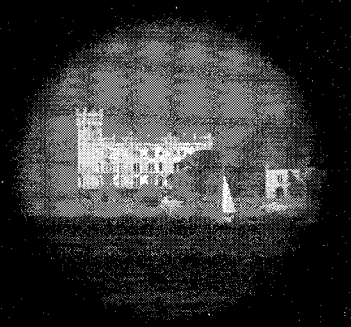
the
abdus salam
international
centre
for theoretical
physics

united nations
educational, scientific
and cultural
organization

international atomic
energy agency



preprint

CLASSICAL ALGORITHMS FOR AUTOMATED PARAMETER-SEARCH METHODS IN COMPARTMENTAL NEURAL MODELS - A CRITICAL SURVEY BASED ON SIMULATIONS USING *NEURON*

R. Mutihac

A. Cicuttin

and

R.C. Mutihac

. . 33 / 10

IC/2001/120

United Nations Educational Scientific and Cultural Organization
and
International Atomic Energy Agency

THE ABDUS SALAM INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS

CLASSICAL ALGORITHMS FOR AUTOMATED PARAMETER-SEARCH METHODS IN COMPARTMENTAL NEURAL MODELS – A CRITICAL SURVEY BASED ON SIMULATIONS USING *NEURON*

R. Mutihac*
Department of Electricity and Biophysics, University of Bucharest, 76900 Bucharest, Romania
and
The Abdus Salam International Centre for Theoretical Physic, Trieste, Italy,

A. Cicuttin
The Abdus Salam International Centre for Theoretical Physics-INFN Microprocessor Laboratory, Trieste, Italy

and

R.C. Mutihac
Department of Electricity and Biophysics, University of Bucharest, 76900 Bucharest, Romania
and
The Abdus Salam International Centre for Theoretical Physic, Trieste, Italy.

MIRAMARE – TRIESTE

September 2001

* *Regular Associate of the Abdus Salam ICTP.*

1. INTRODUCTION

Neuroscience encompasses approaches ranging from molecular and cellular studies to human psychophysics and psychology. Theoretical analysis and computational neuroscience are important in characterizing the activity of various nervous systems and determining their functional specificity. The massive application of classical optimization (minimization) algorithms to many different problems in neural computation, mainly to data fitting, which involves minimization in multidimensions, has given rise to successes and failures that have not been always clearly understood and interpreted. In a broad sense, mathematical modeling and computer simulation are committed to provide deeper insight and generalize problems that are difficult, costly, or even impossible to investigate experimentally.

Neural models are presently being used to model neural structures of various complexity, such as single synapses and synaptic transmission, excitable membranes, neurons and neuromodulation [Hines, 1993; Destexhe, 1994; Hines and Carnevale, 1997] up to sophisticated networks of neurons [Bower and Beeman, 1995a; Koch and Segev 1997; Destexhe *et al.,* 1997; Vanier and Bower, 1999]. Morphologically- and physiologically-detailed, multi-compartmental models have been developed for a number of neuron types [Wilson and Bower, 1989; Bhalla and Bower, 1993; Bush and Sejnowski, 1993; Schutter and Bower, 1994; Shen *et al.*, 1999]. However, increasing the biological accuracy of a model claims generally more parameters, which make simulation computationally demanding and decrease the ability to interpret and generalize the results. Besides, efficient simulation of large neural networks (NNs) enforces simplification over single neuron models, yet retaining a maximum of their biophysiological relevant attributes [Mutihac, 2000].

The traditional approaches for evaluating the parameters in single neuronal models are:

(i) Estimate the passive electrotonic structure with current pulses [Rall *et al.*, 1992];

(ii) Estimate the kinetics, conductance, and reversal potential of active currents [Hodgkin and Huxley, 1952].

However, both methods are questionable for a number of reasons [Tabak *et al.*, 2000]. Yet some parameters are either not entirely characterized experimentally or are difficult to cast as constraints to the model. Therefore it is a common practice of running simulations containing loosely constrained parameters and iteratively set their best values (based on some prior knowledge) in such a way that the model will replicate faithfully the experimental data. The main pitfalls of manually parameterizing a model are:

(i) The more sophisticated the models are, the more parameters they require, which increase the computational demand and the processing time;

(ii) There is intrinsic nonlinear interdependency among the types of ionic conductances [Koch and Segev, 1997; Rinzel, 1985] that cannot be captured by varying parameters iteratively;

(iii) Manual parameter fitting fails to assess quantitatively the degree of similarity between the model predictions and actual physiological data, that is, to determine the optimal parameter values [Vanier and Bower, 1999].

2. MULTIDIMENSIONAL OPTIMIZATION

Data fitting may be difficult because of some inescapable reasons:

(i) The inherent complexity of the given problem;

(ii) Lack of adequately collected data;

(iii) The way the problem is framed in the context of optimization;

(iv) The numerical algorithm used for optimization (usually minimization);

(v) The particular form of the matching (error) function that measures the discrepancy between the model and data.

Basically, the task is to minimize the number of meaningful single neuron model parameters with direct impact on reducing the number of compartments and ionic channel types, yet preserving the realistic behavior of more complex models in terms of firing rate response to synaptic or electrical stimuli [Davison *et al.*, 2000]. Most real-world problems are too complex to solve efficiently on *trial and error*-based or random searching techniques. Mathematically challenging, criteria to assess the efficiency of various optimization methods and particular forms of relevant error functions were critically considered and quantitatively compared.

In practical problems, it is generally desirable to find a global minimum rather than local minima. However, for functions of moderate and large number of variables, the best that we can do is to use a good local minimizer and try several different combinations of starting positions, steplengths etc., in hope that the best local minimum found is a global one. Finding a global extremum is generally a very difficult problem. Two standard heuristics are widely used in this respect [Press *et al.*, 1992]:

(i) Find local extrema starting from widely varying initial values of the independent variables (i.e., chosen quasi-randomly) and then pick the most extreme of these (if they are not all the same);

(ii) Perturb a local extremum by taking a finite amplitude step away from it, and then see if the routine returns a better point, or "always" to the same one.

Two different approaches are theoretically recommended for global extremum search, though not entirely founded yet:

(iii) *Simulated annealing* (SA) methods, which have demonstrated important successes on a variety of global extremization problems, by means of occasionally uphill excursions allowed according to Boltzmann's law [Metropolis *et al.*, 1953; Kirkpatrick *et al.*, 1983; Kirkpatrick, 1984];

(iv) *Evolutionary computation* best represented by *genetic algorithms* (GAs), which are iterative procedures that operate on a population of individuals and apply the biological principle of natural evolution to artificial systems [Mitchell, 1996].

The optimization techniques tested in optimizing the matching functions for various compartmental neural models covered virtually all standard approaches that have been traditionally used in multidimensional nonliniar optimization problems [Polak, 1971; Gill *et al.*, 1981]:

(i) Methods without derivative information (PRAXIS, SIMPLEX, and Powell's algorithm);

(ii) Conjugate gradient methods (Fletcher-Powell's and Polak-Ribière's algorithms);

(iii) Variable metric methods or quasi-Newton methods (Davidon-Fletcher-Powell and Broyden-Fletcher-Goldfarb-Shanno's algorithms).

All algorithms were developed in C using the codes from Press *et al.* [1992], except PRAXIS that was implemented following Brent's approach [1976]. The algorithms were initially run in MS Visual C++ version 6.0, and subsequently integrated onto the NEURON environment (hoc codes). Simulations were run on a Pentium II PC at 233 MHz under Windows 2000 Pro.

3. NEURAL MODELING AND SIMULATION

Information processing in neurons is accomplished by the spread and interaction of electrical and chemical signals. These signals are characterized by spatial nonuniformity and complex temporal dynamics that are intrinsic to the operation of biological neural networks. Models of neurons or neural nets that are closely linked to experimentation must preserve the essential spatiotemporal features of these signals [Hines and Carnevale, 2000].

In modeling neurons, two types of complexity are encountered:

(i) The intricate interplay of active conductances that makes neural dynamics rich and specific;

(ii) The elaborate morphology that allows neurons to receive and integrate inputs from several other neurons.

Model neurons range from greatly simplified objects to highly detailed descriptions involving thousands of differential equations. Choosing the most appropriate level of modeling for a given issue requires a careful assessment of the experimental information available and a clear understanding of the research goal. A frequent mistake is to assume that a more detailed model is necessarily superior. Since models act as bridges between levels of understanding, they ought to be detailed enough to make contact with the lower level, yet necessarily so simple as to yield clear results at the higher level. Oversimplified models may produce misleading results, but excessively detailed models can obscure meaningful results beneath inessential and unconstrained complexity.

The models used for ranking the performance of various algorithms under test were equipped initially with parameters having reasonable average values (i.e., from previous simulations), which were subsequently changed by the program while running the model.

Computer simulation allows us to compare the behavior of an idealized model (a computational model) with the observed behavior of a biological system (an experimental model). This relationship may be seen in terms of two phases of experimental inquiry [Hines and Carnevale, 1994]. The first phase starts when the investigator decides to study a general phenomenon in a particular experimental system, formulates hypotheses or conceptual models, and judges which components of the system seem most important. This phase continues with the design of experiments, selection of experimental subjects, and interpretation of data to characterize these components. Each of these activities imposes inevitable choices and approximations: which details are essential, and which can be omitted? Simulation, the second phase of this experimental inquiry, faces a question that is more straightforward: does recombining the components create a model that reproduces the behavior of the original system? However, answering this question forces another series of choices and approximations. Regardless of the degree of behavioral similarity that is achieved, every simulation is an analogy compounded from multiple levels of approximation, each of which must be carefully examined.

The first level of approximation lies in our speculation about which components of the conceptual model are most important and how faithfully they must be represented in a mathematical or computational model to capture the essential features of the phenomena displayed by the biological system. The second level of approximation lies in the relation between the abstract computational model and its particular instantiation on the computer. Making something work on a computer compels us to move back from the abstract to the concrete. This often means adding details we would rather ignore, or even worse, being forced to invent them.

The simulation environment program NEURON version 4.3.1, which is available in public domain (http://www.neuron.yale.edu), was used throghout for simulating all our models described bellow. A reference library that provides an accessible location for storing and efficiently retrieving compartmental neuron models, which is also available free (http://senselab.med.yale.edu/senselab/), was accessed to download different models developed elsewhere. This database is part of the *Human Brain Project* (http://www.nimh.nih.gov/neuro-informatics/index.cfm), whose aim is to

develop neuro-informatics tools in support of neuroscience research. Models can be coded in any language for any environment, though the database has been initially elaborated for use with NEURON [Hines, 1993] and GENESIS [Bower and Beeman, 1995b]. NEURON incorporates a programming language based on *hoc*, a floating point calculator with C-like syntax described by Kernighan and Pike [1984]. This interpreter has been extended by the addition of object-oriented syntax (not including polymorphism or inheritance) that can be used to implement abstract data types and data encapsulation. Other extensions include functions that are specific to the domain of neural simulations, and functions that implement a versatile graphical user interface.

4. RESULTS AND DISCUSSION

Computational neuroscience, like any scientific discipline, can only be mastered with practice. It is based on the belief that methods of mathematics, physics and computer science can elucidate nervous system functions. Unfortunately, mathematics may sometimes seem more of an obstacle than an aid to understanding if inappropriately employed.

Building compartmental neural models requires assigning values to free parameters to make models match experimental data, which is a most difficult and time-consuming task. Ideally, all model structural elements should be derived directly from experimental data. Unfortunately, experimental evidence is not always complete or difficult to get. Simulations containing a large number of loosely constrained parameters are run and parameter values are assigned on the basis of trial-and-error approach.

Comparing different minimization methods involves numerical results reported in various references that may have been obtained on different computers with different precisions (word length) and with different line minimization procedures (when linear search is involved repeatedly). The effect of machine precision (generally double floating point) is noticeable in the final stages of search, when round-off errors determine the limiting accuracy attainable (except for ill-conditioned problems). Thus it is justified to compare the number of function evaluations required to get the function value within a reasonable neighborhood of the minimum (say 10^{-12}), rather than the number required for convergence. Since apparently minor differences in the linear search method employed can produce important differences, it is the number of linear searches to be more relevant rather than the number of function evaluations. Our comparisons are nevertheless based on the number of function evaluations, since all our algorithms are using the same code for linear search (when needed). Gradient information is obtained by numerical approximations of partial derivatives using an original implementation of the Neville's algorithm in multidimensions. This way, the task of gradient evaluation is also measured in terms of function evaluations required by the numerical evaluations of partial derivatives and meaningful comparison can be performed between algorithms that require gradient information and algorithms that do not.

Initially, we run a set of testing functions like Rosenbrock, Beale, and Leon (Fig. 1) as suggested by Brent [1976] designed to rank the ability of various algorithms written in ANSI C (for maximum portability) to reach the real minimum (local or global) of a function with several variables when its extremum is situated on a quasi-flat region in the variable space.

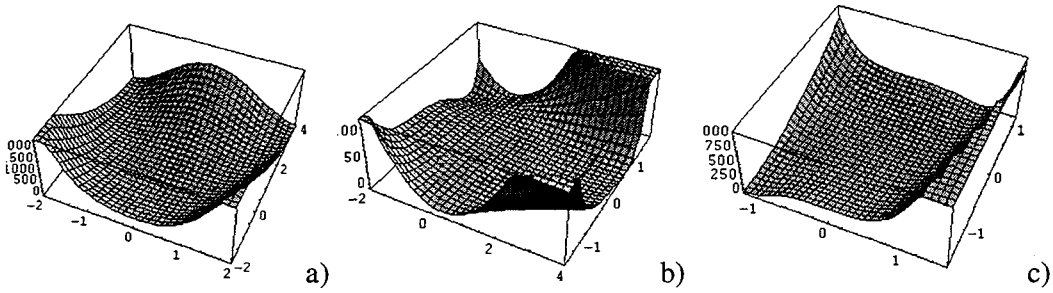


Figure 1. Testing functions for assessing the ability of the minimization algorithms to land in the minimum of a quasi-flat valley.

a) Rosenbrock: $f(x,y)=100\cdot\left(y-x^2\right)^2+(1-x)^2$;

b) Beale: $f(x,y)=(1.5-x(1-y))^2+\left(2.25-x\left(1-y^2\right)\right)^2+\left(2.625-x\left(1-y^3\right)\right)^2$;

c) Leon: $f(x,y)=100\cdot\left(y-x^3\right)^2+(1-x)^2$.

The basic model used in our evaluations was the instance of a neuron with a soma, three dendrites, and an unmyelinated axon. The diameter of the spherical soma is 50 μm. Each dendrite is 200 μm long and tapers uniformly along its length from 10 μm diameter at its site of origin on the soma, to 3 μm at its distal end. The unmyelinated cylindrical axon is 1000 μm long and has a diameter of 1 μm. An electrode (not shown) is inserted into the soma for intracellular injection of a stimulating current (Fig. 2).

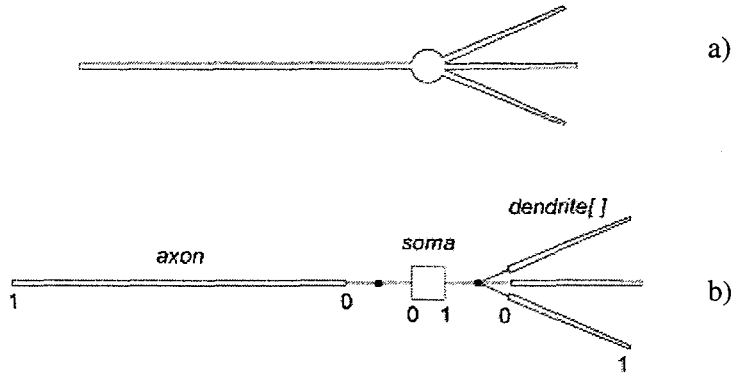


Figure 2. a) A neuron with a soma, three dendrites, and an unmyelinated axon (not to scale); b) Topology of a NEURON model that represents this cell.

NEURON is best suited (in terms of efficiency) for problems ranging from parts of single cells to small numbers of cells in which cable properties play a crucial role [Hines 1984]. It is best suited, in terms of conceptual control, for stylized morphologies represented as connected cable sections and where membrane channel parameters are conveniently represented as piece-wise linear functions of position within each section. Two special classes of problems for which it is well suited are those in which it is important to calculate ionic concentrations and those where one needs to compute the extracellular potential just next to the nerve membrane. It is well suited for investigating new kinds of membrane channels since they are described using a high level model description language that allows the expression of models in terms of kinetic schemes or sets of simultaneous equations.

NEURON is a program designed around the notion of continuous cable "sections" which are connected together to form any kind of branched-tree structure [Hines, 1989]. The value of this notion lies in the ability to specify the physical properties of a neuron without regard for the purely numerical issue of how many compartments will be used to represent each of the cable sections. This means that one can easily trade-off between accuracy and computational speed so that it is convenient for verification of the numerical correctness of the simulation. Figure 3 shows how NEURON represents a cable section in terms of a discrete set of nodes connected by resistors.

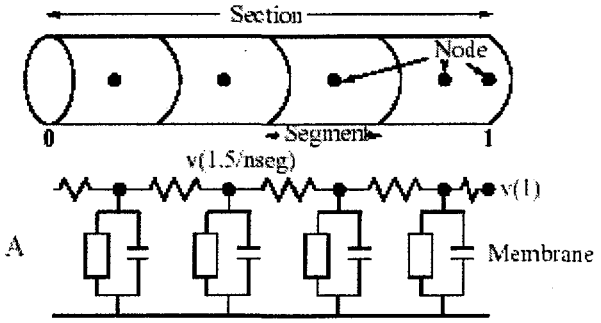


Figure 3. Section representation as electrical equivalent circuit by NEURON.

In Figure 4 there are presented the voltage and ionic currents for an action potential using the Frankenhaeuser-Huxley model [1964] and simulated with NEURON. It is clear that the accurate simulation of fast changing ionic currents requires smaller time steps than that needed for the membrane potential. Experimental data and the simulation results are in fairly good agreement, regardless the integration method used.

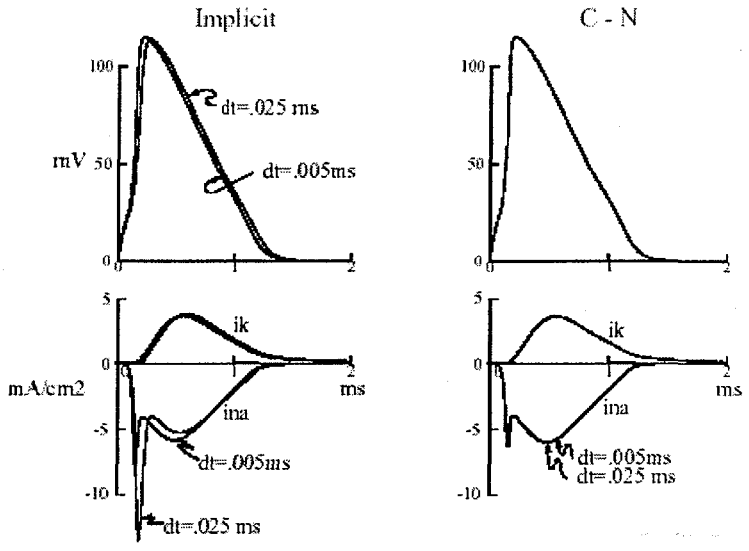


Figure 4. Typical Frankenhaeuser-Huxley action potential pulses. Comparison of fully implicit (Backward Euler) and Crank-Nicholson [1947] integration methods (with two time steps) to compute the mebrane voltage.

5. CONCLUSIONS

Parameter-search methods are problem-sensitive. All methods depend on some meta-parameters of their own, which must be determined experimentally in advance. A better choice of the these intrinsic parameters for a certain parameter-search method may improve its performance. Moreover, there are various implementations of the same method, which may also affect its performance.

The choice of the matching (error) function has a great impact on the search process in terms of finding the optimal parameter set and minimizing the computational cost. An initial assessment of the matching function ability to distinguish between good and bad models is recommended, before launching exhaustive computations. However, different runs of a parameter search method may result in the same optimal parameter set or in different parameter sets (the model is insufficiently constrained to accurately characterize the real system).

Robustness of the parameter set is expressed by the extent to which small perturbations in the parameter values are not affecting the best solution. A parameter set that is not robust is unlikely to be physiologically relevant. Robustness can also be defined as the stability of the optimal parameter set to small variations of the inputs.

When trying to estimate things like the minimum, or the least-squares optimal parameters of a nonlinear system, the existence of multiple local minima can cause problems with the determination of the global optimum. Techniques such as Newton's method, the *Simplex* method and Least-squares Linear Taylor Differential correction technique can be useful provided that one is lucky enough to start sufficiently close to the global minimum. All these methods suffer from the inability to distinguish a local minimum from a global one because they follow the local gradients towards the minimum, even if some methods (*Praxis*, *Simplex*, ...) are resetting the search direction when it is likely to get stuck in presumably a local minimum. Deterministic methods based on gradient-descent techniques are adequate if the parameter space is low-dimensional, relatively smooth, and has a few local minima. Stochastic methods are likely to be more effective for more ragged parameter spaces with many local minima (e.g., parameterizing single-neuron compartmental models). Only fast algorithms and/or a decent (low) number of model parameters are candidates for automated parameter search because of practical reasons. Eventually, the size of the parameter space may be reduced and/or parallel supercomputers may be used.

Data overfitting may negatively affect the generalization ability of the model [Mutihac *et al.*, 1999]. Bayesian methods include Occam's factor, which set the preference for simpler models. Proliferation of (neural) models raises the question of rigorous criteria for comparing the overall performance of various models designed to match the same type of data. Bayesian methods provide the best framework to assess the neural models quantitatively [MacKay, 1992].

Paradoxically, parameter-search methods may sometimes be more useful when they fail by discarding unrealistic mechanisms used in the model design, rather than fitting experimental data to an alleged model.

ACKNOWLEDGMENTS

The research has been partly carried out at the Abdus Salam ICTP, within the framework of the Associate Scheme. The authors are grateful to Dr. Michael Hines and Prof. Gordon Shepherd from the Yale School of Medicine, Section of Neurobiology, for their exquisite tutorials, criticism, and stimulating discussions on the NEURON simulation software. Also, the continuous support and encouragement of Prof. Julian Chela-Flores, the Abdus Salam ICTP, is highly appreciated.

REFERENCES

Bhalla, U.S. and Bower, J.M. (1993) Exploring parameter space in detailed single cell models: simulations of the mitral and granule cells of the olfactory bulb. *J. Neurophysiol.* 69:1948–1965.

Bower, J.M. and Beeman, D (1995a) "The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System." Santa Clara, CA: Telos.

Bower, J.M. and Beeman, D. (1995b) "The Book of GENESIS," New York: Springer Verlag.

Brent, R. (1976) A new algorithm for minimizing a function of several variables without calculating derivatives. In: "Algorithms for Minimization without Derivatives," Englewood Cliffs, NJ: Prentice Hall, pp. 200-248.

Bush, P.C. and Sejnowski, T.J. (1993) Reduced compartmental models of neocortical pyramidal cells. *J. Neurosci. Meth.* 46:159–166.

Crank, J. and Nicholson, P.N. (1947) A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Proc. Cambridge Phil. Soc.* 43:50-67.

Davison, A.P., Feng, J., and Brown, D. (2000) A reduced compartmental model of the mitral cell for use in network models of the olfactory bulb. *Brain Res. Bull.* 51:393–399.

Destexhe, A. (1994), Synthesis of models for excitable membranes, synaptic transmission and neuromodulation using a common kinetic formalism, *J. Comput. Neurosci.* 1:195-230.

Destexhe, A., Mainen, Z.F. and Sejnowski, T.J. (1997) Biophysical models of synaptic transmission. In: C. Koch, and I. Segev, eds. "Methods in Neuronal Modeling," 2nd edition. Cambridge, MA: MIT Press, (http://www.cnl.salk.edu/~alain/alain_articles.html).

Frankenhaeuser, B. and Huxley, A.F. (1964). The action potential in the myelinated nerve fibre of *Xenopus laevis* as computed on the basis of voltage clamp data. *J. Physiol.* (Lond.) 171:302-315.

Gill, P.E., Murray, W., and Wright, M.H. (1981) "Practical Optimization." New York: Academic Press.

Hines, M. (1984) Efficient computation of branched nerve equations. *Int. J. Biomed. Comput.* 15:69-76.

Hines, M. (1989) A program for simulation of nerve equations with branching geometries. *Int. J. Biomed. Comput.* 24:55-68.

Hines M. (1993) NEURON - a program for simulation of nerve equations. In: F.H. Eeckman, ed. "Neural Systems: Analysis and Modeling." Boston, MA: Kluwer Academic Publishers, pp. 127-136.

Hines M.L. and Carnevale, N.T. (1994) Computer modeling methods for neurons. In: M.A. Arbib, ed. "The Handbook of Brain Theory and Neural Networks." Cambridge, MA: MIT Press, pp. 226-230.

Hines M.L. and Carnevale, N.T. (1997) The NEURON simulation environment. *Neural Comput.* 9:1179-1209.

Hines M.L. and Carnevale, N.T. (2000) Expanding NEURON's repertoire of mechanisms with NMODL. *Neural Comput.* 12:839-851.

Hodgkin, A.L. and Huxley, A.F. (1952) A quantitative description of membrane current, and its application to conduction and excitation in nerve. *J. Physiol.* 117:500–544;

Kernighan, B.W. and Pike, R. (1984). The Unix Programming Environment. Englewood Cliffs, New Jersey: Prentice Hall.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1983) *Science* 220:671–680.

Kirkpatrick, S. (1984) *J. Stat. Phys.* 34:975–986.

Koch, C. and Segev, I. (1997) "Methods in Neuronal Modeling," 2nd editon, Cambridge, MA: MIT Press.

MacKay, D.J.C. (1992) Probable network and plausible predictions – A review of practical Bayesian methods for supervised neural networks, available at http://wol.ra.phy.cam.ac.uk/mackay/BayesianNets.html.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller A., and Teller, E. (1953) *J. Chem. Phys.* 21:1087–1092.

Mitchell, M. (1996) "An Introduction to Genetic Algorithms," Cambridge, MA: MIT Press.

Mutihac, R., Cicuttin, A., Cerdeira, A.E., and Colavita, A.A. (1999) Bayesian modeling of neural networks. In: M.J. Mira and J.V.Sanchez-Andres, eds. "Lecture Notes in Computer Science," *Foundations and Tools for Neural Modeling*. Heidelberg: Springer-Verlag, 1606:277-286.

Mutihac, R. (2000) "Fundamentals of Neural Modeling and Simulation." Bucharest: Publishing House of the University of Bucharest.

Polak, E. (1971) "Computational Methods in Optimization." New York: Academic Press.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1992) "Numerical Recipes in C: the Art of Scientific Computing," 2nd edition. Cambridge, MA: University Press, pp. 394-444.

Rall, W., Burke, R.E., Holmes, W.R., Jack, J.J.B., Redman, S.J., and Segev, I. (1992) Matching dendritic neuron models to experimental data. *Physiol. Revi.*, 72, 4:S159-S186.

Rinzel, J. (1985) Excitation dynamics: Insights from simplified membrane models. *Federation Proceedings*, 44, 15:2944-2946.

Schutter, E.D. and Bower, J.M. (1994) An active membrane model of the cerebellar Purkinje cell. I. Simulation of current clamps in slice. *J. Neurophysiol.* 71:375–400.

Shen, G.Y., Chen, W.R., Midtgaard, J., Shepherd, G.M., and Hines, M.L. (1999) Computational Analysis of Action Potential Initiation in Mitral Cell Soma and Dendrites Based on Dual Patch Recordings. *J. Neurophysiol.* 82:3006-3020.

Tabak, J., Murphy, C.R., and Moore, L.E. (2000) Parameter estimation methods for single neuron models, *J. Comput. Neurosci.*, 9:215-236.

Vanier, M.C. and Bower, J.M. (1999) A comparative survey of automated parameter-search methods for compartmental neural models, *J. Comput. Neurosci.* 7:149-171.

Wilson, M.A. and Bower, J.M. (1989) The simulation of large scale neural networks. In: C. Koch and I. Segev, eds. "Methods in Neuronal Modeling." Cambridge, MA: MIT Press, pp. 291-333.