

dxhdf5: A Software Package for Importing HDF5 Physics Data into OpenDX[★]

Ireneusz Szcześniak^{a,c}, John R. Cary^{a,b}

^a*Center for Integrated Plasma Studies, University of Colorado
UCB 390, Boulder, Colorado 80309, USA*

^b*Department of Physics, University of Colorado
Boulder, Colorado 80309, USA*

^c*Institute of Theoretical and Applied Informatics, Polish Academy of Sciences
ul. Bałtycka 5, 44-100 Gliwice, Poland*

Abstract

A software package, `dxhdf5`, for importing HDF5 data into OpenDX is discussed. HDF5 is a hierarchical, self-describing data format that is rapidly becoming a standard for storing large, portable data sets, such as those generated through high-performance computing. OpenDX, formerly IBM Data Explorer, is a visualization package for examining data of any dimensionality. With `dxhdf5`, OpenDX users can easily visualize both field data on regular grids through the `ImportHDF5Field` module, and particle data through the `ImportHDF5Species` module.

Key words: `dxhdf5`; OpenDX; HDF5; field; particle; importing data

1 Introduction

Development of Vorpal [1], a relativistic, arbitrary dimensional, hybrid plasma and beam simulation code, is one of the current projects in the Plasma and Beam Physics Group at the Center for Integrated Plasma Studies. Vorpal profuse results are saved in HDF5 (Hierarchical Data File 5) files [2], because this format is capable of storing large amount of data. Vorpal users want to visualize their results in OpenDX, a free, open source, powerful visualization package [3].

[★] Sponsored by the National Science Foundation under grant no. 0113907.

Vorpal users faced two problems related to the OpenDX visualization of HDF5 data. First, as the HDF5 format is rather new (first released in 1998), OpenDX does not support it. Therefore HDF5 data had to be converted to the ASCII format (or any other format recognized by OpenDX) prior to importing. Second, OpenDX was unable to import large Vorpal data due to the operating memory requirements. To remedy this problem the data had to be filtered before the conversion to the ASCII format.

Thus to import HDF5 data in OpenDX it was required to run specially written filter programs to select the data, convert the selected data to the ASCII format, and finally import the ASCII data in OpenDX. This arrangement is acceptable for a small number of times the data needs importing. When this need becomes frequent, the overhead of work done manually and the possibility of making mistakes during this process is prohibitive.

For Vorpal users the first remedy to the problem of data conversion was the `ImportHDF5` package [4] developed by Thomas Radke of the Max Planck Institute for Gravitational Physics. This package, however, does not fully meet the needs of Vorpal users: importing data that correspond to a field's slab is not supported and it is also impossible to import particle data.

Therefore the `dxhdf5` package [5] has been developed to furnish a unified way of importing HDF5 data directly into OpenDX, without preprocessing. Currently, with the use of `dxhdf5`, importing HDF5 data is fast, easy and reliable. OpenDX users can easily visualize both field data on regular grids through the `ImportHDF5Field` module, and particle data through the `ImportHDF5Species` module.

Our software transforms HDF5 data into OpenDX objects. HDF5 files store data in *datasets*, which are arrays of an arbitrary number of dimensions, while OpenDX offers *fields* as the objects for storing data to be visualized. Therefore the `dxhdf5` modules create OpenDX fields from HDF5 datasets.

The package extends the OpenDX capabilities by providing two *modules*, which are fully integrated with OpenDX and therefore easy to use. The modules are available in the graphical user interface and in the script mode. The package is closely bound with OpenDX and HDF5, since it serves as an interface between them, and for this reason `dxhdf5` cannot be used with other visualization environments.

The article is organized as follows: Section 2 presents the `ImportHDF5Field` module, Section 3 present the `ImportHDF5Species` module, while Section 4 concludes the article.

2 ImportHDF5Field

The `ImportHDF5Field` module imports data of a field defined on a regular and uniform grid of any number of dimensions and in either single or double floating point precision. The user is allowed to import all of the field data or only the fraction corresponding to a slab. Selecting a slab can considerably reduce OpenDX memory requirements as only a portion of a very large set of data is loaded into the operating memory.

The number of dimensions of the created OpenDX field is equal to the number of dimensions of the HDF5 dataset. There is no limit on the number of dimensions or their sizes imposed by the module – these are limited only by the capabilities of HDF5 and OpenDX. Likewise, the performance of the module depends only on the performance of HDF5 and OpenDX.

The `ImportHDF5Field` module offers the capability of selecting a slab of the field, where you specify the coordinates of the lower corner, the thickness of the slab, and the stride (number of grid points to move in each dimension to get to the next grid point).

2.1 Acceptable File Structure

The HDF5 format is flexible and allows the same field data to be described in various ways. Therefore this section explains the HDF5 file structure accepted by the `ImportHDF5Field` module.

Data for a field are stored in one dataset. The maximal size in bytes of a dataset depends on the capabilities of an operating system and HDF5, not of `dxhdf5`. There might be several datasets in a file, so there might be several fields stored there, but the module imports one field only (it does not import a series of fields). The data class of the dataset's data type must be `H5T_FLOAT`, of which data types `NATIVE_FLOAT` and `NATIVE_DOUBLE` are examples.

The data of the dataset are not enough to create a field; there is still information needed about the field's *origin* and *delta*. This information must be provided in the dataset's attributes named “origin” and “delta” which must be vectors. The vector's data type must be of the class `H5T_FLOAT` with the number of elements at most equal to the number of dimensions of the field.

The “origin” attribute provides Cartesian coordinates of the first grid point. The first element of the vector provides the coordinate in the first dimension, the second element in the second dimension, etc.

Each entry in the “delta” attribute supplies information about the distance in Cartesian coordinates between grid positions in a specific dimension. The first element specifies the difference in the first dimension, the second element in the second dimension, etc.

2.2 Example

We used Vorpal to generate a file which contains the description of an electric field with two wave packets. The data are stored in a dataset of dimensions $200 \times 100 \times 3$ which describes a two-dimensional field of dimensions 200×100 grid points. Positions along the third dimension of the dataset contain three components of the electric field. The `h5dump` utility, which is provided together with the HDF5 library, reports the following structure of the file:

```
> h5dump -H samples/ImportHDF5FieldUsage.h5
HDF5 "ImportHDF5FieldUsage.h5" {
GROUP "/" {
  DATASET "YeeElecFieldData" {
    DATATYPE  H5T_IEEE_F32LE
    DATASPACE  SIMPLE { ( 200, 100, 3 ) /
                       ( 200, 100, 3 ) }
    ATTRIBUTE "time" {
      DATATYPE  H5T_IEEE_F32LE
      DATASPACE  SCALAR
    }
    ATTRIBUTE "origin" {
      DATATYPE  H5T_IEEE_F32LE
      DATASPACE  SIMPLE { ( 2 ) / ( 2 ) }
    }
    ATTRIBUTE "delta" {
      DATATYPE  H5T_IEEE_F32LE
      DATASPACE  SIMPLE { ( 2 ) / ( 2 ) }
    }
  }
}
```

The file structure complies with the stated requirements: there is a dataset defined with the data type of the class `H5T_FLOAT`, and the attributes “origin” and “delta” of the correct format are present. The additional attribute “time” is ignored.

Figures 1 and 2 show the values of one component of a field’s slab which contains one wave packet. Figure 1 depicts the field’s slab with all data available

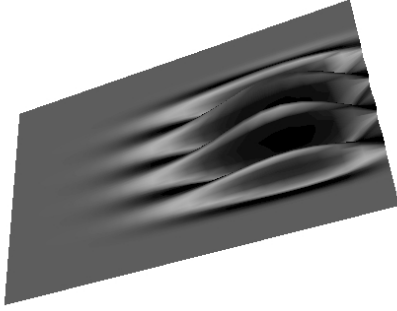


Fig. 1. Field at original resolution.

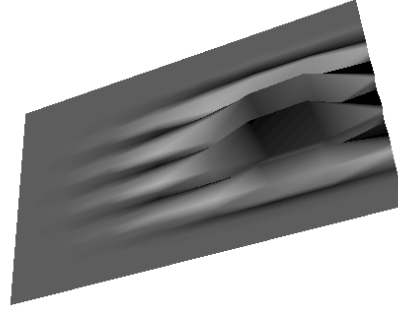


Fig. 2. Field at lower resolution.

from the HDF5 file, while Figure 2 shows the field at lower resolution, using only one-ninth of the HDF5 data (every third point in both dimensions taken).

3 ImportHDF5Species

The `ImportHDF5Species` module imports particle floating point data in single or double precision. A particle is described by its position (coordinates) and its data elements (typically velocities or weight). The number of dimensions of the created OpenDX field is equal to the number of coordinates of the particles in the HDF5 dataset. The module can import particles that have any number of coordinates and any number of data elements – these numbers are limited only by the capabilities of HDF5 and OpenDX.

Again, if needed, only a portion of a very large set of data is loaded into the operating memory. Particles can be selected randomly, periodically, or by limits (maxima and minima) on the particle coordinates and data elements. The module accepts conditions on up to three coordinates and up to three data elements. Particles with more than three coordinates can be loaded, but the conditions can be specified for the first three coordinates only. The same limitation applies to data elements.

3.1 *Acceptable File Structure*

This section defines the structure of the HDF5 file which is acceptable by the `ImportHDF5Species` module. Particle data must be stored in one dataset. The maximal size in bytes of a dataset depends on the capabilities of an operating system and HDF5, not of `dxhdf5`. There might be several datasets in a file, and therefore several particle sets stored there. The module imports particle data only from one dataset, which must be an array of dimensions (number of particles) \times (a particle's length). Particle positions and data ele-

ments must be of the same data type, which additionally must be of the class `H5T_FLOAT`. A particle's length is the number of elements that compose a particle. For instance, particles which have two coordinates and three velocity components, are of the length five.

The dataset must be supplied with the attribute "numSpatialDims" which must be of the *dataspace*¹ `H5S_SCALAR` and of the class `H5T_INTEGER`. The value of the attribute provides the number of a particle's coordinates. Other attributes may be defined for a dataset, but they are of no use for the module and therefore are ignored.

3.2 Example

Again, we employed Vorpil to produce a sample file, which contains a dataset with a description of electrons. Each electron is described by two spatial coordinates and three velocity components. The `h5dump` utility reports the following structure of the file:

```
> h5dump -H samples/ImportHDF5SpeciesUsage.h5
HDF5 "samples/ImportHDF5SpeciesUsage.h5" {
GROUP "/" {
  DATASET "ImportHDF5SpeciesUsage.h5" {
    DATATYPE  H5T_IEEE_F32LE
    DATASPACE SIMPLE { ( 5889, 5 ) / ( H5S_UNLIMITED, 5 ) }
    ATTRIBUTE "numSpatialDims" {
      DATATYPE  H5T_STD_I32LE
      DATASPACE SCALAR
    }
  }
}
```

The file structure complies with the previously stated requirements: there is a dataset defined with the data type of the class `H5T_FLOAT`, and the attribute "numSpatialDims" of the correct format is present.

The easiest way to use the module is only to supply the file name to the module, in which case all the particles from the dataset will be imported as shown in Figure 3. Figure 4 shows randomly selected particles (roughly 750 particles), and the particles with $x_0 > 10^7$, where x_0 is the horizontal velocity component of a particle.

¹ Please consult the HDF5 documentation for the definition.

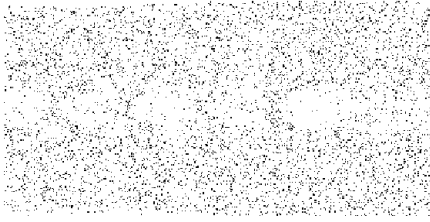


Fig. 3. All particles.

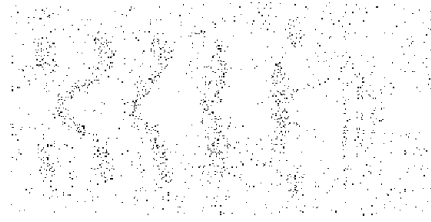


Fig. 4. Selected particles.

4 Conclusion

The package proves useful for the Vorpall users and for users outside of the plasma physics domain. John Stark of the Southampton Oceanography Centre, UK, reports using a customized version of `dxhdf5` to visualize OCCAM [6] data which are saved in numerous files of 3.5GB size, totaling to 500GB of data. Andreas Klaedtke of the University of Surrey, UK, uses a customized version of `dxhdf5` to import results of his simulations of a microdisk cavity with a laser active material inside.

The package successfully functions not only on desktop computers running the Linux, Mac OS X, or Sun Solaris operating systems but also on large mainframes that operate under the control of the AIX or IRIX operating systems. The portability of `dxhdf5`, which allows OpenDX to import the same HDF5 data on various computer platforms, is achieved by harnessing free GNU tools: Autoconf and Automake.

`dxhdf5` is an open source project and is available under a very liberal license and so can be incorporated into any other packages. Anyone wishing to extend the functionality of the package is welcome to contribute. The package along with documentation can be obtained from its website [5], from where it is downloaded several times a week.

References

- [1] <http://www-beams.colorado.edu/vorpall>
- [2] <http://hdf.ncsa.uiuc.edu/HDF5>
- [3] <http://www.opendx.org>
- [4] <http://www.cactuscode.org/VizTools/OpenDX.html>
- [5] <http://www-beams.colorado.edu/dxhdf5>
- [6] <http://www.soc.soton.ac.uk/JRD/OCCAM>