Chornoboy ES, Schramm LP, Karr AF (1988) Maximum likelihood identification of neural point process systems. Biol Cybern 59:265–275

Czanner G, Eden UT, Wirth S, Yanike M, Suzuki WA, Brown EN (2008) Analysis of between-trial and within-trial neural spiking dynamics. J Neurophysiol 99:2672–2693

Daley D, Vere-Jones D (2003) An introduction to the theory of point processes, vol I & II. Springer, New York

DeGroot M, Schervish M (2012) Probability and statistics, 4th edn. Pearson, Upper Saddle River

Kass RE, Kelly RC, Loh WL (2011) Assessment of synchrony in multiple neural spike trains using loglinear point process models. Ann Appl Stat 5:1262–1292

Kim S, Putrino D, Ghosh S, Brown EN (2011) A Granger causality measure for point process models of ensemble neural spiking activity. PLoS Comput Biol 7:e1001110

McCullagh P, Nelder A (1989) Generalized linear models. Chapman & Hall, London

Ogata Y (1988) Statistical models for earthquake occurrences and residual analysis for point processes. J Am Stat Assoc 83:9–27

Okatan M, Wilson MA, Brown EN (2005) Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity. Neural Comput 17:1927–1961

Paninski L (2004) Maximum likelihood estimation of cascade point-process neural encoding models. Network 15:243–262

Shimazaki H, Amari S, Brown EN, Grun S (2012) State-space analysis of time-varying higher-order spike correlation for multiple neural spike train data. PLoS Comput Biol 8:e1002385

Truccolo W, Eden UT, Fellows MR, Donoghue JP, Brown EN (2005) A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. J Neurophysiol 93:1074–1089

GENESIS, the GEneral NEural SImulation System

James M. Bower[1], Hugo Cornelis[2] and David Beeman[3]

[1]Department of Computer Science, University of California, Santa Cruz, Santa Cruz, CA, USA
[2]University of Texas, San Antonio, TX, USA
[3]University of Colorado, Boulder, CO, USA

Synonyms

GEneral NEural SImulation System

Definition

GENESIS (The *GEneral NEural SImulation System*) is a simulation environment for constructing realistic models of neurobiological systems at many levels of scale including subcellular processes, individual neurons, networks of neurons, and neuronal systems (Wilson et al. 1989; Bower 1992; Bower and Beeman 2006; Beeman 2013). First released to the public in July 1990, it was one of the first simulation systems specifically designed for modeling nervous systems and also one of the first open-source software projects in computational biology. As a consequence, the GENESIS software system has benefited from many different contributors and has been responsible for many technical innovations and advancements.

Detailed Description

GENESIS is written in C and was originally developed for UNIX platforms (Wilson et al. 1989), although it now also runs under Mac OS and with Windows under the Cygwin environment. Originally developed in the author's laboratory at Caltech, the system was first released in June of 1988 in association with the first annual Methods in Computational Neuroscience Course at the Marine Biological Laboratory in Woods Hole, MA and then released as full source code to the public via anonymous FTP in July 1990. GENESIS version 2.0 was released in August 1995, with the latest update (version 2.3) released in May 2006 (available through the web site http://sourceforge.net/projects/genesis-sim). GENESIS Version 3.0 (G-3) is currently in beta release (http://genesis-sim.org) and represents a backwards compatible but complete restructuring of the software system.

From the outset, GENESIS design has predicated on several assumptions regarding computational neuroscience. These include: (1) the construction of computer models based on actual anatomy and physiology would be essential to understanding the computational organization of the nervous system (Bower 1991b, 1992, 2005, 2013; Bower and Beeman 2007);

(2) understanding nervous system function would depend on being able to construct and link models at many different levels of biological scale (Bower 1991a). (3) growth of the modeling system would depend on the ability of individual modelers to develop and share model features and components; (4) the system should be as machine independent as possible; (5) successful use would depend on a graphical interface supporting users with a range of computer expertise (Uhley et al. 1989; Bhalla 1998); (6) considerable time, effort, and resources would need to be committed to user support; and (7) in order to build modeling expertise within the neuroscience community GENESIS should support use in education as well as research (Fig. 1 from Bower and Beeman 1994).

The commitment of the GENESIS project to educational use is reflected in its first release in the inaugural year of the Methods in Computational Neuroscience course at the Marine Biological Laboratory in Woods Hole, MA, as well as in

subsequent courses offered in Europe, Asia, and Latin America. The tutorials developed for those courses provided the basis for "The Book of GENESIS" published in two editions (Bower and Beeman 1994, 1998). That book (now available as a free download: http://www.genesis-sim. org/GENESIS/bog/bog.html), was written both to support undergraduate and graduate education in computational neuroscience as well as an introduction to modelers interested in using GENESIS for research.

The development of educational tutorials has continued, with web-based tutorials for both GENESIS 2.3 and G-3. The GENESIS Documentation Home Page (http://genesis-sim.org/userdocs/ documentation-homepage/documentation-homepage.html) contains links for documentation and tutorials for both versions. The "Ultimate GENESIS Tutorial Distribution" (http://genesis-sim.org/ GENESIS/UGTD.html) is a complete self-paced GENESIS modeling course, combined with GENESIS 2.3 distributions and installation instructions.

GENESIS, the GEneral NEural SImulation System, Fig. 1 Original cover for the Book of GENESIS (first edition) (used with permission Bower and Beeman 1994)

It provides an easy way to get everything for GENESIS 2.3 by downloading a single package.

As one of the first neural simulations systems, the GENESIS project has also had a long history of innovation in the application of numerical simulation techniques to computational neuroscience. For example, GENESIS was one of the first open software systems in computational biology (Wilson et al. 1989); was the first simulation system supporting realistic modeling across a wide range of scales from large scale networks (Wilson and Bower 1989, 1991, 1992) to single neurons (De Schutter and Bower 1992) to subcellular (Blackwell et al. 2013); has fostered the application of new numerical simulation techniques for neurobiological modeling (Vanier and Bower 1999; Eichler West et al. 1997), was the first simulation system to promote quantifying parameter searches and modeling results (Bhalla and Bower 1993; Eichler West et al. 1997; Vanier and Bower 1996, 1999; Baldi et al. 1998); was one of the first computational modeling systems to be adapted for use on parallel computers (Bower et al. 1988; Nelson et al. 1989; De Schutter and Bower 1992), and was the first system to link geographically remote computers together in a single neuronal simulation (Leigh et al. 1993). The GENESIS project was also the first to propose standards against which to judge the speed and reliability of neural simulators (Bhalla et al. 1992, 1993), and was one of the first simulation systems to adopt the Python object-oriented declarative scripting language (Vanier 2001 thesis). From a user perspective, the GENESIS group was the first to develop a graphical interface specifically for use by neurobiologists for modeling (Uhley et al. 1990; Bower and Hale 1991; Forss et al. 1999), was the first to use virtual reality visualization techniques (Leigh et al. 1995), and was the first to use the Internet in support of intercommunication between neurobiologists and their modeling efforts (Leigh et al. 1995). The GENESIS project has pioneered the development of tools to convert neurobiological data into usable forms for models (West et al. 1997) and also pioneered the use of modeling databases as sources of biological data (Hucka et al. 2002). From the outset, the GENESIS development group has also promoted and supported efforts to provide interoperability between neural simulators (Beeman and Bower 2004; Cannon et al. 2007; Crook et al. 2013), beginning with the Modelers Workspace project (www.modelersworkspace.org), which drafted and promoted the first markup language for realistic neural modeling (Goddard et al. 2001). That initial effort subsequently provided the foundation for the development of NeuroML (Crook et al. 2007; Gleeson et al. 2010) (http://neuroml.org), which, in turn, has fostered the development of simulator interoperability tools including neuroConstruct (Gleeson et al. 2007).

A Modeler's View of GENESIS

From the standpoint of a modeler who is implementing a neural model with GENESIS version 2.3 or 3.0, the object oriented interface presented to the user is one of its most useful features. The design of GENESIS simulations and their Graphical User Interfaces are based on a "building block" approach. Simulations are constructed from modules that receive inputs, perform calculations on them, and then generate outputs. Model neurons are constructed from compartments to which are added variable conductance ion channels forming multi-compartmental neurons of any desired level of complexity. Neurons can then be linked together to form neural circuits of any size. A high level scripting language is used to recognize commands for combining these objects into neuron, network or system level models, and to optionally create a custom GUI for the simulation, as shown in Figs. 2 and 3. GENESIS has also been adapted to modeling subcellular structures as well (Blackwell et al. 2013).

This object-oriented approach is central to the generality and flexibility of the system, as it allows modelers to easily exchange and reuse models or model components. In addition, it makes it possible to extend the functionality of GENESIS by adding new commands or simulation components to the simulator, without having to modify the GENESIS base code itself.
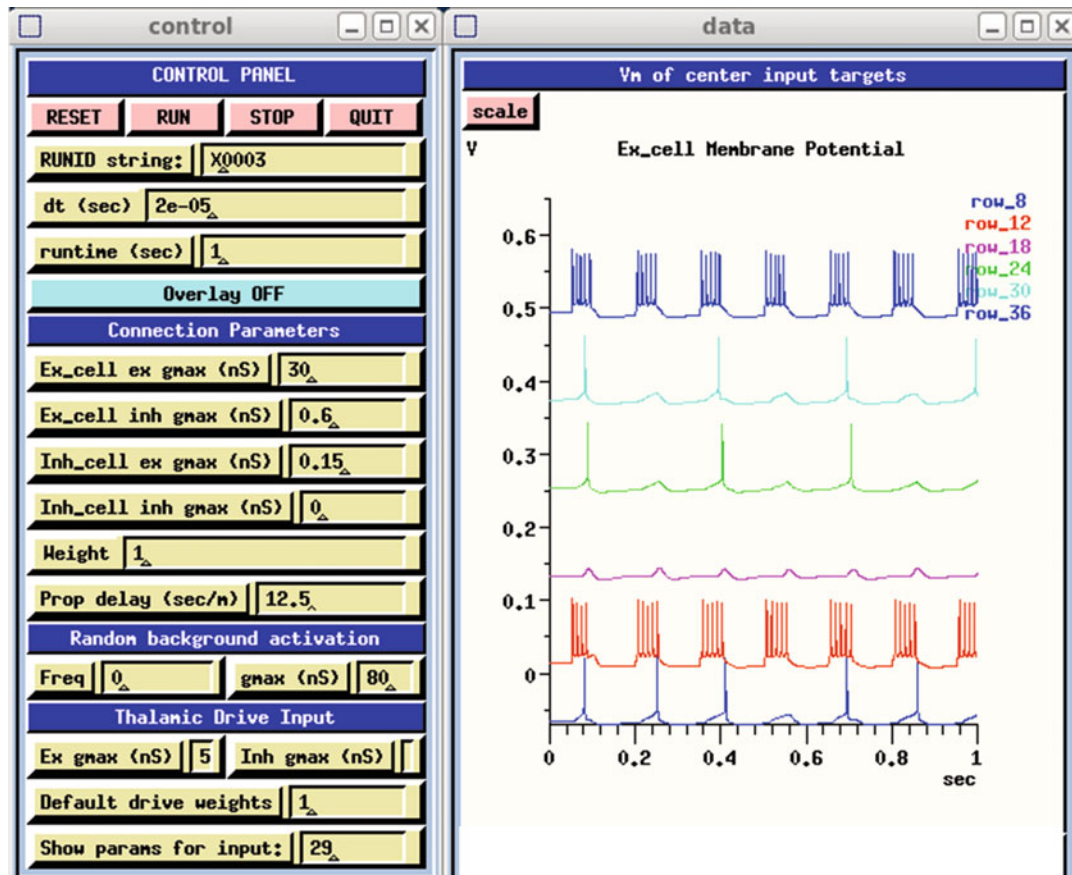
GENESIS, the GEneral NEural SImulation System, Fig. 2 A GUI for running a network simulation scripted with GENESIS 2.3

GENESIS 3 (G-3)

After the release of GENESIS 2.3, development efforts shifted to re-implementing GENESIS as a true twenty-first century simulation system (Beeman 2013), while preserving the most desirable features of the present version. This effort, known as GENESIS 3.0, or G-3, is intended to take simulator interoperability as well as flexibility in simulator design, development and use to a new level. With G-3 the GENESIS development group has completely restructured the historical monolithic core of GENESIS and reimplemented the system as a collection of independent software components. Based on the modular CBI architecture (Cornelis et al. 2012a), G-3 consists of a set of largely independent components which can be used either individually or in combination to perform the functions desired in running a particular simulation. Keeping with the original design objectives for the GENESIS project, this modularization considerably enhances the ability to run models across different levels of biological scale, and greatly facilitates interactions between those levels.

By separating user-specified model description from their machine level implementation, G-3 simplifies the development of new mathematical solvers and other run-time software components. This separation also allows for an assessment of model structure independent of its algorithmic implementation, providing among other advantages, the opportunity to build a new form of model-based publication system (Cornelis et al.
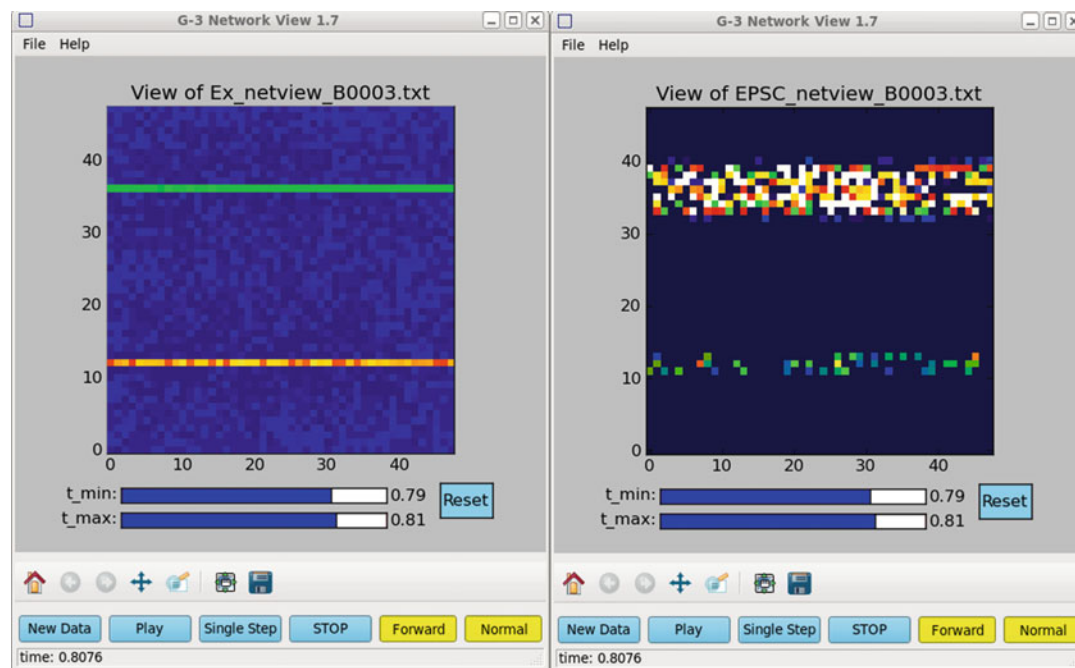
GENESIS, the GEneral NEural SImulation System, Fig. 3 View of the network membrane potential and summed excitatory post-synaptic currents produced by the simulation in Fig. 1. This was produced by the G-3 Python 'netview' tool

2010). Through its modularization G-3 allows simulations to be run locally and/or distributed over networks (Coop et al. 2010). The use of multiple parsers for scripting simulations allows G-3 to maintain backwards compatibility with GENESIS 2.3, and to support the use of scripts written in modern scripting languages such as Python. The modular CBI architecture also provides a template for interacting with and even incorporating other software systems (Cornelis et al. 2012b). As the first truly modular neural simulation system, G-3 provides a plastic and extensible framework that is designed to represent the next step in evolution of simulation systems for computational neuroscience (Brette et al. 2007; Beeman 2013).

References

Baldi P, Vanier MC, Bower JM (1998) On the use of Bayesian methods for evaluating compartment neural models. J Comput Neurosci 5:285–314

Beeman D (2013) History of neural simulation software, chapter 3. In: Bower JM (ed) 20 years of computational neuroscience, Springer series in computational neuroscience, vol 9, pp 33–77. https://doi.org/10.1007/978-1-4614-1424-7_3

Beeman D, Bower JM (2004) Simulator-independent representation of ionic conductance models with ChannelDB. Neurocomputing 58–60:1085–1090

Beeman D, Bower JM, De Schutter E, Efthimiadis EN, Goddard N, Leigh J (1997) The GENESIS simulator-based neuronal database, chap 4. In: Koslow SH, Huerta MF (eds) Neuroinformatics: an overview of the human brain project. Lawrence Erlbaum Associates, Mahwah, pp 57–80

Bhalla US (1998) Advanced XODUS techniques. In: Bower JM, Beeman D (eds) The book of GENESIS: exploring realistic neural models with the GEneral NEural SImulation System, chap 22, 2nd edn. Springer, New York, pp 381–405

Bhalla US, Bower JM (1993) Exploring parameter space in detailed single neuron models: simulations of the mitral and granule cells of the olfactory bulb. J Neurophysiol 69:1948–1965

Bhalla US, Bilitch DH, Bower JM (1992) Rallpacks: a set of benchmarks for neuronal simulators. Trends Neurosci 15:453–458

Bhalla U, Bilitch D, Bower JM (1993) Rallpacks: a set of benchmarks for neuronal simulators. In: Bower JM,

Eeckman F (eds) Computation and neural systems 1992. Kluwer Press, Boston, pp 133–140

Blackwell KT, Wallace LJ, Kim B, Oliveira RF, Koh W (2013) Modeling spatial aspects of intracellular dopamine signaling. Methods Mol Biol 964:61–75

Bower JM (1991a) Relations between the dynamical properties of single cells and their networks in piriform (olfactory) cortex. In: McKenna T, Davis J, Zornetzer S (eds) Single neuron computation. Academic Press, San Diego, pp 437–462

Bower JM (1991b) Exploring biological neural networks using realistic computer simulations. Naval Res Rev XLIII:17–22

Bower JM (1992) Modeling the nervous system. Trends Neurosci 15:411–412

Bower JM (2005) Looking for newton: realistic modeling in modern biology. Brains, Minds and Media 1: bmm217, (urn:nbn:de:0009-3-2177)

Bower JM (2013) Community models in Computational Neuroscience: The cerebellar Purkinje cell. In: Bower JM (ed) 20 years of progress in Computational Neuroscience. Springer, Berlin

Bower JM, Beeman D (1994) The book of GENESIS: a workbook of tutorials for the GEneral NEural SImulation System. Springer, New York

Bower JM, Beeman D (1998) The book of GENESIS: exploring realistic neural models with the GEneral NEural SImulation System, 2nd edn. Springer, New York

Bower JM, Beeman D (2006) Constructing realistic neural simulations with GENESIS. In: Crasto C (ed) Neuroinformatics. Humana Press, Totowa

Bower JM, Beeman D (2007) Constructing realistic neural simulations with GENESIS. Methods Mol Biol 401: 103–125. PMID: 18368363

Bower JM, Hale J (1991) Exploring neuronal circuits on graphics workstations. Sci Comput Autom:35–45

Bower JM, Nelson ME, Wilson MA, Fox GC, Furmanski W (1988) Piriform (Olfactory) cortex model on the hypercube. In: Proceedings of 3rd conference on hypercube concurrent computers & applications. ACM, New York, pp 977–999

Brette R, Rudolph M, Carnevale T, Hines M, Beeman D, Bower JM, Diesmann M, Morrison A, Goodman PH, Harris FC, Zirpe M, Natschläger T, Pecevski D, Ermentrout B, Djurfeldt M, Lansner A, Rochel O, Vieville T, Muller E, Davison AP, El Boustani S, Destexhe A (2007) Simulation of networks of spiking neurons: a review of tools and strategies. J Comput Neurosci 23:349–398. https://doi.org/10.1007/s10827-007-0038-6

Cannon R, Gewaltig M, Gleeson P, Bhalla U, Cornelis H et al (2007) Interoperability of neuroscience modeling software: current status and future directions. Neuroinformatics 5:127–138

Coop AD, Cornelis H, Santamaria F (2010) Dendritic excitability modulates dendritic information processing in a Purkinje cell model. Front Comput Neurosci 4:6

Cornelis H, Coop AD, Bower JM (2010) Development of model-based publication for scientific communication. BMC Neurosci 11(Suppl 1):P69. https://doi.org/10.1186/1471-2202-11-S1-P69

Cornelis H, Coop AD, Bower JM (2012a) A federated design for a neurobiological simulation engine: the CBI federated software architecture. PLoS One 7: e28956. https://doi.org/10.1371/journal.pone.0028956

Cornelis H, Rodriguez AL, Coop AD, Bower JM (2012b) Python as a federation tool for GENESIS 3.0. PLoS One 7(1):e29018. Epub 2012 Jan 20. PMID: 22276101

Crook S, Gleeson P, Howell F, Svitak J, Silver R (2007) MorphML: level 1 of the NeuroML standards for neuronal morphology data and model specification. Neuroinformatics 5:96–104. https://doi.org/10.1007/s12021-007-0003-6

Crook S, Davison AP, Plesser HE (2013) Learning from the past: approaches for reproducibility in computational neuroscience, chapter 4. In: Bower JM (ed) 20 years of computational neuroscience, Springer series in computational neuroscience, vol 9. Springer, pp 33–77. https://doi.org/10.1007/978-1-4614-1424-7_4. (This volume)

De Schutter E, Bower JM (1992) Purkinje neuron simulation on the Intel Touchstone Delta with GENESIS. In: Mihaly T, Messina P (eds) Proceedings of the grand challenge computing fair. CCSF publishing/Caltech, Pasadena, pp 268–279

Forss J, Beeman D, Bower JM, Eichler West RM (1999) The modeler's workspace: a distributed digital library for neuroscience. Future Gener Comp Syst 16: 111–121

Gleeson P, Steuber V, Silver R (2007) NeuroConstruct: A tool for modeling networks of neurons in 3D space. Neuron 54:219–235

Gleeson P, Crook S, Cannon RC, Hines ML, Billings GO, Farinella M, Morse TM, Davison AP, Ray S, Bhalla US, Barnes SR, Dimitrova YD, Silver RA (2010) NeuroML: a language for describing data driven models of neurons and networks. PLoS Comput Biol 6(6):e1000, 815. https://doi.org/10.1371/journal.pcbi.1000815

Goddard NH, Hucha M, Howell F, Cornelis H, Shankar K, Beeman D (2001) Towards NeuroML: model description methods for collaborative modelling in neuroscience. Phil Trans R Soc Lond Ser B 356:1209–1228. https://doi.org/10.1098/rstb.2001.0910

Hucka M, Shankar K, Beeman D, Bower JM (2002) The modeler's workspace: making model-based studies of the nervous system more accessible, chap 5. In: Ascoli G (ed) Computational neuroanatomy: principles and methods. Humana Press, Totowa, pp 83–115

Leigh J, De Schutter E, Lee M, Bhalla U, Bower J, DeFanti TA (1993) Realistic modeling of brain structures with remote interaction between simulations of an inferior olivary neuron and a cerebellar Purkinje cell. In: Proceedings of the SCS simulations multiconference, Arlington

Leigh J, Vasilakis A, DeFanti TA, Grossman R, Assad C, Rasnow B, Protopappas A, De Schutter E, Bower JM (1995) Virtual reality in computational neuroscience. In: Earnshaw RA, Vince JA, Jones H (eds) Virtual reality applications. Academic Press, London, pp 293–306

Nelson M, Furmanski W, Bower JM (1989) Simulating neurons and neuronal networks on parallel computers, chap 12. In: Koch C, Segev I (eds) Methods in neuronal modeling. MIT Press, Cambridge, MA, pp 397–438

Uhley JD, Bhalla US, Wilson MA, Bilitch DH, Nelson ME, Bower JM (1989) An interpretive tool for creating and manipulating widgets and their attributes under XODUS: an X-based output and display utility for simulations. In: Proceedings of Xhibition '89, San Jose

Uhley J, Wilson MA, Bhalla U, Bower JM (1990) A unix, X-windows-based neural network simulation system. USENIX '90, conference proceedings

Vanier M (2001) Realistic modeling of the olfactory system. Ph.D. thesis, California Institute of Technology

Vanier MC, Bower JM (1996) A comparison of automated parameter-searching methods for neural models. In: Bower J (ed) Computational neuroscience: trends in research 1995. Academic Press, New York, pp 477–482

Vanier MC, Bower JM (1999) A comparative survey of automated parameter-search methods for compartmental neural models. J Comput Neurosci 7:149–171

West RME, Wilcox GL, Bower JM, De Schutter E (1997) Mesh generation software and numberical convergence analysis for the production of morphometrically realistic neuron models. University of Minnesota-IBM shared research report. UM-IBM 97/7

Wilson MA, Bower JM (1989) The simulation of large scale neural networks, chap 9. In: Koch C, Segev I (eds) Methods in neuronal modeling. MIT Press, Cambridge, MA, pp 291–333

Wilson M, Bower JM (1991) A computer simulation of oscillatory behavior in primary visual cortex. Neural Comput 3:498–509

Wilson M, Bower JM (1992) Cortical oscillations and temporal interactions in a computer simulation of piriform cortex. J Neurophysiol 67:981–995

Wilson MA, Bhalla US, Uhley JD, Bower JM (1989) GENESIS: a system for simulating neural networks. In: Touretzky D (ed) Advances in neural information processing systems. Morgan Kauffman, San Mateo, pp 485–492

GHK Current Equation

▶ Goldman-Hodgkin-Katz Equations

GHK Voltage Equation

▶ Goldman-Hodgkin-Katz Equations

Gillespie Algorithm for Biochemical Reaction Simulation

Daniel T. Gillespie
Dan T Gillespie Consulting, Castaic, CA, USA

Synonyms

Stochastic Simulation Algorithm (SSA)

Definition

The Gillespie Algorithm, also known as the Stochastic Simulation Algorithm (SSA), is a computer-oriented procedure for simulating the changes in the molecular populations of chemical species in a chemically reacting system. The algorithm requires the reactant molecules, typically solute molecules in a sea of many much smaller solvent molecules, to be dilute and well-mixed throughout the containing volume. In contrast to the traditional differential equations of chemical kinetics, which imposes not only those requirements but also the requirement that the molecular populations be very large, the SSA simulates the occurrence of individual reaction events in a way that properly reflects their inherent randomness. That randomness is often important for the relatively low molecular populations that commonly occur in cellular systems.

Since the middle of the nineteenth century, ordinary differential equations (ODEs) have been the standard mathematical tool for modeling the way in which molecular populations in chemically reacting systems evolve in time. The broad success of that modeling strategy is, however, a little surprising: ODEs prescribe a behavior that is