: PC/Workstation
Operating system: running MATLAB installation required (tested on MS Win XP - 7, Ubuntu Linux 11.04-13.04)
RAM: MATLAB’s typical RAM requirement of 196MB is sufficient for most simulations
Has the code been vectorized or parallelized?: parallelization for dynamical XRD computations
Number of processors used: 1-12 for MATLAB Parallel Computing Toolbox; 1–∞ for MATLAB Distributed Computing Toolbox
Keywords: ultrafast dynamics, heat diffusion, N-temperature model, coherent phonons, incoherent phonons, thermoelasticity, dynamical X-ray theory
Classification: 7.8 Structure and Lattice Dynamics, 7.9 Transport Properties, 8 Crystallography
External routines/libraries: required (included in the package): TIMEX, Fast Matrix Multiply for MATLAB by James Tursa, XML, O2TOOLS by Jaroslaw Tuszyński, TEXTPROGRESSBAR by Paul Proteus
Nature of problem: Simulate the lattice dynamics of 1D crystalline sample structures due to an ultrafast excitation including thermal transport and compute the corresponding transient X-ray diffraction pattern.
Solution method: The program provides an object-oriented toolbox for building arbitrary layered 1D crystalline sample structures including a rich database of element-specific parameters. The excitation, thermal transport and lattice dynamics are simulated utilizing MATLAB’s ODE solver. Alternatively, the lattice dynamics can also be calculated analytically utilizing MATLAB eigenproblem solver. The dynamical X-ray diffraction is computed in a parallelized matrix formalism.
Restrictions: The program is restricted to 1D sample structures and is further limited to longitudinal acoustic phonon modes and symmetrical X-ray diffraction geometries.
Unusual features: The program is highly modular and allows to include user-defined inputs at any time of the simulation procedure.
Running time: The running time is highly dependent on the number of unit cells in the sample structure and other simulation parameters such as time span or angular grid for X-ray diffraction computations. However, the example files are computed in approx. 1-5 min. each on a 8 Core Processor with 16GB RAM available.

*Corresponding author
Email address: daniel.schick@uni-potsdam.de (D. Schick)
URL: http://www.udkm.physik.uni-potsdam.de (M. Bargheer)

Preprint submitted to Computer Physics Communications September 23, 2013
1. Introduction

Physics on the ultrafast time scales and nanometer length scales has received enormous attention during the last decade. Ultrafast X-ray diffraction (UXRD) techniques allow for directly studying structural dynamics on the atomic length and time scales. The knowledge of the time-resolved structural response to an ultrafast optical stimulus is essential for the understanding of various condensed matter phenomena.\[1-4\]

The udkm1Dsim toolbox is a collection of classes and routines to model 1D crystalline sample structures on the atomic level and to simulate incoherent (heat diffusion) as well as coherent lattice dynamics (acoustic phonons) by semi-coupled equations of thermoelasticity.\[5, 6\] The resulting transient X-ray diffraction response for the 1D sample structure is computed by dynamical X-ray theory.\[7, 8\]

Due to the high modularity of the toolbox it is easy to introduce user-defined procedures in between the simulation steps. The complete package is written in the MATLAB programming language and requires the installation of the MATLAB software environment. In order to use the multi-core capabilities of MATLAB the Parallel Computing has to be installed but is not required for udkm1Dsim to work. As a convention for this document, all files and directories are formatted without serifs (/path/file.ext) and all MATLAB code is written in typewriter format (code = [1 10]). Furthermore, all physical quantities have to be input in SI units and the same applies for all output variables.\[1\]

The latest udkm1Dsim package files can be downloaded from www.udkm.physik.uni-potsdam.de/udkm1dsim including a detailed documentation and example files. It is highly recommended to be familiar with the basics of MATLAB programming as well as with fundamental object-orientated programming schemes. Please refer to the rich MATLAB documentation on these topics for further help.

In the following, we introduce the implementation and common workflow of the udkm1Dsim toolbox as well as as the underlying physical concepts. Please refer to the specific class documentations in the ./documentation/ folder of the toolbox for detailed information on all available methods and properties. Finally, we provide examples of the udkm1Dsim package which are compared with selected ultrafast experiments on nano-layered thin film samples.

2. Implementation & Workflow

The udkm1Dsim package is developed as MATLAB toolbox with a command-line/script-based user-interface. The backbone of the fully object-orientated toolbox is a collection of classes in the ./classes/ folder which hold the complete logic for building 1D sample structures and to calculate the ultrafast dynamics in these structures. Additional helper routines (/helpers/) and material parameter files (/parameters/) are included to improve the user experience.

2.1. Structure Generation

The common workflow of a simulation procedure is to create a crystalline sample structure at the beginning. This 1D structure is build of atoms which form unit cells. Unit cells are then grouped to layers/sub-structures which can be further nested, e.g. to build multilayer structures. All physical properties which are necessary for the later simulations are stored in this structural objects. The involved files are atomBase.m, atomMixed.m, unitCell.m and structure.m.

2.1.1. Atoms

The smallest building block for a structure is an atom, which is represented by the atomBase class. Atomic properties are automatically loaded on construction of each atomBase instance from the given parameter files, by providing the correct symbol of the desired chemical element:

\[
C = \text{atomBase('C')};
\]
\[
\text{Ni} = \text{atomBase('Ni')};
\]

By executing the command C.disp() all properties of the corresponding atomBase object are displayed. Solid solutions, i.e. stoichiometric atomic substitutions, can be modelled by the atomMixed class. Here, atomBase objects can be added with an according relative amount to the solution:

\[
\text{ZrTi} = \text{atomMixed('0.2 Zirconium/0.8 Titanium')};
\]
\[
\text{ZrTi}.\text{addAtom(\text{atomBase('Zr')}, 0.2)};
\]
\[
\text{ZrTi}.\text{addAtom(\text{atomBase('Ti')}, 0.8)};
\]

The resulting mixed atomic properties are the weighted average of the constituent’s properties.

2.1.2. Unit Cells

The unitCell class holds most of the physical properties which are necessary for the further simulations. In addition to structural information, i.e. the position of atoms in the unit cell, thermal and mechanical properties are stored here. The only required parameters on initialization of a unitCell instance are a unique identifier (ID), name, and the c-axis (lattice parameter normal to the sample surface) of the unit cell. All other properties can be optionally handed over within a parameter struct on construction, or can be added/modified later:

\[
c\text{Axis} = 3.95e-10; \quad \% \text{[m]}
\]
\[
\text{prop}.\text{soundVel} = 5100; \quad \% \text{[m/s]}
\]
\[
\text{prop}.\text{heatCapacity} = 465; \quad \% \text{[J/kg K]}
\]

so SrRuO3 - Perovskite

\[
\text{SRO} = \text{unitCell('SRO', 'SrRuO3', cAxis, prop);}
\]
After the construction of a unitCell object, one can add atomBase or atomMixed object at relative positions in the 1D unit cell, e.g. for the cubic SrRuO₃ (SRO) perovskite unit cell:

SRO.addAtom(Sr, 0.5);
SRO.addAtom(O, 0);
SRO.addAtom(Sr, 0.5);
SRO.addAtom(O, 0.5);

In the 1D approximation the lateral position of the atoms in the unit cell is not relevant. One needs to determine the position of the individual atoms in the unit cell along their projection onto the surface normal of the sample. Moreover, it is not possible to add fractions of atoms at certain unit cell position. Hence, one has to translate the origin of the unit cell accordingly.

All available unit cell properties can be easily displayed by executing the command SRO.disp(). The position of atoms in the unit cell can be visualized by executing SRO.visualize().

### 2.1.3. Structures

The final 1D crystalline samples are represented by the structure class which only requires a name on initialization. One can add any number of unitCell objects to a structure, as well as nested substructures. An example of a SrRuO₃/SrTiO₃ superlattice with 10 periods on a SrTiO₃ (STO) substrate is shown in the listing below:

```matlab
DL = structure('Double Layer');
DL.addSubStructure(SRO, 13);
%
S = structure('Superlattice Sample');
S.addSubStructure(DL, 10);
%
S.addSubStructure(STO, 1000);
```

In order to simplify the sample structure creation, all of the above mentioned steps can be included in an external XML file which holds all information on atoms, unit cells and on the structure itself. Hence, it is easy to store structures outside of MATLAB in a unified and open standard. An example XML file is provided in the ./example/ folder of the toolbox. In order to load the data from the XML file into the MATLAB workspace one needs to execute the following command providing the relative or absolute path to the XML file:

```matlab
S = structure('void', './structure.xml');
```

Again, the structure properties can be displayed with the command S.disp() and the structure can be visualized by S.visualize().

### 2.2. Simulation Classes

Besides the 1D sample structures, also all simulations are programmed as classes and inherit from the super-class simulation. All simulation-inherited classes provide fundamental properties and methods for storing and loading of simulation results from a so-called /cache/ folder. The UDKM1DSIM toolbox can decide independently by comparing a unique hash of all simulation input parameters whether a simulation result (once calculated) can be loaded from the cache folder or needs to be (re-)calculated. The hash algorithm decides also which parameter changes are relevant for a simulation model, e.g. a change of the sound velocity of a unit cell does not change the result of the heat diffusion calculation, however it does change the result of the lattice dynamics simulation. Further functionalities of the simulation class are to enable/disable any command-line messages during the simulations, e.g. to display the elapsed time for a simulation step, and to change the mode of progress displaying.

In order to calculate the time-dependent X-ray diffraction response of a 1D crystalline sample structure to an ultrafast stimulus the following three simulations steps are necessary:

1. The excitation is described as temperature changes in an N-temperature model with optional heat diffusion which determines the temperature evolution in the N coupled subsystems.
2. The resulting lattice dynamics due to thermal stress possibly generated by any of the N subsystems are calculated by a 1D linear-chain model.
3. Dynamical X-ray theory is applied to calculate the UXRD response to the lattice dynamics.

These three steps are encapsulated in the simulation classes heat, phonon, and XRD which all require a structure object on initialization.

It is important to note, that each of the simulation steps listed above may be executed independently with user-defined inputs. Thus it is not necessary to execute the heat and phonon simulations if the user needs to calculate the X-ray diffraction result e.g. for artificial or externally calculated lattice dynamics.

### 2.3. Thermal Excitation & Diffusion

The UDKM1DSIM toolbox allows for different excitation scenarios and optional thermal transport. The most general model is an N-temperature model (NTM) which is described in section 2.3.1. However, for various experimental cases it is convenient to simplify the simulation procedure in order to save computational time.

In all cases it is assumed that the sample structure is excited by a light pulse which is absorbed following Lambert-Beer’s law:

\[
I(z) = I_0 e^{-z/\zeta},
\]  

(1)
with $\zeta$ as optical penetration depth of the material for the considered wavelength. Accordingly, one can define the transmission and absorption in the material as follows:

$$\tau = \frac{I(z)}{I_0} = e^{-z/\zeta}, \quad \alpha = 1 - \tau. \quad (2)$$

Note that we do not consider the reflected laser light here. The deposited optical energy is then given by the spatial derivative of the absorption:

$$\frac{\partial \alpha}{\partial z} = \frac{1}{\zeta} e^{-z/\zeta}. \quad (3)$$

The most simplified excitation scenario is represented by an instantaneous temperature jump of the excited sample structures (infinitely short laser pulse). This assumption is generally valid if the excitation and thermal equilibration between all $N$ subsystems happen much faster than the subsequent thermal and/or lattice dynamics. The instantaneous temperature jump at depth $z$ can be calculated from the energy absorbed by the corresponding unit cell via:

$$\Delta E(z) = \int_{T_1}^{T_2} m \cdot c[T(z)] \, dT(z), \quad (4)$$

where $T_1$ is the initial and $T_2$ is the final temperature of the unit cell, $m$ is the unit cell mass, and $c(T)$ is the temperature-dependent specific heat capacity. In order to calculate the absorbed energy per unit cell at the depth $z$ in the sample structure one can linearize Eq. (3) for small $\Delta z$ in terms of energy instead of intensity to get

$$\Delta E = \frac{\partial \alpha}{\partial z} E_0 \Delta z, \quad (5)$$

where $\Delta z$ is the size of the according unit cell. The initial energy $E_0$ which is incident on the first unit cell can be derived from the incident absorbed fluence $F = E_0/A$, where $A$ is the area of a single unit cell. Hence, one has to minimize the following modulus in order to obtain the final temperature $T_2$ of a unit cell after optical excitation:

$$\left| \int_{T_1}^{T_2} m \cdot c[T(z)] \, dT(z) - \frac{E_0}{\zeta} e^{-z/\zeta} \Delta z \right| \leq 0. \quad (6)$$

In order to solve the above minimization problem it is necessary that the heat capacity $c(T)$ is input as a polynomial of any order, thus enabling MATLAB to integrate $c(T)$ algebraically with respect to the temperature $T$.

The temperature jump resulting from the optical excitation at $t = 0$ can be further used as initial condition for solving the 1D heat diffusion equation:

$$c[T(z,t)] \frac{\partial T(z,t)}{\partial t} = \frac{\partial}{\partial z} \left( k[T(z,t)] \frac{\partial T(z,t)}{\partial z} \right) \quad (7)$$

including the thermal conductivity $k(T)$ and mass density $\rho$ of the individual unit cells. The UDKM1DSIM toolbox is capable of calculating the optical excitation and thermal dynamics independently for a given sample structure, with corresponding parameters, and excitation scenario. The corresponding code listing for an excitation at $t_0 = 0$ with a fluence of $F = 5 \text{ mJ/cm}^2$ including heat diffusion for a given sample structure might look as follows:

```matlab
% initialization of heat simulation
H = heat(S,forceRecalc);
% S - structure object
% forceRecalc - boolean
% enable heat diffusion
H.heatDiffusion = true;
% initial temperature of the structure
initTemp = 300 * u.K;
% define the excitation
F = 5*u.mJ/u.cm^2;
% the temperature profile is calculated:
[tempMap, deltaTempMap] = ...
    H.getTempMap(time,F,initTemp);

Here, initTemp is the initial temperature of the sample, which can be defined globally or per unit cell and the vector time defines the time grid of the calculation. The actual numerical calculation is executed by the last command in the above listing and requires no further insight into the involved mathematics. The UDKM1DSIM toolbox allows for more sophisticated excitation scenarios, such as optical pulse sequences with arbitrary temporal pulse separations and durations as well as user-defined pulse energy distributions. Please refer to the corresponding examples for further details on this topic.

2.3.1. N-Temperature Model

The so-called N-temperature model (NTM) is a very general model for laser heating of metals and semiconductors. In the NTM materials are described by $N$ thermal subsystems having individual temperatures $T_j(z,t)$, heat capacities $c_j(T_j)$, thermal conductivities $k_j(T_j)$ and coupling terms $G_j(T_1,...,T_N)$. The subsystems might be represented by e.g. electrons, lattice, or spins of the according material:

$$c_1(T_1) \frac{\partial T_1}{\partial t} = \frac{\partial}{\partial z} \left( k_1(T_1) \frac{\partial T_1}{\partial z} \right) + G_1(T_1,...,T_N) + S(z,t)$$

$$:$$

$$c_N(T_N) \frac{\partial T_N}{\partial t} = \frac{\partial}{\partial z} \left( k_N(T_N) \frac{\partial T_N}{\partial z} \right) + G_N(T_1,...,T_N). \quad (8)$$

The UDKM1DSIM toolbox limits the excitation of a structure with $N$ subsystem to happen exclusively in the first subsystem. The excitation can be either given as an initial condition due to an instantaneous temperature jump (see above) or by a spatially and temporally varying source term $S(z,t)$. This source term is the energy flux per vol-
state of the laser-excited crystal. In order to calculate the temperature $T$ of the subSystemCoupling property which represents the term $T_j$.

2.4. Lattice Dynamics

The optically induced temperature change usually induces thermal stress in laser-heated materials. This thermal stress eventually relaxes via thermal expansion which is quantified by the linear thermal expansion coefficient:

$$\alpha(T) = \frac{1}{L} \frac{dL}{dT} .$$

(12)

Since the temperature change $\Delta T(z,t)$ for each unit cell at each time step is known one can calculate the actual thermal expansion of each unit cell by

$$l = \Delta L = L_1 \left( e^{A(T_2) - A(T_1)} - 1 \right) ,$$

(13)

where $L_1$ is the initial length ($c$-axis of the unit cell), $A(T)$ is the integral of $\alpha(T)$, $T_1$ and $T_2$ denote the initial and final temperatures of each unit cell, respectively. It is again necessary to define $\alpha(T)$ as a polynomial of any order of the temperature $T$ to enable MATLAB for simple and fast algebraic integration.

The thermally expanded unit cells are only the final state of the laser-excited crystal. In order to calculate the transient lattice dynamics (including only longitudinal acoustic phonons) towards this final state, we set up a model of a linear chain of masses and springs in which each unit cell represents a mass $m_i$ that is coupled to its neighbors via springs with the spring constant $k_i = m_i v_i^2/\epsilon_i$ ($c_i$ - lattice $c$-axis, $v_i$ - longitudinal sound velocity). \[10\] $m_i \ddot{x}_i = - k_i (x_i - x_{i-1}) - k_{i+1} (x_{i+1} - x_i) + m_i \gamma_i (\dot{x}_i - \dot{x}_{i-1}) + F_i^{\text{heat}}(t) .$$

(14)

Here $x_i(t) = z_i(t) - z_i^0$ denotes the shift of each unit cell from its initial position. Furthermore, we introduce an empirical damping term $F_i^{\text{damp}} = \gamma_i (\dot{x}_i - \dot{x}_{i-1})$ and the external force (thermal stress) $F_i^{\text{heat}}(t)$. In order to solve this system of coupled differential equations for each of the $i = 1 \ldots N$ unit cells the udkm1Dsim toolbox provides an analytical (phononAna) and a numerical model (phononNum) which are described in detail below. Examples for both models are given in the example files phononExample.m, and phononAnharmonicExample.m.

2.4.1. Analytical Solution

To obtain an analytical solution of Eq. (14) we neglect the damping term $F_i^{\text{damp}}(t)$ and derive the homogeneous differential equation in matrix form

$$\frac{d^2}{dt^2} X = K X .$$

(15)

Here $X = (x_1 \ldots x_N)$ and $K$ is the tri-diagonal force matrix. \[10\] The matrix $K$ can be diagonalized to obtain the eigenvectors $\Xi_j$ and eigenfrequencies $\omega_j$ in order to find the general solution

$$X(t) = \sum_j \Xi_j (A_j \cos(\omega_j t) + B_j \sin(\omega_j t))$$

(16)

Mathematical details on the analytical model are given in Ref. [10] and in the documentation of the phononAna class. Generally, we use MATLAB’s capability to solve the eigenproblem for $K$ in order to get the results for $X(t)$ for each time step. One can implement the thermal stress as new equilibrium position $x_i^\text{eq}(t)/\text{initial conditions}$ for the general solution Eq. (10) by doing an according coordinate transformation. The thermal stress [$F_i^{\text{heat}}(t)$] can be modeled as spacer sticks $l_{ij}$ in between the unit cells which are calculated from Eq. (13).

As an example listing of the analytical solution of the coherent phonon dynamics we continue the above code, having the structure $S$, time and the results of the heat simulation ($\text{tempMap}, \text{deltaTempMap}$) in memory.

```matlab
% initialization of analytical phonon simulation
P = phononAna(S,forceRecalc);
% the strain profile is calculated:
strainMap = ...
    P.getStrainMap(time,tempMap,deltaTempMap);

The matrix deltaTempMap is the temporal derivative of the temperature profile tempMap. The analytical model
has the advantage that once the eigenproblem is solved for a fixed $K$ (fixed sample structure) the strain profile can be easily solved for any excitation profile at any time. In the case of a quasi-instantaneous excitation without heat diffusion this results in an extremely fast calculation since the initial conditions $X(0)$ change only once for the excitation. However, the analytical model becomes rather slow for time-dependent thermal stress, because of the recalculation of these initial conditions for each time step. Accordingly, the temporal variation of the thermal stress due to damping has not been implemented in this model. The main disadvantage of the analytical model is the limitation to purely harmonic inter-atomic potentials which is overcome by the numerical model, described below. The numerical model is generally also faster in the total computational time and further accounts for phonon scattering and damping effects.

### 2.4.2. Numerical Solution

Mathematical details on the numerical model for the coherent phonon dynamics can be found in the documentation of the phononNum class and in Ref. [13]. Generally, we use MATLAB’s ODE solver to calculate the results for Eq. [14] which can be simplified to

$$m_i\ddot{x}_i = F^\text{spring}_i + F^\text{damp}_i + F^\text{heat}_i.$$  

Here $F^\text{spring}_i = -k_i(x_i - x_{i-1}) - k_{i+1}(x_i - x_{i+1})$ is the force acting on each mass due to the relative shifts in respect to the left and right neighboring masses. The numerical solution also allows for non-harmonic inter-atomic potentials of up to the order $M$. Accordingly, $k_i = (k^{(1)}_i \ldots k^{(M-1)}_i)$ can be a vector accounting for higher orders of the potential which is purely quadratic ($k_i = k^{(1)}_i$) in the harmonic case. Thus we can introduce the following term into $F^\text{spring}_i$:

$$k_i(x_i - x_{i-1}) = \sum_{j=1}^{M-1} k^{(j)}_i (x_i - x_{i-1})^j,$$

which accounts for the anharmonic interaction. In order to calculate anharmonic phonon propagation, including damping, one needs to set the according properties of the unitCell object. For the example of the SRO unit cell defined in Sec. 2.7 one has to write

```matlab
SRO.phononDamping = 1e-12; % [kg/s]
SRO.setHOspringConstants([-7e12]); % [kg/m s^-2]
```

which sets the damping constant to $\gamma_{\text{SRO}} = 10^{-12}$ kg/s and the second-order of the spring constant to $k^{(2)}_{\text{SRO}} = -7 \times 10^{12}$ kg/m s$^2$. The actual numerical calculation for the coherent phonon dynamics is similar to the analytical model expect for the initialization of the phononNum object at the beginning:

```matlab
% initialization of numerical phonon simulation
P = phononNum(S, forceRecalc);

% the strain profile is calculated:
P.getStrainMap(time, tempMap, deltaTempMap);
```

We want to highlight, that the analytical and numerical lattice dynamics calculations share the same syntax in order to calculate the strain profile after optical excitations. In addition, the user can input any temperature profile for the thermal stresses and is not limited to the results of the heat simulations. In accordance to the NTM described in Sec. 2.3.1 the thermal stresses can account for multiple thermodynamic subsystems in the sample by introducing different unitCell linear thermal expansion coefficients $\alpha_j(T_j)$ for the $j$th subsystem.

### 2.5. X-Ray Diffraction

In order to probe transient lattice dynamics with atomic resolution, time-resolved XRD techniques have emerged as an appropriate method in experimental physics. The UDKM1DSim toolbox provides methods to simulate the static and transient XRD response of crystalline sample structures. Due to the limitation to 1D sample structures only symmetrical X-ray diffraction in co-planar geometry is implemented. For the calculation of static XRD curves ($\theta/2\theta$-scans) for homogeneously strained layers two different theoretical approaches are provided: kinematical and dynamical XRD. In kinematical XRD theory ($xrd\text{kin}$) the incident X-ray beam is unaffected by the crystal, since absorption and multiple reflections are neglected. In the $xrd\text{kin}$ class no refraction correction has been implemented so far. However, the kinematical theory is a rather fast analytical approach for thin crystal layers, ideally imperfect mosaic crystals, and diffraction at the wings of Bragg peaks. For high quality crystals, thick crystals, and diffraction close to the maximum of strong Bragg peaks, so-called dynamical XRD theory ($xrd\text{dyn}$) should be considered. Dynamical XRD theory accounts for absorption, refraction, scattering, and multiple reflections (extinction) of the incident beam. In comparison to kinematical theory, dynamical XRD is generally slower to calculate due to its complex matrix formalism. However, in order to calculate the transient XRD response of a 1D sample structure due to ultrafast lattice dynamics only dynamical theory is implemented in the UDKM1DSim toolbox, since here its matrix formalism has no disadvantageous against the kinematical theory in terms of computational time. Examples on the applications and limitations of the two models are given in the example file $xrd\text{example.m}$.

For both theories the smallest scatterers in each structure are the individual atoms, whose scattering cross sections are given by the atomic form factor $f_a$. Generally, these atomic form factors dependent on the energy $E$ and scattering vector $q_s = 2k \sin(\theta)$ of the incident X-ray beam, where $k = 2\pi/\lambda$ is the X-ray wave number and $\theta$ is the incidence angle:

$$f(q_s, E) = f_C(q_s) + \delta f_1(E) - i f_2(E).$$  

(18)
The dispersion corrections $\delta f_1(E)$ and absorption correction $f_2(E)$ have been experimentally determined whereas the angle-dependence $f_{	ext{cu}}(q_z)$ is a theoretical correction from Hartree-Fock calculations. The values of $f(q_z, E)$ are automatically loaded and calculated from the according parameter files by the uDkM1Dsim toolbox for each atom/ion for a given $E$ and $q_z$ and the reader may refer to the documentation of the a&bse class for further details. In order to account for the polarization of the X-rays one has to introduce a $\theta$-dependent polarization factor $P(\theta)$ in kinematical and dynamical XRD calculations given by:

$$P(\theta) = \begin{cases} \frac{1}{\cos(2\theta)} & \text{s-polarized} \\ \frac{1}{1+\cos(2\theta)} & \text{p-polarized} \\ \frac{1}{2} & \text{unpolarized} \end{cases}$$

(19)

\subsection*{2.5.1. Kinematical XRD}
For the calculation of rocking curves using kinematical theory one further introduces the structure factor of a unit cell

$$S(q_z, E, \epsilon) = \sum_{i} f_i e^{-i q_z z_i(\epsilon)} .$$

(20)

The structure factor $S(q_z, E, \epsilon)$ is the summation of all atomic form factors $f_i(q_z, E)$ in a specific unit cell and also depends on the lattice strain $\epsilon$ by the position $z_i(\epsilon)$ of the individual atoms in the unit cell. From Ref. [12] one can now calculate the diffracted wave field amplitude at the detector from a single layer of similar unit cells as follows:

$$E_p = \frac{i e^2}{\varepsilon_0 m_e c_0} \frac{P(\theta) S(q_z, E, \epsilon)}{A q_z} ,$$

(21)

with $e$ as electron charge, $m_e$ as electron mass, $c_0$ as vacuum light velocity, $\varepsilon_0$ as vacuum permittivity, and $A$ as area of the unit cell in the plane normal to $q_z$. For the case of N similar planes of unit cells one can then write:

$$E_p^N = \sum_{j=0}^{N-1} E_p e^{i q_z z_j} ,$$

(22)

where $z$ is the distance between the planes (c-axis of the unit cells). The above equation can be simplified to

$$E_p^N = E_p \psi(q_z z, N) ,$$

(23)

introducing the interference function

$$\psi(q_z z, N) = \sum_{n=0}^{N-1} e^{i q_z z_n} = 1 - e^{i q_z z N} .$$

(24)

The total reflected wave field $E_p^T$ of all $i = 1 \ldots M$ homogeneous layers is the summation of the individual wave fields $E_p^{N,i}$:

$$E_p^T = \sum_{i=1}^{M} E_p^{N,i} e^{i q_z z_i} ,$$

(25)

where $Z_i = \sum_{j=i}^{N-1} (N_j z_j)$ is the distance of the $i^{th}$ layer from the surface. Finally, the actual reflectivity of the sample structure is calculated by $R = E_p^T (E_p^T)^*$. In order to obtain the static kinematical diffraction curve of a given sample structure one can follow the code listing below:

```matlab
% set the simulation parameters
E = 8047*u.eV; % X-ray energy
pol = 0.5; % mixed X-ray polarization
theta = (22:0.001:26)*u.deg; % angular range
% initialization of XRDkin simulation
K = XRDkin(S,forceRecalc,E,pol);
% set the qz-range by a theta-vector
K.setQzByTheta(theta);
% calculate the static diffraction curve:
Rs = K.homoeneousReflectivity();
```

\subsection*{2.5.2. Dynamical XRD}
In dynamical XRD theory a complex matrix formalism is applied to calculate the reflection and transmission of X-rays by individual atomic layers forming the sample structure. The basic building blocks for this formalism are the reflection-transmission matrices of the atomic planes

$$H = \frac{1}{\tau} \begin{pmatrix} \tau^2 - \rho^2 & \rho \\ -\rho & 1 \end{pmatrix} ,$$

(26)

and propagation matrices

$$L = \begin{pmatrix} \exp(i\phi) & 0 \\ 0 & \exp(-i\phi) \end{pmatrix} .$$

(27)

The matrix elements are defined as follows:

$$\rho = -i 4 \pi r_e f(q_z, E) P(\theta) e^{-M} ,$$

(28)

$$\tau = 1 - i 4 \pi r_e f(0, E) e^{-M} ,$$

(29)

$$\phi = \frac{q_z d}{2} ,$$

(30)

where $r_e$ is the classical electron radius, $M = (dbf q_z^2)/2$ with dbf $= \langle u^2 \rangle$ as average thermal vibration of the atoms (Debye-Waller factor), and $d$ is the distance between two layers of scattering objects.

In order to obtain the final reflectivity of the sample structure one has to carry out the according matrix multiplications of the $H$ and $L$ matrices. The reflectivity-transmission matrix (RTM) of a single unit cell $M_{\text{RT}}$ is calculated from the individual $H_i$ of each atom and the propagation matrices between the atoms $L_i$:

$$M_{\text{RT}} = \prod_i H_i L_i .$$

(31)

For $N$ identical layers of unit cells one can calculate the $N^{th}$ power of the unit cell’s RTM ($M_{\text{RT}}^N$) instead of carrying out $N$ matrix multiplications in order to save computational time. The RTM for the homogeneous sample...
The RTM of the inhomogeneous sample $M_{RT}^{\text{inhom,tot}}$ consisting of $K$ homogeneous substructures then becomes:

$$M_{RT}^{\text{inhom,tot}} = \prod_{k=1}^{K} \left( M_{RT}^{(k)} \right)^{N_k} .$$

(32)

For the case of an inhomogeneously strained sample one has to carry out the matrix multiplication for each individually strained unit cell. Thus, the RTM of the inhomogeneous sample $M_{RT}^{\text{inhom,tot}}$ containing $m = 1 \ldots M$ unit cells is calculated by:

$$M_{RT}^{\text{inhom,tot}} = \prod_{m=1}^{M} M_{RT}^{(m)} ,$$

(33)

which is a rather expensive calculation since it has to be carried out for all differentially strained types of unit cells, for all $\theta$ or $q_z$, and for all time steps. The final reflectivity $R$ of the sample is the calculated from the matrix elements of the $2 \times 2$ RTM matrix as follows:

$$R = \left| \frac{M_{RT}^{(1,2)}}{M_{RT}^{(2,2)}} \right|$$

(34)

In the following code listing we refer again to the results of the heat and phonon simulations for the given sample structure $S$ introduced above. For the static case the syntax for kinematical and dynamical XRD is similar. However, the simulation of UXRD from transient lattice dynamics which inevitably involves inhomogeneously strained layers is only implemented in the XRDdyn class.

```matlab
% initialization of XRDkin simulation
D = XRDdyn(S, forceRecalc, E, pol);
% set the qz-range by a theta-vector
D.setQzByTheta(theta);
% calculate the static diffraction curve:
Rh = D.homogeneousReflectivity();
% calculate a reduced number of strains per unique unit cell in order to save computational time
strainVectors = ...;
P.getReducedStrainsPerUniqueUnitCell(strainMap);
% calculate the transient XRD:
R = D.getInhomogeneousReflectivity(...
strainMap, strainVectors, type);
```

3. Examples

In this section we want to provide physical examples for the application of the UDkm1DSim toolbox. The complete example code can be found in the `/examples/` folder.

3.1. Bragg-Peak Splitting Evidences Inhomogeneous Expansion

Here we consider a 95 nm metallic SRO thin film on a dielectric STO substrate which is photoexcited by an ultrashort laser pulse with a fluence of $F = 20 \text{mJ/cm}^2$. The excitation is modeled as instantaneous temperature jump and we further neglect heat diffusion. The temperature change at $t = 0$ is shown in Fig. 1a) and features an exponential decay in the absorbing SRO layer in accordance with Eq. 1. Subsequent coherent phonon dynamics are calculated by the phononum class and the resulting spatio-temporal strain profile is depicted in Fig. 1b).

![Figure 1](image)

Figure 1: (Color online) a) Temperature change in the SRO thin film after excitation at $t = 0$. b) Spatio-temporal strain profile due to optical excitation of the SRO film. The SRO/STO interface is at $z = 95 \text{nm}$.

Using the result of the phononum simulation as input for the dynamical XRD calculations (XRDyn) we obtain the UXRD response of the ultrafast excitation of the SRO layer which is shown in Fig. 2 as a waterfall plot. Here, the SRO Bragg peak splits up due to the excited lattice
dynamics and does not continuously shift. Details for this example simulation and comparison to experimental data can be found in Ref. [15].

3.2. Superlattice Oscillations

In this example a superlattice (SL) structure is excited by an ultrashort laser pulse with a fluence of \( F = 30 \text{ mJ/cm}^2 \). The SL consists of 11 double layers (DL) each of which is composed of 20 unit cells of SRO and 38 unit cells of STO. The SL is grown on an STO substrate. The excitation is again modeled as instantaneous temperature jump at \( t = 0 \) neglecting thermal transport. The temperature profile after excitation is shown in Fig. 3a). The comb-like temperature profile originates from the alternating metallic and dielectric layers in the SL and exhibits an exponential decay towards the substrate. Due to the excitation profile, a longitudinal optical SL phonon mode, also known as zone-folded longitudinal acoustic phonon (ZFLAP), is excited which results in the complex spatio-temporal strain pattern shown in Fig. 3 b). Here, the strain oscillation directly indicates the frequency of the optical phonon mode.

The SL structure also results in complex static XRD signatures as can be seen in Fig. 4. This static diffraction curve is calculated by the \( \text{xrd} \) class which allows to access also the individual diffraction curves of the repeated substructures. The equidistant Bragg peaks originate from the SL structure and are numerated as SL\( i \). The most intense Bragg peak is the STO substrate reflection.

The transient X-Ray diffraction calculations using the coherent phonon result as input feature intensity oscillations of the SL Bragg peaks due to the excited longitudinal optical SL phonon. The integrated intensities of the SL0 and SL+2 Bragg peaks are plotted as transients in Fig. 5. For the SL+2 peak a non-linear X-ray response is observed. Details on this simulation and a comparison to UXRD experiments are given in Ref. [16].

3.3. Quasi-Monochromatic Phonon Wave Packet

In the last example a thin 15 nm SRO layer on an STO substrate is excited by a pulse sequence of 8 ultrashort laser pulses with a pulse separation of 7.2 ps in order to generate a coherent quasi-monochromatic phonon wave packet in the substrate. The average temperature in the SRO layer is plotted in inset of Fig. 6, where the excitation is again modeled as instantaneous temperature jump without heat diffusion. The corresponding transient strain pattern is calculated by the \( \text{phononNum} \) class including damping in the STO substrate. The average temperature in the SRO layer is plotted in inset of Fig. 6, where the excitation is again modeled as instantaneous temperature jump without heat diffusion. The corresponding transient strain pattern is calculated by the \( \text{phononNum} \) class including damping in the STO substrate.
4. Conclusions

The UDKM1Dsim toolbox enables the user to easily build 1D crystalline structures on the atomic-level using a rich database of element-specific physical parameters. The excitation and thermal transport in such 1D structures is calculated within the frame of an N-temperature model. The results are then plugged into an analytical or numerical model for evaluating the dynamics of coherent longitudinal acoustic phonon in the structure. Kinematical and dynamical XRD theory are provided to further calculate the static rocking curves of the structures for symmetrical Bragg reflections in coplanar diffraction geometry. The transient XRD response of the structures due to coherent phonon dynamics is evaluated exclusively by dynamical XRD theory.

The UDKM1Dsim toolbox is programmed fully object-orientated and highly modular in order to allow for user-defined inputs at any step of the simulation procedure. Hence the toolbox is not only applicable for the comparison of experimental UXRD data to the introduced theoretical models but also as an educational/theoretical test ground for students and researchers in the scientific field of ultrafast structural dynamics and ultrafast XRD.


