Design of a Generic Crystal Structure Container Class

Mike McKerns

DANSE Software Workshop
Caltech Materials Science
Two Obvious Questions

What the heck is a crystal structure container class?

A container class is a way of organizing values that describe some desired system. A crystal structure container class thus should allow for a method of describing any desired crystal system.

Why bother?

In any major software project, the base container classes will hold the most reused portions of code. These containers, if well written, will provide the most robust and most efficient way of describing any given system. Further, the user or programmer will not have to repeatedly recode to describe a new type of simulated structure.
A Generalized Crystal Structure

The most general case allows for local regions of order & disorder, and local impurities.

We must construct a container that can successfully describe all cases simultaneously.
Regions of Order & Disorder

A disordered region cannot be further decomposed, and thus must be described only in the most brute-force manner.

However, an ordered region can be immediately broken into the underlying components that determine the periodic nature of the local system. This is, in the most general sense, a crystal primitive cell and the dimensions of the structure into which the primitive cell is repeated.
Crystal Primitive Cell: the Basic Unit

The most basic unit of an ordered region is the crystal primitive cell. The crystal primitive cell itself has two essential parts: the basis and the lattice primitive.

Shown here are a simple cubic lattice primitive with a two atom basis, and a hexagonal lattice primitive with a single atom basis.
Building a Lattice Primitive

A lattice primitive is built from a set of $N$ primitive vectors $P$ that define the underlying geometric structure of the crystal primitive cell.

The lattice primitive can be thought of as holding a transformation matrix $T$, which relates the external laboratory coordinate system $X$ to that of the crystal.

\[ P = T X, \text{ where } P_1 = T_{11}X_1 + T_{12}X_2 + T_{13}X_3 \]

or less generally, \[ a = T_{11}x + T_{12}y + T_{13}z. \]
Building a Basis

The other component of a crystal primitive cell is the basis. A basis is a representation of the positions of the unique periodic objects in the crystal.

The basis can be thought of as holding a coefficient matrix \( U \), which relates the positions of the basis objects to the primitive vectors \( P \).

\[
B = U P,
\]

where in the figure:

\[
B_1 = 0P_1 + 0P_2 + 0P_3
\]
\[
B_2 = 1/2P_1 + 1/2P_2 + 1/2P_3.
\]
But what about the Physics?

As defined here, a basis does NOT contain any information about what flavor of object sits at the vector head. Here, a basis is a placeholder that only maintains that 'something' is at the periodic positions within the crystal.

A basis will, however, be constructed to have the basis objects point to a table that maintains the properties used for the simulation. This is where some physics begins to creep in to the picture...
A Brief Aside to Discuss Properties

The property table is another container class that will be filled with properties that are utilized by the calculation algorithm, and holds a pointer that refers back to the basis in the crystal structure container class.

Important properties that may be held are: spin, scattering cross-section, NN distance, NN index, local displacement, ...
Utilizing the Crystal Primitive Cell

'Single object' properties (scattering cross-section, spin) are easily imported from a database or directly from calculation. However, 'dual object' properties (bonds, NN distance) need some initial calculation before they can be stored in a property table.

By temporarily mapping the crystal primitive cell neighborhood to build 'dual object' information, a set of rules for linking basis objects can be constructed.
Determining Relative Periodicity

A local neighborhood is constructed by temporarily copying the crystal primitive cell so as to form a number of complete surrounding layers.

Now 'dual object' properties for the basis objects in the central cell can be determined, by creating a relative index of all nearest neighbor basis objects.

This relative index can be then applied in turn to the local region of order.
Building a Crystal

A crystal is constructed to hold $M$ crystal primitive cells in a periodic arrangement ($l \times m \times n$ cells). Starting from an origin, an array is built that has a modulus of the dimensions of the crystal.

This allows the crystal to incorporate non-periodic properties, and essentially treats each individual crystal primitive cell as a unique entity within the crystal.
How a Crystal Container Works

A crystal container does NOT hold copies of the individual lattice primitive and basis, but only holds an index that determines the position of a particular crystal primitive cell within the crystal.

The real work is done by adding an extra dimension to the properties table, so both crystal and the crystal primitive cell point to the table.
The Power of Non-Periodicity

Most physical systems are not composed of a purely periodic basis, and contain imperfections and impurities of some sort.

Vacancies, substitutions, & dislocations can now be treated by localized change to the properties table, while the periodic index of the crystal is retained.
Simple Local Changes in Property

It is easy to visualize a local change in 'spin', as modification of a value in the property table, but what if the property table is used to store 'local displacement' from a lattice site?

Through 'local displacement', distortions corresponding to stress & strain caused by substitutions, vacancies (using a null atom), and dislocations can now be modeled.
Superimposing a Second Crystal

Interstitial objects can be treated by creating another separate crystal composed only of impurities. This new crystal may only have a single crystal primitive cell containing containing a single basis object, however, this architecture allows the old crystal to maintain a modulus indexing system.

Now, however, a method is needed to relate the position of the impurity crystal to the periodic crystal...
Applying Some Brute-Force

The crystal structure class at the top level is now utilized.

A crystal can now be related to another crystal by maintaining the position and orientation with respect to the crystal structure origin.
Class Map of Crystal Structure

CRYSTAL STRUCTURE

Crystal Primitive Cell
Lattice Primitive
Vector
Basis
Vector

Crystal Position & Orientation
* Index

* Index

... CRYSTAL#1

CELL#1

| basis_object1 | property1 | property2 | property3 | ...
|---------------|-----------|-----------|-----------|---
|               |           |           |           |   
| basis_object2 |           |           |           |   
|               |           |           |           |   
| basis_object3 |           |           |           |   
|               |           |           |           |   
...
Summary & Outlook

A crystal structure container class should provide the most robust and most efficient way of describing any given system. Further, the user or programmer will not have to repeatedly recode to describe a new type of simulated structure.

A first version of the code has been completed, but decisions still need to be made on how to implement the object-to-property relationship through the entire crystal structure. Also, sample cases (BVK & spin dynamics) need to be tested in this framework.