Quick tutorial for `almaBTE`

`almaBTE` consists of a library and a collection of executables allowing the user to calculate thermal transport properties of heterogeneous structures, using only ab-initio data, and solving the phonon Boltzmann Transport Equation (BTE).

Features of `almaBTE 1.3` include

Supports a variety of material types

- crystals
- alloys
- superlattices

Enables steady-state transport simulations in multiscale structures

- bulk systems (thermal conductivity & capacity)
- thin films (effective thermal conductivity)
- multilayered structures (temperature profile, spectrally resolved heat flux, effective thermal conductivity)

Enables transient transport simulations in bulk systems

- single pulse temperature response
- mean square displacement of thermal energy

`almaBTE` includes all the capabilities of ShengBTE for single crystals (but in a fully new implementation with more efficient solution algorithms). `almaBTE` is fully written in `C++` and freely distributed as open source under `Apache 2.0 License`.

Installing and compiling `almaBTE`

`almaBTE` relies on several external libraries (`hdf5` , `boost` and `openMPI`) that must be installed before compiling the main program. Detailed instructions for each supported operating system can be found below.

Installation on Ubuntu Linux

The following instructions have been tried on Ubuntu 16.10, freshly installed from the official AMD64 desktop image. They should be easily trasferable to other Debian-based distributions, and provide useful hints as to the required packages for other Linux systems.

1. Install CMake and the HDF5 and Boost libraries

From a terminal window, run this command:

```
sudo apt-get install cmake libboost-all-dev libhdf5-dev
```

You will be asked for your password. The system will then download and install all required dependencies.

2. Create a build directory

Move into the directory containing the almaBTE source code and run the following commands:

```
mkdir build
cd build
cmake ..
```

3. Compile `almaBTE`

From the build directory, run the following command:

```
make all
```

Note: you can speed up the process with parallel compilation. For example, if your CPU has 4 cores, you can use

```
make -j 4 all
```

4. Run the set of unit tests

`almaBTE` can run a series of test to check that the main modules work correctly on your system. To do so, run the following command from the build directory:

```
make test
```

5. Add the `almaBTE` executables to your `$PATH` so they can be run from any directory

Open `~/.bashrc` in a text editor and add the following line:

```
export PATH=$PATH:"<alma_dir>/build/src"
```

where `<alma_dir>` must be replaced by the location of the main alma directory, for example `/home/my_user/alma`. After saving this change, reload the file with `source ~/.bashrc` or by opening a new shell.

Installation on macOS

`almaBTE` requires OS X Yosemite (10.10.5) or higher.

To begin the installation, open a `Terminal` window and perform the following steps.

1. Obtain `Xcode` command line tools

From the terminal window, type

```
xcode-select --install
```

A popup will appear, click `Install` and follow the graphic prompts.

2. Update the `Xcode` development environment

Obtain the current version of `Xcode` from the Mac App Store. You must have/create an Apple iTunes account to do so, but the download (about 4.5 GB, allow up to 2 hours) is free of charge. `Xcode` is required for correct functioning of `Homebrew` and proper compilation of `almaBTE`.

Once the download and installation are complete, open `Xcode` from the `Applications` folder. Accept the License Agreement, then close `Xcode` again.

3. Install the `Homebrew` package manager

Return to the `Terminal` window and type

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

4. Obtain `cmake` command line tools

```
brew install cmake
```

5. Obtain `HDF5` libraries

Note: as of this writing the latest `hdf5` package (1.10.2) was found to cause compilation issues on macOS. The problem can be avoided by installing `hdf5` version 1.08 as shown below.

```
brew install hdf5@1.8
sudo cp /usr/local/opt/hdf5@1.8/lib/* /usr/local/lib/.
sudo cp /usr/local/opt/hdf5@1.8/include/* /usr/local/include/.
```

6. Obtain `boost` and `openMPI` libraries

```
brew install boost-mpi
```

7. Compile `almaBTE`

Navigate to the `alma` main folder and execute the following

```
mkdir build
cd build
cmake .. -DCMAKE_BUILD_TYPE=RELEASE
make all
```

Note: you can speed up the process with parallel compilation. For example, if your CPU has 4 cores, you can use

```
make -j 4 all
```

8. Verify that `almaBTE` components function properly

Run the series of self-tests by executing

```
make test
```

9. Add the `almaBTE` executables to your `$PATH` so they can be run from any directory

```
sudo nano ~/.bash_profile
```

In the file, add the following line

```
export PATH="$PATH:<alma_dir>/build/src"
```

where `<alma_dir>` must be replaced by the location of the main alma directory, for example `/Users/my_user_name/alma`. Save the file using `ctrl-O` followed by `enter`, and close the editor with `ctrl-X`.

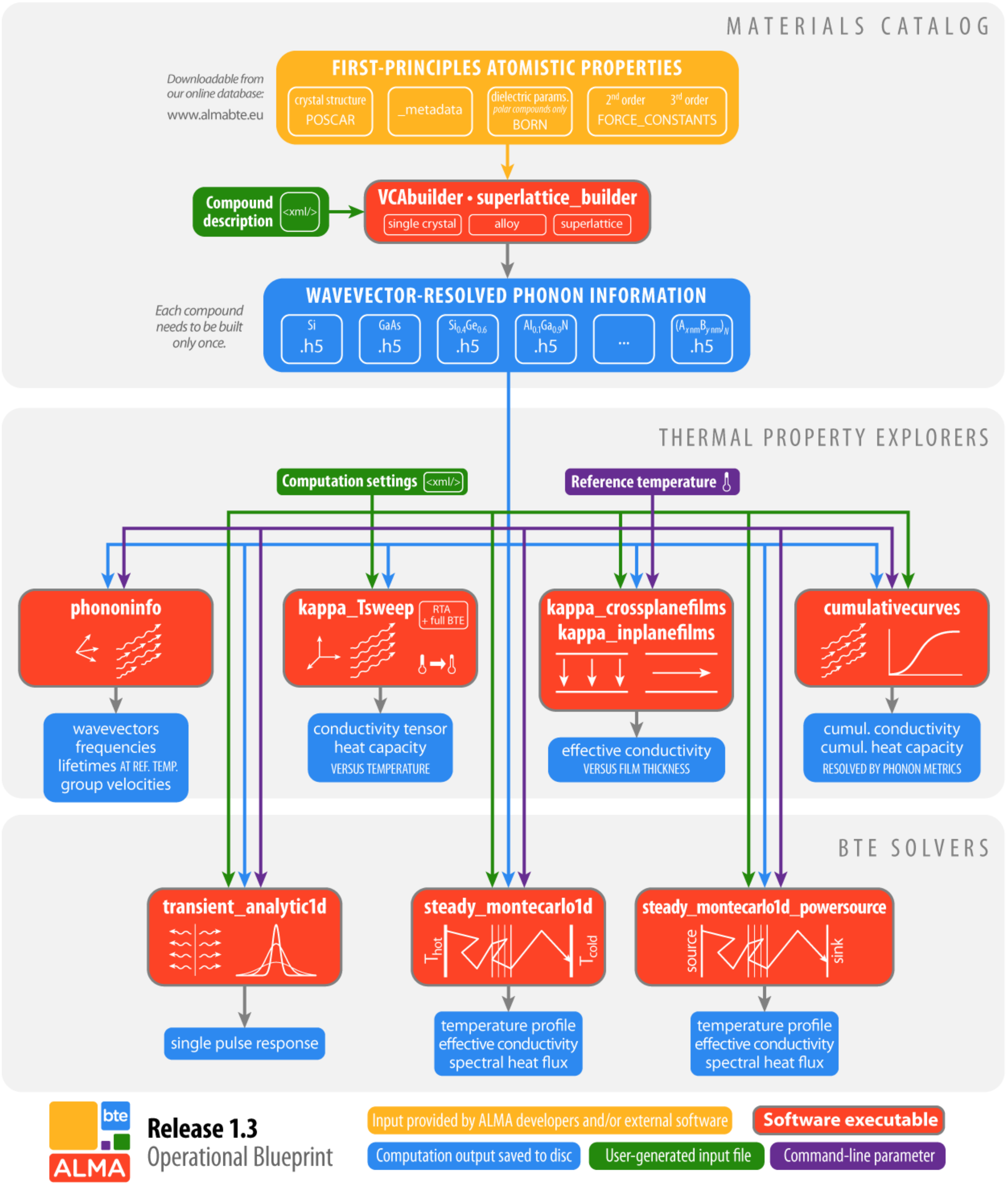
Finally, end the session by typing

```
exit
```

and close the `Terminal` window.

Main workflow of `almaBTE`

The workflow for performing thermal computations with `almaBTE` is graphically illustrated in the figure below (click for large version) and typically consists of the following two steps.


MATERIALS CATALOG
Downloadable from our online database: www.almabte.eu
FIRST-PRINCIPLES ATOMISTIC PROPERTIES
crystal structure POSCAR
_metadata
dielectric params. polar compounds only BORN
2nd order 3rd order FORCE_CONSTANTS
Compound description <xml/>
VCAbuilder • superlattice_builder
single crystal
alloy
superlattice
Each compound needs to be built only once.
WAVEVECTOR-RESOLVED PHONON INFORMATION
Si .h5
GaAs .h5
$Si_{0.4}Ge_{0.6}$ .h5
$Al_{0.1}Ga_{0.9}N$ .h5
...
$(A_{x\,nm}B_{y\,nm})_N$ .h5
THERMAL PROPERTY EXPLORERS
Computation settings <xml/>
Reference temperature
phononinfo
wavevectors frequencies lifetimes AT REF. TEMP. group velocities
kappa_Tsweep RTA + full BTE
conductivity tensor heat capacity VERSUS TEMPERATURE
kappa_crossplanefilms kappa_inplanefilms
effective conductivity VERSUS FILM THICKNESS
cumulativecurves
cumul. conductivity cumul. heat capacity RESOLVED BY PHONON METRICS
BTE SOLVERS
transient_analytic1d
single pulse response
steady_montecarlo1d
T_{hot}
T_{cold}
temperature profile effective conductivity spectral heat flux
steady_montecarlo1d_powersource
source
sink
temperature profile effective conductivity spectral heat flux
bte
ALMA
Release 1.3
Operational Blueprint
Input provided by ALMA developers and/or external software
Software executable
Computation output saved to disc
User-generated input file
Command-line parameter

1. Obtain phonon properties of the materials of interest

In this step you will use `VCAbuilder` and/or `superlattice_builder`. These executables use ab-initio source files (downloadable from our online materials database) to determine the phonon dispersions, group velocities, and scattering matrix for three-phonon processes. The phonon properties are computed over a discrete wavevector grid with user-supplied resolution and stored in `<material>.h5` files (`hdf5` format). For superlattices, the file also contains the phonon scattering rates induced by mass disorder and barriers.

The `hdf5` files only need to be created one single time for each material with a given grid resolution.

2. Solve the BTE for the structure of interest to extract thermal transport properties

For this step a variety of executables are available, each dedicated to perform specific simulations on bulk systems, thin films, and multilayer structures.

Thanks to the previously created `hdf5` repository, phonon properties of the constituting compounds do not have to be recomputed but can simply be loaded from the files.

The structure geometry and computational settings are controlled by the user via `xml` input files.

Using `almaBTE` : Overview of each executable

Below follows a brief description of the executables.

Executables marked with [mpi] are parallellised; speed up execution by running them on multiple cores.

For a detailed illustration of the `xml` input syntax and generated outputs, please refer to the xml examples manual .

`VCAbuilder` [mpi]

```
VCAbuilder <inputfile.xml>
```

This executable builds `hdf5` phonon property files for bulk materials.

Input

[Required] `xml` input file describing the desired material and grid resolution.

Several types of materials are supported:

- `<singlecrystal>` (e.g. Si, GaAs, InAs, GaN, ...)
- `<alloy>` (e.g. $Si_{0.5}Ge_{0.5}$, $In_{0.53}Ga_{0.47}As$, ...)
- `<parametricalloy>` Automatic batch creation of alloy families (e.g. $Si_{1-x}Ge_{x}$, $In_{x}Ga_{1-x}As$)

[Required] ab-initio source files downloadable from our online materials database

- `_metadata`
- `POSCAR`
- `FORCE_CONSTANTS`
- `FORCE_CONSTANTS_3RD`
- `BORN` *(polar compounds only)*

Alloy creation requires ab-initio files for each constituting compound, for example generation of $In_{0.53}Ga_{0.47}As$ needs files for InAs and GaAs.

Output

`h5` file(s) for the specified material(s)

`superlattice_builder` [mpi]

```
superlattice_builder <inputfile.xml>
```

This executable builds `hdf5` phonon property files for superlattices.

Input

[Required] `xml` input file describing the desired superlattice and grid resolution.

The superlattice is described in terms of its two consituting compounds, and the atomic concentration profile of a single period. (For details see the examples.)

[Required] ab-initio source files for the constituting compounds (see `VCAbuilder`).

Output

`h5` file for the specified superlattice.

`phononinfo`

```
phononinfo <materialfile.h5> <OPTIONAL:Tambient>
```

This executable writes phonon information associated with a previously generated `h5` file to a tabulated data file.

Input

[Required] Path to a previously generated `h5` file for the material of interest.

[Optional] Ambient temperature. Set to 300K when omitted.

Output

Comma-separated data file (including a single-line header) with extension `.phononinfo` containing the following phonon properties:

- q-point index nq [-]
- branch index nbranch [-]
- first wavevector coordinate qa [normalised to reciprocal lattice vector a]
- second wavevector coordinate qb [normalised to reciprocal lattice vector b]
- third wavevector coordinate qc [normalised to reciprocal lattice vector c]
- angular frequency omega [rad/s]
- volumetric heat capacity C [J/m^3-K] *evaluated at the specified Tambient*
- lifetime tau [s] *evaluated at the specified Tambient*
- x-component of group velocity vx [m/s]
- y-component of group velocity vy [m/s]
- z-component of group velocity vz [m/s]

`kappa_Tsweep`

```
kappa_Tsweep <inputfile.xml>
```

This executable computes the thermal conductivity tensor and volumetric heat capacity of bulk media as a function of ambient temperature.

Input

[Required] `xml` input file specifying the material of interest and desired temperature sweep.

[Required] previously generated `h5` file for the material of interest

Output

`csv` file containing the thermal properties

`cumulativecurves`

```
cumulativecurves <inputfile.xml> <OPTIONAL:Tambient>
```

This executable computes the contribution of different phonons to bulk thermal conductivity and heat capacity at a given temperature.

Input

[Required] `xml` input file specifying the material of interest, thermal transport axis, and desired quantities.

The phonon contributions to thermal conductivity and capacity can be resolved by a variety of properties:

- mean free path [m]
- 'projected' mean free path (MFP measured along the thermal transport axis) [m]
- angular frequency [rad/ps]
- frequency [THz]
- energy [meV]
- relaxation time [s]

[Required] previously generated `h5` file for the material of interest

[Optional] Ambient temperature. Set to 300K when omitted.

Output

File(s) in `csv` format containing the requested cumulative quantities.

`kappa_crossplanefilms`

```
kappa_crossplanefilms <inputfile.xml> <OPTIONAL:Tambient>
```

This executable performs semi-analytic computation of the cross-plane apparent conductivity in thin films as a function of thickness at a given temperature. The program is also capable to provide a compact parametric fitting to the computed conductivity curve.

Input

[Required] `xml` input file specifying the material of interest, film orientation, and desired thickness sweep

[Required] previously generated `h5` file for the material of interest

[Optional] Ambient temperature. Set to 300K when omitted.

Output

File in `csv` format containing the film conductivity, and optionally another `csv` file containing the parametric model parameters and curve.

`kappa_inplanefilms`

```
kappa_inplanefilms <inputfile.xml> <OPTIONAL:Tambient>
```

This executable performs semi-analytic computation of in-plane apparent conductivity in thin films as a function of thickness at a given temperature.

Input

[Required] `xml` input file specifying the material of interest, film orientation, and desired thickness sweep

[Required] previously generated `h5` file for the material of interest

[Optional] Ambient temperature. Set to 300K when omitted.

Output

File in `csv` format containing the film conductivity

`steady_montecarlo1d` [mpi]

```
steady_montecarlo1d <inputfile.xml> <OPTIONAL:Tambient>
```

This executable performs variance-reduced Monte Carlo simulations of one-dimensional transport in thin films and multilayer structures embedded between two isothermal reservoirs.

Input

[Required] `xml` input file specifying the structure geometry, materials, and simulation settings

The following quantities will be computed by default:

- temperature profile versus space
- average heat flux, and its stochastic tolerance
- system-wide effective thermal conductivity, and its stochastic tolerance
- system-wide effective thermal resistivity and conductance

Upon user request, the program will also determine heat flux resolved by phonon frequency at user-specified locations.

[Required] previously generated `h5` files for all materials present in the structure

[Optional] Ambient temperature. Set to 300K when omitted.

Output

`csv` file containing the computed temperature profile

`txt` file containing the thermal metrics

Optionally: `csv` files of spectrally resolved heat flux profiles

`steady_montecarlo1d_powersource` [mpi]

```
steady_montecarlo1d_powersource <inputfile.xml> <OPTIONAL:Tambient>
```

This executable performs variance-reduced Monte Carlo simulations of one-dimensional transport in thin films and multilayer structures with a planar power source at the top and ideal heat sink at the bottom.

Input

[Required] `xml` input file specifying the structure geometry, materials, and simulation settings

The following quantities will be computed by default:

- temperature profile versus space
- temperature rise at the heat source, and its stochastic tolerance
- system-wide effective thermal conductivity, and its stochastic tolerance
- system-wide effective thermal resistivity and conductance

Upon user request, the program will also determine heat flux resolved by phonon frequency at user-specified locations.

[Required] previously generated `h5` files for all materials present in the structure

[Optional] Ambient temperature. Set to 300K when omitted.

Output

`csv` file containing the computed temperature profile

`txt` file containing the thermal metrics

Optionally: `csv` files of spectrally resolved heat flux profiles

`transient_analytic1d`

```
transient_analytic1d <inputfile.xml> <OPTIONAL:Tambient>
```

This executable computes semi-analytic solutions of the one-dimensional time-dependent BTE in bulk media at a given temperature.

Input

[Required] `xml` input file specifying the material of interest, the thermal transport axis, and the desired time sweeps.

Several quantities of interest can be computed:

- single-pulse temperature profiles versus space at user-specified times (*only valid in weakly quasiballistic and diffusive regimes*[1])
- single-pulse temperature response at the heat source versus time (*only valid in weakly quasiballistic and diffusive regimes*[1])
- mean square displacement (MSD) of thermal energy versus time (*valid at all times, from fully ballistic to diffusive regimes*)

[1] Time values must exceed characteristic phonon relaxation times.

[Required] previously generated `h5` file for the material of interest

[Optional] Ambient temperature. Set to 300K when omitted.

Output

File(s) in `csv` format containing the requested transient solution(s)