**A PDESolve Interpreter in Python**

*Chris Myers*
*Beam Technologies, Inc., 110 N. Cayuga St., Ithaca, NY 14850*
*myers@beamtech.com*

PDESolve is a C++ class library for formulating and solving partial differential equations (PDEs) that has been under development at Beam Technologies, Inc., for the past several years.[1] Furthermore, PDESolve serves as a computational substrate for a web-based, collaborative engineering environment that Beam is developing. Recently, we have begun building an interpreted Python interface to PDESolve by wrapping pieces of the library using Beazley's SWIG package[2] to build a Python extension module. In a manner reminiscent of the way that MATLAB was originally developed to provide an interpreted, higher-level interface to standard linear algebra libraries (LINPACK and EISPACK), we aim to provide more flexible and convenient control of objects relevant to PDEs (functions, domains, differential and integral operators, boundary conditions, meshes, solvers, etc.). The PDESolve C++ library defines classes and methods which present a symbolic user interface that enables concise formulation of PDE-based problems. Replicating that user interface to a reasonable approximation within the interpreted Python environment is important, as is leveraging relevant Python modules developed elsewhere. For example, novel approaches to automatic derivation of sensitivity equations (describing the sensitivity of PDE-based models to tunable design parameters) may be possible by combining flexible automatic differentiation schemes written in Python[3] with the inherent symbolic capabilities of PDESolve. This sort of approach can ultimately pave the way to rapid prototyping of methods for PDE-based optimization and control.

## References

[1] Beam Technologies, Inc., PDESolve, http://www.beamtech.com/products/pdesolve

[2] D.M. Beazley, SWIG: http://www.cs.utah.edu/ beazley/SWIG/swig.html

[3] K. Hinsen, Scientific Python Modules (Derivatives.py), http://starship.skyport.net/crew/hinsen/Derivatives.py