Electrostimulation

SENN Startup Guide

Installation and Test of the

Spatially Extended Nonlinear Node (SENN) Model

This document is adapted from Chapter 11 of the book *Electrostimulation: Theory, Applications, and Computational Model*, by Reilly & Diamant, published by *Artech House Publishers* (2011), http://www.artechhouse.com/static/reslib/reilly/reilly.html.

The SENN model source code, input parameter file, executables for PC and Macintosh, this SENN model User's Startup Guide, and supplementary files are included in a free download from the above website. The download is provided without copyright restrictions.

Chapters 1 through 10 (Part I) of the book *Electrostimulation* cover theory and applications related to electrical excitation of nerve fibers and other biological tissue. SENN model theory, history and analysis results also appear in Part I. Chapters 11 through 14 (Part II) serve as a comprehensive *User's Guide* for the SENN model. This Startup Guide instructs the user in the model's installation and test. Detailed explanations and demonstrations of the model's parameters and modes are covered in Part II of the book.

This Startup Guide contains references to parts of the book, but stands alone for SENN model installation and test.

1
Getting Started

1.1 Introduction

Part II: User's Guide of the book *Electrostimulation* is a manual for operating the Spatially Extended Nonlinear Node (*SENN*) model, and is a companion to *Part I: Theory and Applications*. Where it might be helpful, cross-references between the parts are provided. We recommend that the user read at least the first three chapters of Part I as preparation for studying and applying Part II. An initial light reading of the four chapters of Part II would also be helpful.

Part II of the book, beginning with Chapter 11, combines features of a tutorial and a reference, somewhat like a favorite textbook. Chapter 11 tells how to install the model and see that it is fundamentally sound. Chapter 12 goes over the model's controls for basic familiarization. Chapter 13 describes several modes of operation, through the use of one or more examples per mode. Chapter 14 presents some details of "troubleshooting", to help in the analysis of unexpected model behavior. (Such analysis can produce insights, and even lead to new discoveries.) As with a tutorial, there is a progression of detail and complexity. As with a reference, the user with some experience can jump into a particular section of interest for information. Especially for reference-book mode, there are many cross-references to help the user find supplemental material on a topic, and there is some repetition to lessen the need to search.

1.1.1 Technical level. Part II of the book is written for the investigator familiar with computer operating systems and scientific analysis software. The user of the SENN model must be able to manipulate and organize files within the user's particular operating system. For pre- and post-processing of the model's inputs and outputs, facility with such analysis tools as Microsoft Excel is important. Graphical analysis plots of data on Excel spreadsheets appear throughout the book. The analyst must have the ability to manipulate and reduce arrays of spreadsheet data to create such plots, and to have access to the precise numbers behind them.

If the user wants to modify the model, experience with programming in a high-level language, such as Fortran, is beneficial. The novice programmer will need to study a language textbook or tutorial. Even an experienced programmer occasionally may need to refer to technical programming manuals. For those who would modify the model, Part II of the book gives instructions for re-compiling (i.e., reconstructing or "rebuilding") the SENN model for particular classes of PC and Macintosh platforms, and for particular language software packages. Users of other hardware and software suites must compile the model under the rules of their systems. This is not a daunting task, but instructions for all possible systems cannot be given here.

1.1.2 Terminology. When referring to Part II of the book, the terms "Part II", "User's Guide" or "Guide" may be used.

The terms “nerve fiber”, “fiber” and “neuron” are used interchangeably to describe a single nerve fiber.

The term “pulse” is used to indicate a single occurrence of any finite waveform, or as further described in context.

The SENN model is a compiled program that when “run” or “launched” performs tasks according to a set of high-level computer language instructions (such as Fortran) known as the “source code”. In the PC world, such a program is called an “executable” and the extension “.exe” is appended to its name. In the Mac world, the program is called an “application” and the extension “.app” is appended to its name. These extensions are important to the operating systems, but are often rendered invisible to the user. We will often refer to SENN.exe (or SENN.app) simply as “SENN” or “the SENN model” or “the model”. We will use the terms “executable” and “application” interchangeably.

Mathematical terms and their source code equivalents will be explained in context, and are listed in Appendix B.

1.1.3 Symbology. Standard mathematical symbols and equation formats are used in Part II. Also, Part II uses alternative operator notation, familiar to programmers and users of applications such as MS Excel, where the linear style of program code notation (i.e., no superscripts or subscripts) may be appropriate. These operators include the asterisk (*), the forward slash (/) and the double-asterisk (**). They indicate multiplication, division and exponentiation, respectively. A numerical subscript, or index, is placed inside parentheses. For example, z_n in program code might be written Z(N), meaning the contents of the *n*th row of an array of z-values. Program notation is used when equations are presented as they would appear in the model’s source code. (Elsewhere, one might encounter the carrot (^) to indicate exponentiation, and horizontal carrots indicating logical greater-than or less-than comparisons. Carrots are not used in our version of Fortran 77. Consult a Fortran 77 manual or code examples for the rules of notation.)

In book Chapter 12, *x* and *y* refer to the rectangular coordinates of an electrode, with respect to a neuron (Section 12.3.3). In book Chapters 13 and 14, we refer to arrays of data pairs generically as “(*x,y*)” arrays, whether *x* and *y* represent time and current (Chapter 13), or time and voltage (Chapter 14). Array names likewise will indicate the type of array, e.g., “*XYfile.txt*”. The meanings of array data pairs will be defined in context. In standard mathematical equations, *t* represents time.

For help in relating expressions in Part II to their counterparts in Part I, refer to book Appendix B for a listing of mathematical and (model) program variable equivalents.

1.1.4 Brief description of the SENN model. The SENN model, a computerized analogue of a myelinated nerve fiber, is coded in the *Fortran 77* source language. All examples in the book were run by an executable compiled under a software development kit available from the *Absoft® Corporation*[1], as will be discussed further below.

SENN mimics the reactions of the fiber to electrical stimuli having various temporal and spatial characteristics. SENN computes the minimum threshold current, or

[1] The Absoft® Corporation will hereinafter be referred to as Absoft, with the registration symbol implied.

current density, necessary to generate an *action potential* (AP, book Section 1.1). It can be adapted to approximate the reactions of unmyelinated fibers as well as cardiac and brain tissue, as discussed in book Chapters 6 and 9.

In practice, the user enters parameters to the model through the file *inparam.txt*. The file's contents specify physical characteristics of the fiber and its environment, temporal and spatial details of the stimulus, and the geometries affecting the nature of the electrical field along the trajectory of the fiber. Other parameters control modes of operation and details of computation.

When the SENN application is launched, it converges to a solution that is within a program-prescribed precision of about one percent, by iterations in a titration process that alternates between excited and unexcited conditions (book Sections 3.1.1 and 12.4.4). At the end of a run, the model presents a *standard output* window with the minimum threshold and an iteration-by-iteration summary of key variables. This window can be saved as a file. The model also generates the file *data.out***,** readable by text editors such as WordPad (PC) or TextEdit (Mac), which provides a more detailed history of temporal and spatial variables.

Output data text files are also generated for plotting the time progression of an AP at a node (*fort.30*), or the spatial progression of an AP across nodes (*fort.17*). Depending on the model's mode, other input data or parameter text files are used. These files will be described below.

1.2 Model installation and test

1.2.1 Overview. The purpose of this section is to provide SENN installation instructions, and a test case to verify model operation. The minimal explanations given here will be expanded in later chapters.

Free downloads of the SENN source code, executables and supporting files are available for PC and Macintosh platforms from *Artech House*, the publisher of the book (http://www.artechhouse.com/static/reslib/reilly/reilly.html). The downloads are provided without copyright restrictions. This abbreviated user's guide is included with the downloads, to help the potential user install the model and test its operation.

The PC and Mac executables were compiled using Absoft's *Pro Fortran* compilers (v10.2.2 for the PC, and v10.2.1 for the Mac.) Analysts may create their own executables from the Fortran 77 source code, using compilers of their choice. The computer-window figures and output table in this Startup Guide were copied from a PC screen, but would be similar for a Mac. Where there are significant operational differences between the two platforms, they will be described. Startup Guide Figure 1.1 shows the contents of a PC installation download. The *SENN PC ReadMe (date).rtf* file instructs the user to copy the folder *SENN_PC_(date)* to the hard drive, open that folder, and follow the instructions in the document *SENN StartupGuide (date).rtf*.[2]

[2] To ensure proper formatting, the Startup Guide has been changed from a .rtf to a .pdf document in the download.

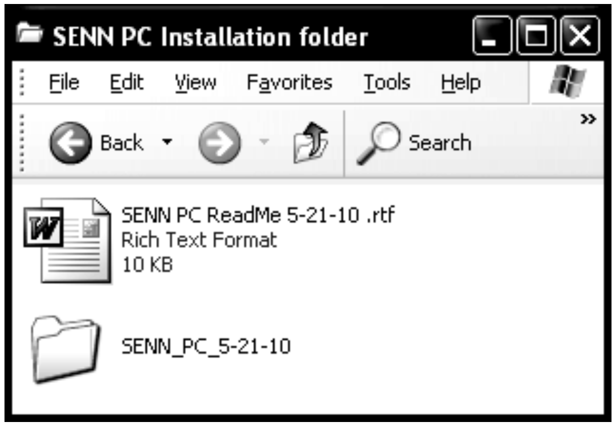

Figure 1.1. Contents of a SENN installation download.

Startup Figure 1.2 shows the contents of the *SENN_PC_(date)* folder shown in Startup Figure 1.1. This folder is the home directory containing all the files necessary for model operation, except for application development resources in the language package and operating system. The files in the home folder, including compiler files, comprise a *Project*, in this case a SENN project. To eliminate the need for user-provided path descriptions, store all data files accessed by the model for a particular project in this home folder, which may be renamed. For the same reason, any compiler file containing project-specific settings (e.g., Absoft file *SENN.atools* in the download) must also remain in this folder. More compiler details for the system used in the writing of the book are in Startup Section 1.2.3. Find full compiler details for this or any other system in the compiler manufacturer's documentation.

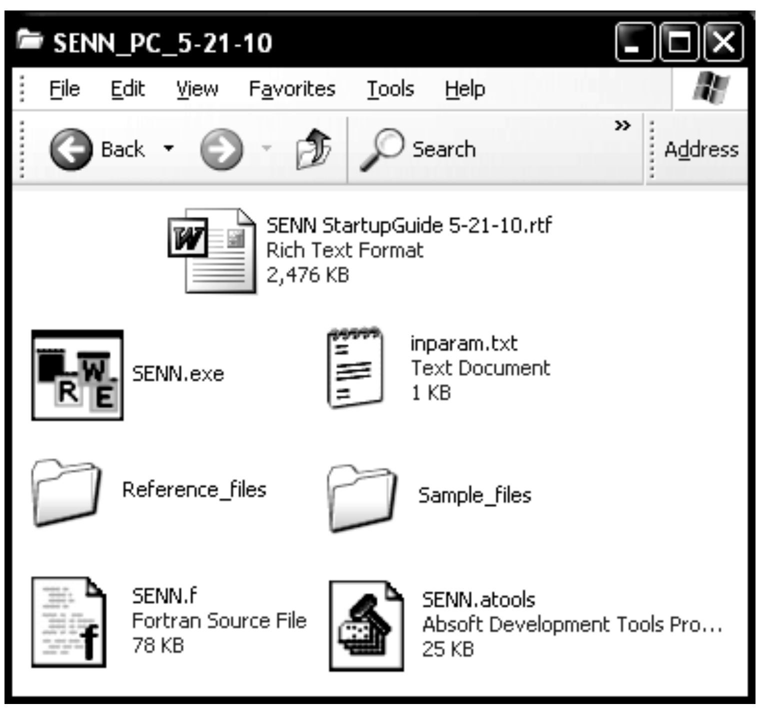

Figure 1.2. Contents of SENN_PC_(date) folder, the home directory.

1.2.2 Installation. Installation consists of copying (dragging) the SENN home folder named *SENN_PC_(date)* from the download to your hard drive, if that has not yet been done. (Mac users drag the *SENN_Mac_(date)* folder.) This folder may be re-named, but must contain all project files accessed by the application. Open the SENN folder that is now on the hard drive. Its contents, shown in Figure 11.2, consist of:

SENN StartupGuide (date).rtf: An abbreviated user's guide containing installation instructions and details of a test case for verifying operation. This document is meant for the potential user possessing only the download from publisher Artech House. The information in this Startup Guide is adapted from book Chapter 11.

SENN.exe (*SENN.app*, or just *SENN*, for Mac users): An executable program of the SENN model.

inparam.txt: A list of SENN input parameters to control its operational conditions.

SENN.f: The SENN source code. The code follows Fortran 77 rules. It was compiled by an Absoft F95 compiler in legacy mode to create the application SENN.exe.

SENN.atools: An Absoft application for setting compiler options. For those using an Absoft compiler, launching SENN.atools is the first step in setting compiler options and building a new executable (Startup Section 1.2.3). For other compilers, refer to their manuals.

Reference_files: A folder containing two saved output files from a run of a sample case. These are for comparison to the test case files generated when the executable is launched (in the next subsection). Within this folder:

- *StdOutRef.txt* provides a brief running summary of the model's iterations as it converges to a solution. It appears in a window during program execution, and can be saved as an option.
- *dataRef.out* provides a more detailed analysis of the progression in time and space of key variables. Also, the front part of this file repeats the input parameters and briefly explains their meanings.

Sample_files: A folder containing sample input data files for the external-input examples in book Chapter 13.

All input/output files are readable by text editors such as *WordPad* (PC) or *TextEdit* (Mac). The source code also can be read by general text editors, but won't have special features provided by the compiler editor. *Important*: *For proper viewing and printing alignment, use a monospaced font* such as Courier New, Courier PS, Lucida Console or Monaco.

1.2.3 Compile guidance. This section can be skipped if you have an executable and do not need to rebuild it. The instructions assume Absoft Pro Fortran compiler software is installed. (See Startup Section 1.2.1 for versions. You can determine your precise version by navigating to *C/Absoft10.2/absoft_release* for a PC, or *Applications/Absoft10.2/absoft_release* for a Mac.)

Note: Aggressive anti-virus software can block parts of compiler installation or operation. Troubleshooting should include disabling potentially interfering security functions. For safety, troubleshoot off line.

1a) For a PC, in folder *SENN_PC_(date)* double-click on *SENN.atools*.

1b) For a Mac, navigate to *Applications/Absoft10.2* and launch *AbsoftTools*. Select *Project/Open Project* and navigate to folder *SENN_Mac_(date)*. Double-click on *SENN.atools*.

2) If desired, modify the options according to compiler procedures. The Absoft Fortran options used by the authors are:

for the PC: -f77_compat -m32 -mrwe -march=host -f -s -01
for the Mac: -f77 _compat -m32 $(IMSL_F90FLAGS) -mrwe -march=host -f -s -01 -N9

3) Select *Build* or *Rebuild* from the menu options (*Rebuild* gives complete replacement.)

4) When you get the "*Build completed*" message at the bottom left of the window, close the Absoft tools window. You should now have an updated SENN application icon in this folder.

Note: Getting the "*Build completed*" message does not guarantee successful compilation. Scan the compiler's progress reports for error messages. Errors may be major or minor, and should be investigated for their effects on the operation of the model.

1.2.4 Test. Testing the model consists of double-clicking on *SENN.exe*. SENN will read *inparam.txt* and take several seconds to converge to a solution and stop. As mentioned in Startup Section 1.1.4, results will appear in a savable standard output window (sometimes referred to simply as the output window), and a text file named *data.out*. Also generated will be text files *fort.17* and *fort.30* which are useful in tracing AP progressions in time and space (The latter three diagnostic files will be further explained in Section 13.4.4 and Chapter 14 of the book. The standard output window will be referred to in every example to follow.)

Table 1.1 is the output window, generated iteration-by-iteration (in the titration process) during running of the test case, and referred to as *StdOutRef.rtf* in the Reference files subfolder. The input parameters (listed and described in book Chapter 12) are set for a 20 μm diameter fiber, and a monopolar cathodic point electrode on the surface of the medium surrounding the neuron.[3] The surface electrode is 1 cm radially distant from central node #26 of a 51-node fiber. The stimulus waveform is 10 cycles of a 5-kHz sine wave. The minimum threshold is 15.25 mA (UIO) in the 7th iteration (ITER), the last iteration during which the AP exceeded an excitation threshold of 80 mV (VMAX) in three consecutive nodes (NXGT), at time TMAX from onset of the stimulus. The iterations ended when there was only about 1 % difference between a suprathreshold level and a subthreshold level. Excitation (i.e., the formation of an AP) began at node #26. This window can be saved as a text file. (Mac users: After the optional Save, press the *return* key to complete the run, as noted at the bottom of the output window.)

More detailed explanations of SENN theory, inputs, modes and outputs are in the book. Our primary purpose here is to run an operational test to verify successful installation. The test case is meant to begin familiarity and validate the model's response

[3] Among its spatial field options, SENN calculates thresholds for current spreading from electrodes on the outer surface of, or within, the medium surrounding the neuron (*in situ*). Thresholds for surface electrodes are half the level of those *in situ*.

to input parameters. The numerical contents of the test run's output window should match those in Table 1.1. Minor details of formatting may vary from system to system, or as a result of future source code updates, which will be posted on the publisher's website (http://www.artechhouse.com/static/reslib/reilly/reilly.html).

Table 1.1. Standard output window generated by SENN.

```
SPATIALLY EXTENDED NON-LINEAR NODE (SENN) MODEL, 2010
IWAVE  2    FS  0    S  1
NNODES  51
pl  0.000000  pt  2.00000  dly  0.000000
AREA  1.099558E-06
  SINUSOID  FREQUENCY     5.000  PHASE   0.000  PERIOD     0.200
   I=     16.0000000
ITER#  1 VMAX 112.203  UIO    16.0000 #NODES  3  TMAX  0.39 NODE  26
   I=      8.0000000
ITER#  2 VMAX  13.603  UIO     8.0000 #NODES  0  TMAX  0.08 NODE  26
   I=     12.0000000
ITER#  3 VMAX  20.556  UIO    12.0000 #NODES  0  TMAX  0.08 NODE  26
   I=     14.0000000
ITER#  4 VMAX  24.326  UIO    14.0000 #NODES  0  TMAX  0.08 NODE  26
   I=     15.0000000
ITER#  5 VMAX  26.450  UIO    15.0000 #NODES  0  TMAX  0.08 NODE  26
   I=     15.5000000
ITER#  6 VMAX 113.914  UIO    15.5000 #NODES  3  TMAX  0.49 NODE  26
   I=     15.2500000
ITER#  7 VMAX 110.785  UIO    15.2500 #NODES  3  TMAX  0.79 NODE  26
   I=     15.1250000
ITER#  8 VMAX  26.737  UIO    15.1250 #NODES  0  TMAX  0.08 NODE  26

   ITER    VMAX     NXGT        UIO       TMAX   NODE
     1   112.2030     3       16.0000    0.3916   26
     2    13.6030     0        8.0000    0.0772   26
     3    20.5561     0       12.0000    0.0782   26
     4    24.3258     0       14.0000    0.0804   26
     5    26.4499     0       15.0000    0.0828   26
     6   113.9135     3       15.5000    0.4894   26
     7   110.7850     3       15.2500    0.7868   26
     8    26.7370     0       15.1250    0.0832   26

************* END OF RUN *************  1
HIT EOF ON INPUT
```

If the test run is not successful, see that there are no file-access (i.e., read/write) restrictions on input file *inparam.txt*: On a PC, right-click the input file icon for *Properties/Attributes* controls; on a Mac, click on the input file icon and select *File/Get Info* for *Permissions/Privilege* controls. Depending on the version of the operating system, the access controls may be worded differently; but the important thing is to give yourself full read/write privileges.

If the test run is still not successful, the model may have to be re-compiled for your particular computer system. The executables for Part II were created using PC and Macintosh compiler packages by the ABSOFT Software Development Corporation. File *SENN.f* is the Fortran source code. The code follows Fortran 77 rules, and was compiled by Fortran 95 compilers in Legacy mode.