

# CONICAL: The Computational Neuroscience Class Library

Joseph J. Strout  
University of California, San Diego  
3033 Basic Sciences Building  
9500 Gilman Drive  
La Jolla, CA 92093-0608  
jstrout@ucsd.edu

## Introduction

Compartmental modeling of biological neurons and neural networks plays an important role in computational neuroscience. Typically, this modeling is accomplished with monolithic Unix-based simulation packages, e.g., *Genesis* (Wilson *et al.*, 1989) or *Neuron* (Hines 1989). These packages are powerful research tools, and well suited for an experienced user with access to an appropriate computer platform. However, such packages suffer several drawbacks for other users: they can be difficult to learn; are not easy to modify, extend, and integrate with other software; and are generally not portable to non-Unix/DOS platforms.

To meet these different requirements, a C++ class library was created. CONICAL, the Computational Neuroscience Class Library, consists of a non-monolithic hierarchy of C++ classes. The library can be used in whole or in part as needed. Each object class is designed to be readily extended or modified through the inheritance mechanism; complex classes are derived from base classes in several steps, allowing behavior to be efficiently overridden at any point. The library has been co-developed and tested with Unix, DOS, and MacOS compilers. C++ constructs which tend to be problematic (e.g., templates and exception handling) have been avoided to ensure maximum portability among various platforms and compilers.

## Applications

CONICAL can be used with a very simple main program, in a manner similar to the script-based simulators mentioned above. In direct comparisons, it was found that CONICAL required roughly the same amount of user-written code as the equivalent *Genesis* script. Moreover, since CONICAL is written in C++, learning a new language is not necessary for many users. In this usage, output is sent to the screen or a file for graphing or analysis by other software, and the complete program can be fully cross-platform compatible.

CONICAL can also be embedded in larger software packages, such as graphical simulators, educational packages, or plug-in modules for general-purpose mathematics software. This will allow programmers with little knowledge of compartmental modeling techniques, or physiologists with only modest programming skill, to produce accurate and efficient neural simulations, with a wide range of applications in research and education.

## Methods

CONICAL consists of approximately 20 classes. The hierarchy tree has four roots, with the longest branch measuring five classes. Multiple inheritance is applied wherever needed to segregate functionality. Two main branches dominate the hierarchy: one develops compartments, ending in a standard cylindrical compartment; the other contains currents, including voltage- and ligand-gated channels. The library includes a standard Hodgkin-Huxley channel, whose kinetics are set by selecting an equation form and parameters, as well as a standard alpha-function synapse. A more sophisticated Markov-model synapse is also included, as are current injectors and data-recording objects. Numerical integration is done by the exponential Euler method. Documentation is provided in hypertext format via the World-Wide Web.

## Conclusion

CONICAL is not intended as a replacement for current simulation packages; rather, it serves a complimentary purpose. It may be the preferred simulation environment in cases where (1) the user prefers C++ to a new script language; (2) considerable modification or extension of the simulator is planned; (3) the user needs to embed a neural model within other software; or (4) simulation is needed on platforms other than Unix or DOS/Windows. The CONICAL library, including source code, may be freely used for non-commercial purposes. It may be obtained via the World-Wide Web<sup>1</sup>, or directly from the author.

## Acknowledgements

This work was supported in part by an NSF Predoctoral Fellowship.

## References

- Hines, M. (1989). A program for simulation of nerve equations with branching geometries. *International Journal of Biomedical Computing*, 24, 33-68.
- Wilson, M. A., Bhalla, U. S., Uhley, J. D., and Bower, J. M. (1989). GENESIS: A system for simulating neural networks. In D. Touretzky (Ed.), *Advances in Neural Information Processing Systems* (pp. 485-492). San Mateo, CA: Morgan Kaufmann.

<sup>1</sup> <http://www-acsc.ucsd.edu/~jstrout/conical/>