# Matlab Horace redesign

## Purpose

Current Horace has been written using Matlab without modern OOP capabilities. As the result the polymorphism in Horace is implemented using custom code, which makes changing Horace complicated due to high binding between different parts of code.

The purpose of the planned redesign would be:

1. Rebuild Horace according to modern OOP design
2. Separate large objects into sequence of the smaller one, interacting through well-defined interfaces.
3. Design these interfaces to satisfy user/developer requests namely:
4. Unified support for parallel operations over file based and memory based sqw and dnd objects.
5. Generic projection, allowing to make cuts in any (e.g. spherical or cylindrical, or q-E mixed) coordinate systems.
6. Split the further improvement among number of developers

## Workplan:

The table below outlies estimated development time to spend on various Horace development tasks and the dependencies between the tasks.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| W1 | W2 | W2 | W4 | W5 | W6 | W7 | W8 | W9 | W10 | W11 | W12 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| MPI frmwk investigation | | |  |  |  |  |  |  |  |  |  |  |
|  |  |  | Parallel C++ messaging framework | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  | compiled Horace | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| rewriting sqw/dnd as new type classes | | | |  |  |  |  |  |  |  |  |  |
|  |  |  |  | extracting projection interface | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  | spherical projection | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | cylindrical projection | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | symmetrisation | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | q-dE projection | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | hdf file format | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | Parallel cut | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | parallel unit operations/sqw eval | | | | |  |  |  |  |

## Main interfaces:

1. The interface supporting operations over the objects is outlined on the Figure 1



Figure General interface for accessing the objects and doing binary-unary operations.

The main methods, presented on the diagram are **load, save** and **operation managers**. The load method would be static method of generic interface, which generates objects, corresponding to the file types. An options on the method would be ‘-nopix’, which would create a file-based sqw object, where only main part of the object is loaded in memory and the pixels are left in the file.

Q: a) should the object become file-bound like an Mantid MD objects are? (New way of operating Horace)

b) Should we allow **save** operation to modify the pixels?

The **operationManager**-s would accept *’-parallel‘* option (or instance of parallel cluster) to allow running operations in parallel.

Specific operations over SQW or DND objects would be implemented in the terms of using **operationManager**-s to access image data of DND objects and pixels data of SQW objects.

1. DND object contains information related to processed neutron image. Suggested public interface to this object is presented on the Figure 2. The difference from current implementation would be an instance of aProjection class, accumulating physical information related to transformation of pixel information from crystal Cartesian coordinate system (lab frame) into the physical coordinate system on a neutron image. In more details this class would be discussed below.



Figure Public interface and composition of an DND object.

1. SQW object would contain a DND object, information on pixels (neutron events) presented in crystal Cartesian coordinate system and the information about the instrument where the pixels come from. Suggested composition for this class and public interface to its properties is presented on the Figure 3.



Figure The composition and the public interface of a SQW object

The main operation sqw object should provide would be get\_pixels operation. It would return pixels in the coordinate system, defined by DND object

1. The purpose of aProjection class, providing an interface and its children, providing particular implementation, is to describe the transformation, necessary for conversion of pixels from crystal Cartesian coordinate system into physical (e.g. hkl or spherical) coordinate system. The class needs to provide information, necessary for two main methods to work:
2. get\_pixels\_hkl – transformation from pixels expressed in crystal Cartesian coordinate system into user requested coordinate system
3. Get\_keys – (or get\_contributing\_bins) the method returns the coordinates of the pixels, which may contribute to a cut, expressed in different from current DND object coordinate system. The system may be a just rotated one (like existing cuts) or some different (e.g. spherical).

## File format

Additional work, adjacent to the redesign of the Horace object would be change internal Horace file format to HDF.

The pre-request for this change would be deep understanding of the ways a MPI application would read chunks of large pix array stored in HDF on a parallel file system. If sufficient performance for this operation is achieved, everything else would be trivial as all IO operations on SQW/DND objects are already performed using well-defined interface.



Figure Additions to existing sqw file access interface.

The preliminary work show that C++MPI reader combined with single writer provide performance comparable with similarly configured Matlab MPI binary parallel reader/ single writer. Single threaded C++ HDF reader has performance, comparable with Matlab binary reader. This means that we can deploy HDF file format for Horace files without substantial performance loss.