Jem/Jive 2.1

Release Notes

Dynaflow Research Group

11-6-2015

1 Overview

Version 2.1 of Jem/Jive is a minor release with some new features, improvements and bug fixes. This version should be source compatible with version 2.0.

The following list provides an overview of the new features and improvements. More detailed information can be found in the following sections.

- Support for quadrilateral elements with 12 nodes and cubic shape functions. Use the class `Quad12` to create a `Shape` object for calculating the shape functions.
- Support for reading and writing HDF5 files.
- Extension of the `Properties` class with operators for getting and setting properties.
- Switch from headers provided by the standard C library to the equivalent headers provided by the standard C++ library.
- Some code changes to better conform to the recent C++ standard and the latest C++ compilers.
- Support for creating static libraries using the Makefiles provided by Jem and Jive.
- Support for the `clang` compiler.
- Better support for recent versions of MacOS X.
- Bug fix in the `ArclenModule` so that it properly handles linear constraints.
- Various small Windows-specific bug fixes.
- Some internal changes and code re-organisations to increase code re-use.

2 Support for HDF5 files

The new Jem package `hdf5` provides an interface for creating, modifying and reading HDF5 data files. The interface automatically translates Jem objects into equivalent HDF5 objects. See `www.hdfgroup.org/HDF5/` for more information about HDF5.

3 Extension of the `Properties` class

The `Properties` class from Jem implements the `[]` (subscript) operator that can be used to both get and set properties. Here is an example:

```
Properties  p;
String      s;
double      x;
int         i;

s = "hello";
x = 0.0;
i = 1;

p["first"]  = s;
p["second"] = x;
p["third"]  = i;

s = p["first"];
x = p["second"];
i = p["third"];
```

The operator calls are automatically translated to the equivalent calls the `get` and `set` member functions.

4 Create static libraries

Use the `lib.mk` Makefile to build a static library instead of an executable. Here is an example Makefile:

```
library = femutil

include $(JIVEDIR)/makefiles/packages/fem.mk
include $(JIVEDIR)/makefiles/lib.mk
```