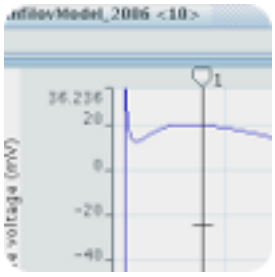
Welcome to CESE

Table of contents

Simulogic Inc. provides commercial products and services tailored to CESE platform, including ready-to-use models for CESE, custom-developed models, extensions to the CESE environment, and additional support and training.

- **CESE Plus 2.0 is here!**



CESE Plus

Full support for MacOS X in addition to Windows and Linux, innovative split display for comparison of multiple simulated data traces, data import/export in MS Excel, Axon text files and ASCII tables, powerful cursors to select regions of interest and perform online measurements and statisical analysis, VirtuClamp to simulate voltage clamp and current clamp protocols, and much more.

Take a video tour of CESE Plus 2.0.

Analyze action potential parameters with Action potential analysis plugin for CESE Plus.

Action potential clamp is now one click away with CESE Plus and HEKA stimulus plugin.

- **CESE 1.4.7** (December 2007): Plot types can be changed, performance improvements, and bugfixes.
- **CESE 1.4.6** (February 2007): Improved result printing, export to scalable vector graphics (SVG), additional simulation options, improved continuous simulations, and many bugfixes.
- **CESE 1.4.5** (September 2006): Switched to Java 1.5 (5.0), massive updates to multithreading improve stability, many updates to cursors and data analysis, optimized plots, and much more.

Cell Electrophysiology Simulation Environment (CESE) is a comprehensive framework specifically designed to perform computational electrophysiological simulations, for example, simulations of cardiac myocyte electrical activity. CESE is useful for simulations of action potentials, individual ionic currents, and changes in ionic concentrations.

CESE is a cross-platform program, it runs on any system that has Java runtime environment (JRE) version 1.5 (5.0) or above. It was tested on Windows, Linux, Solaris, MacOS X, and

AIX.

This document outlines the main features of CESE. They are presented for the two major domains: Users and Developers.

1. CESE users

CESE is an integrated environment for performing computational simulations using a variety of electrophysiological models.

At this stage CESE allows creation and execution of the single-cell models (containing both Hodgkin-Huxley (HH) and Markovian current formulations). Models of electrical activity of cardiac myocytes with source code are included in the CESE distribution. We hope to extend the number of available models, and add certain neuronal models in the future.

The main strength of CESE is in its uniformity — a program interface remains the same for different types of models. You can easily switch between models and compare simulation outputs. Model parameters can be modified, selected for output and/or clamped in the same, standard way.

CESE extends the conventional electrophysiological meaning of the "voltage clamp". You can clamp virtually any model variable, including voltage (membrane potential), total or individual ionic currents, ionic concentrations, temperature, gating variables, etc. The clamping commands can be complex piece-wise functions, individually set for the model variable of interest. This opens endless possibilities for the exploration of complex model behavior.

CESE provides simple, but efficient data visualizations. Simulation results can be presented in the graphic and tabulated forms. Plots can be customized, and regions of interest zoomed.

Even though CESE was not designed to be a data analysis tool, you can generate current-voltage relationships (I-Vs) and calculate statistical parameters for a given signal within the program. You can export your data to ASCII, Axon Text File (ATF), and NetCDF formats to continue analysis in your favorite package.

2. CESE developers

CESE was created from the ground up to incorporate the best programming practices available to Java developers, both in terms of user interface consistency and code clarity and reuse. Wherever possible, CESE rely on available Java APIs (for example Java2D, JavaBeans, JAXP) to simplify the code.

Model creation requires a number of house-keeping functions to be coded — these include

ODE integrators, routines for handling model parameters, saving/restoring model state, visualizing simulation results, etc. CESE provides you with implementation for these routines, hence, you can concentrate on writing the code for concrete ionic current(s), and CESE will handle the rest.

CESE is not trying to create complicated programming frameworks on its own — rather, it utilizes core Java APIs. For example, models are Java components conforming to the JavaBeans specification. We use XML to specify clamping commands, and Java object serialization to save/restore model parameters.

CESE is an open-source project and created with a code reuse in mind. We encourage you to extend CESE and contribute back to the project. Mail support@simulogic.com with feedback and questions. Visit our SourceForge page to obtain access to the bug database, mailing lists, CVS repository, files, and forums.

3. References

- Vascotto S, Missan S, La C
  CESE Plus: a computational approach for accelerating electrophysiology research.
  Laboratory Focus, 13(4):8-11, 2009.
- Missan S, McDonald TF
  CESE: Cell Electrophysiology Simulation Environment.
  Appl Bioinformatics, 4(2):155-6, 2005.

If you used CESE for your research, please let us know and we will add your paper to the list.

1. Garny A, Nickerson DP, Cooper J, Weber dos Santos R, Miller AK, McKeever S, Nielsen PM, Hunter PJ
   CellML and associated tools and techniques.
   Philos Transact A Math Phys Eng Sci 13;366(1878):3017-43, 2008.
2. Hurmusiadis V
   Virtual Heart: Simulation-Based Cardiac Physiology for Education.
   Computers in Cardiology, 34:65-68, 2007.
3. Moukabary T, Haines DE
   Relationship between the Potassium Currents Block and the Occurrence of Early after Depolarizations in the Setting of Sodium Current Blockade.
   Computers in Cardiology, 34:361-362, 2007.
4. Petsios SKD, Fotiadis DI
   A Computational Framework for the Analysis of Biological Models.
   Engineering in Medicine and Biology Society, 1101-1104, 2007.
5. Sun Z, Finkelstein A, and Ashmore J

Using Ontology with Semantic Web Services to Support Modeling in Systems Biology. in Web Information Systems Engineering - WISE 2007 Workshops, ISSN 0302-9743 (Print) 1611-3349 (Online), Volume 4832/2007, 2007.

6. Wilders R
   Computer modelling of the sinoatrial node.
   Med Biol Eng Comput, 45(2):189-207, 2007.
7. Cooper J, McKeever S, and Garny A
   On the application of partial evaluation to the optimisation of cardiac electrophysiological simulations.
   Proceedings of the 2006 ACM SIGPLAN symposium on Partial evaluation and semantics-based program manipulation, 2006.
8. Hunter PJ
   Modeling Human Physiology: The IUPS/EMBS Physiome Project.
   Proceedings of the IEEE, 94(4):678-691, 2006.
9. Nickerson D, Hunter P
   The Noble cardiac ventricular electrophysiology models in CellML.
   Prog Biophys Mol Biol, 90(1-3):346-59, 2006.
10. Nickerson D, Nash M, Nielsen P, Smith N, and Hunter P
    Computational multiscale modeling in the IUPS Physiome Project: Modeling cardiac electromechanics.
    IBM Research: Functional Genomics and Systems Biology, 50(6):617, 2006.
11. Sarai N, Matsuoka S, and Noma A
    simBio: a Java package for the development of detailed cell models.
    Prog Biophys Mol Biol, 90(1-3):360-77, 2006.
12. Shimayoshi T, Komurasaki K, Amano A, Iwashita T, Matsuda T, and Kanazawa M
    A Method to Support Cell Physiological Modelling Using Description Language and Ontology.
    IPSJ Digital Courier, 2:726-735, 2006.
13. Takao S, Kazuhiro K, Akira A, Takeshi I, Masanori K, Tetsuya M
    Computer support for physiological cell modelling using an ontology on cell physiology.
    Conf Proc IEEE Eng Med Biol Soc., 1:4171-4, 2006.
14. Zaniboni M, Cacciani F, and Groppi M
    Effect of input resistance voltage dependency on DC estimate of membrane capacitance in cardiac myocytes.
    Biophys J., 89(3):2170-81, 2005.