The Logo Lowdown
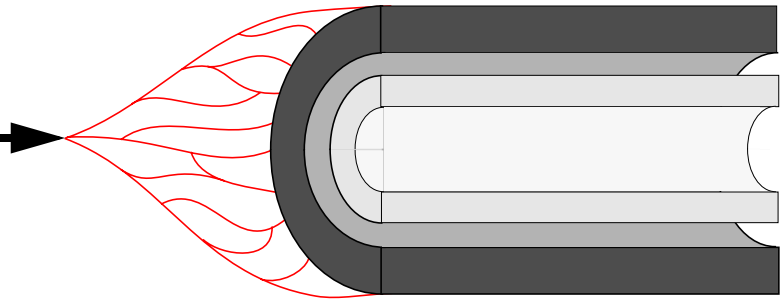
New editors come with new ideas. Eric Lawson has joined us from the University of Minnesota in Minneapolis. Here at NSR he is helping our staff communicate the results of the labors that make us a resource for scientific investigators around the world.

Assistant Editor Carson Lemmen, from the University of Washington, has devised the logo that tells you what kinds of models we've been concentrating on in recent times, and is designed to assure you that we are a Resource Facility for Circulatory Mass Transport and Exchange. The sole support for the core effort is the National Center for Research Resources (NCRR is the new name replacing DRR) of the National Institutes of Health (NIH grant RR1243); however, this newsletter and its content is the sole responsibility of NSR.

Models such as that hinted at in the logo are based on deterministic approaches to physiological exchanges and metabolic transformations where high resolution in time and space are desired. The applications are for experimental design and data analysis on cell cultures, isolated and intact organs, and whole body tracer distributions. Modeling packages for the interpretation of images, in particular images from Positron Emission Tomography (PET) and X-ray Computed Tomography (X-ray CT), are the focus of current developments.

In this issue, SIMCON, our modeling interface is featured. It is continuously changing in an upwardly compatible fashion, and the next year will see big changes. This issue contains update on SIMCON, and in depth articles on the actual models will be left for later issues.

Over the years, other programs for continuous system simulation have developed well; ACSL and CSSL-IV are good, commercially available examples. Our cousin resource facilities support others: SCoP from Simulation Resources, Inc. in Durham N.C. provides a wide variety of physiological and biophysical models; the Resource Facility for Kinetic Analysis here at the University of Washington supports SAAM, a widely used compartmental modeling system; Biomedical Simulation Resource at the University of Southern California, providing ADAPT II and LYSIS, reminds us that little of biological regulation is linear; and the Resource for Simulation of Stochastic Micropopulation Models at the University of Minnesota brings biology to the population level. As a family, simulation resources cover everything from basic biophysics to population dynamics.

Jim Bassingthwaighte

SIMCON Overview

SIMulation CONtrol (SIMCON) provides menus to help users run and control computer simulations. Users may plot as many as eight output variables on two separate line graphs, edit model parameters, and compare model output to reference data.

SIMCON operations take place at three or more levels, depending on the complexity of the model: Level 1, the SIMCON program itself, provides the user interface and does not change; levels 2 and 3 are externally supplied, variable subroutines that perform the simulations. Level 2 takes values given to SIMCON (level 1), calls a simulation model subroutine (level 3) which calculates output parameters, and then passes the parameters back to SIMCON.

SIMCON Interface	**Level 1**
Model Interface	**Level 2**
User Model	**Level 3**
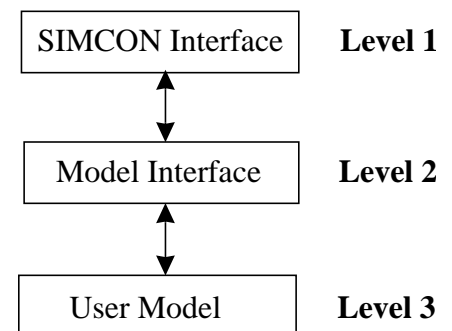
MMID4 Model Program

MMID4 (Multiple path, Multiple tracer, Indicator Dilution, 4 region model) simulates a blood-tissue exchange that reveals the behavior of three types of tracers: a tracer that remains in the

vasculature, a tracer that can leave the vasculature but remains extracellular, and a fully permeant tracer. MMID4 aids the study of the physiology of the exchange process.

The exchange process may be examined in terms of instantaneous outflow concentration, instantaneous extraction, or amount of tracer remaining in the exchange unit (i.e., the residue function). Features of MMID4 include methods of examining the effects of indicator delay and dispersion between the injection site and the target organ, and methods of examining the effects of flow heterogeneity on exchange within the organ.

Users' guides for SIMCON and MMID4 with more complete descriptions and tutorials are available from the Center for Bioengineering: (206) 685-2005, simcon@nsr.bioeng.washington.edu

New SIMCON Release

Here is a summary of changes made for SIMCON Version 2.3.0, which was installed in November 1990.

1. New Features/Enhancements:
   - The user can select the plot symbols used for simulation output.
   - A new model error handler is implemented.
2. Changes:
   - No routines from the NSR math library are duplicated in the SIMCON library.
   - When loops are on, the results file, simcon.out, contains the output from each solution and the parameter file, simcon.par, contains the initial values for the parameter(s) being incremented. When loops are on, hardcopy plots show the initial value and increment of the parameter(s) being incremented.
   - The format of the parameter plot code, p(146)-p(153), has been changed.
3. Bug Fixes:
   - Configuration parameters load correctly from the .simrc file.
   - P(1) can now be plotted.
   - Loading a parameter set from a file or database does not change the value of the parameter file controller, p(199).

This release of SIMCON is completely compatible with the previous release. Although the format of the plot codes, p(146)-p(153), has been modified, the plotting behavior is unchanged when existing values are used.

SIMCON Users Guide Update

All changes made to SIMCON through Version 2.3.0 are now documented in the Users Guide. The current version of the Users Guide is Release 4.0 (Dated 17JAN91).

Check the Documentation History on Page A (just after the cover page) to see if you have the current version. Contact Rick King at 685-2007 or by e-mail at simcon@nsr.bioeng.washington.edu for current documentation. If your manual's Documentation History shows that you have Release 3.0, request the 4.0 update. If you have Release 2.2 or earlier, request an entirely new manual.
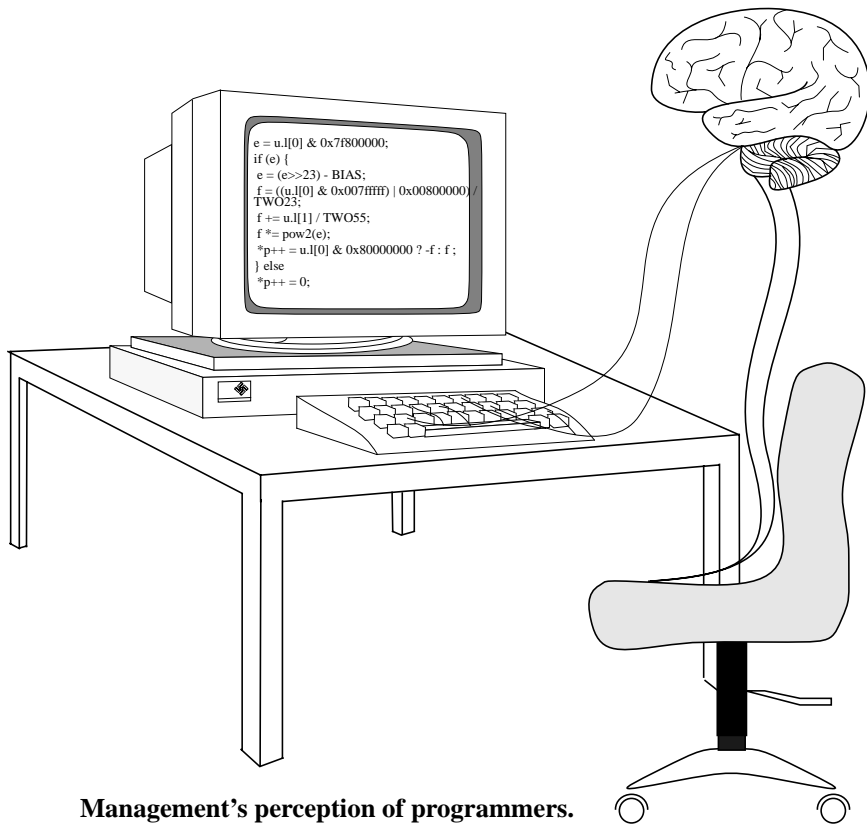
MMID4 Users Guide Update

A correction in Table B-1 of Appendix B of the MMID4 Users Guide has been made. To get this update, request the 18 January 1991 Users Guide update from Richard King at 685-2007 or by e-mail at simcon@nsr.bioeng.washington.edu.

Work In Progress

The SIMCON interface is being modified to split a SIMCON session into two tasks that use Inter-Process Communication (IPC) to pass necessary information between them. One of these tasks will handle all the functions of the interface and the other all the functions of the user model. Splitting the session into two tasks will have several advantages: first, executable versions of SIMCON model programs will not have to be remade when the interface is changed; second, executable versions of SIMCON models will be considerably smaller, because they will no longer contain code for the interface; third, the user will be able to switch to another model during a SIMCON session.

This change will not require any change in the code for SIMCON models: the communication of parameters through the P-array will remain unchanged.


```
e = u.l[0] & 0x7f800000;
if (e) {
e = (e>>23) - BIAS;
f = ((u.l[0] & 0x007fffff) | 0x00800000) /
TWO23;
f += u.l[1] / TWO55;
f *= pow2(e);
*p++ = u.l[0] & 0x80000000 ? -f : f ;
} else
*p++ = 0;
```

Management's perception of programmers.

Two flavors of IPC will be available. One will use BSD sockets to provide the communication. This use of BSD sockets for IPC allows the interface and model tasks to run on different computers. Thus, the user can run the interface on a local workstation and the model on a remote computer (such as a supercomputer). The other IPC flavor uses pipes, which requires both the interface and the model to run on the same machine. This flavor of IPC will be available for systems that do not support BSD socket IPC.

A version of this SIMCON suitable for testing (beta release) is now available.

A Brief Collaboration

In the infancy of NSR a representative of the local transit authority called our office. "We hear you're working on simulations of mass transport. Perhaps we could work together?"

SIMCON Question & Answer

Q. What should I do when I notice an error in a program or documentation?

A. If the error relates to SIMCON, SIMCON models, or their documentation, send e-mail to simcon@nsr.bioeng.washington.edu. If the problem is with one of our math subroutines, send the mail to locmat@nsr.bioeng.washington.edu. Be sure to send as complete a description as possible and copies of any input files that are needed to duplicate the problem. Also, if you have a request for enhancement of a product, send e-mail describing your idea.

Q. I'm a new user. What do I need to do to run a SIMCON model?

A. Section 3 of the SIMCON Users Guide describes how to set up and initialize a SIMCON environment. You must create a SIMCON database and add a UNIX environment variable that tells SIMCON where to find the database. Set up the environment variable first by adding the line: setenv SIMDAT $HOME/SIMDAT to the file .login in your home directory (See Section 3.2.2). Then enter the command: source .login to add the variable to your environment. You can now create the SIMCON database by entering the command:

mksimdat. This causes a database, named SIMDAT, with 10 parameter sets, to be created in your home directory (See Section 3.2.3).

It is also recommended that you create a SIMCON initialization file in your home directory so that SIMCON will start with the configuration that you want. See Section 3.2.1 for information about this file.

You are now ready to run a SIMCON model. Note, however, that when the database is created, it is filled with zeros. After you start a model, the first thing you should do is load a set of parameter values that can be used as a starting set of values for that model. To do so, use the database manipulation menu. Select option 1 (short or medium menu) or option G (long menu), and load the values from a file. Normally the name of the file to use is: /usr/local/lib/simcon/ {model}.par where {model} is replaced by the name of the model you are running.

Q. What is the difference between control parameters and model parameters?

A. When you run a SIMCON model, it uses a set of 500 parameters. These parameters come from the p-array (parameter array). Of these, parameters 100 through 200 have specific meanings that are known to the SIMCON interface and control how the interface works,

hence the name control parameters. They control such functions as what values are plotted, what scales are used for the graph, how the independent variable is changed, etc. See Section 5 of the SIMCON Users Guide and the SIMCON Control Parameters Reference Card for more detail.

The remaining p-array locations, 1-99 and 201-500, have no intrinsic meaning to SIMCON and are called model parameters. These parameters have specific meanings for a given model but these meanings may be entirely different in different models. Note that most models do not use all the available model parameters.

Q. Can you explain what is meant by the p-array, SIMCON databases, parameter sets, current set, working sets, and backup sets?

A. The p-array is the 500 location parameter array that is talked about above.

A SIMCON database is a file stored on disk that contains a fixed number of parameters sets (usually 10). Each of these parameter sets contains 500 values. When SIMCON starts, you are asked which of these sets should be used. When you enter a parameter set number, the 500 values from that set are read from the file and loaded into the 500 locations of the p-array. These are the values used by SIMCON and the model

when you do a simulation or optimization run.

The current set is the parameter set that is currently selected from the database. This is the set that will be affected by any changes that you make to the parameter values (or by changes that are made by the model or parameter optimizer). When you enter SIMCON, the current set is the one that you specify in response to the statement, "Enter number of the set to be used." During a simulation session, you may use the database manipulation menu to switch to another parameter set. The values from this parameter set are read from the disk and loaded into the p-array, and this set becomes the current set. Note that the number of the current set is displayed at the right of the status line above the menu.

There are actually two copies of each parameter set in the database. One is called the working set and the other the backup set. When you use SIMCON to alter parameter values, you are changing the values in the p-array (i.e. in the computer memory rather than what is stored on disk). These changes are written to the working set of the database on disk at specific times. (See the next question for more detail about when the changes are saved to disk.) Note that changes are always written to the working set. The only way that the backup set can be changed is by selecting option B of the database manipulation

menu. This causes all 500 values in the working set to be written into the backup for that set. Note that working set 1 is always backed up in backup set 1, etc. The values from the backup set can be recovered, i.e., written into the 500 locations of the working set, by selecting option R of the database manipulation menu.

Q. When are the changes that I make to parameter values saved to the disk?

A. The parameter values are written to the current working set of the database on disk at three specific times: (1) when a simulation run, a run of the model, completes successfully; (2) when you exit SIMCON; and (3) when you enter the database manipulation menu.

This means that if you make changes to the parameters, run the model, and the program crashes during the run, the changes will be lost. When you restart SIMCON and load the parameter set, you will get the values that were there before you started your last set of changes. The rationale for this behavior is that a "bad" set of parameters, i.e., one that causes the model to crash, should not be saved.

The problem is that you may have changed twenty parameters; if only one of the values was bad, you will have to reenter all the changes. To get around this, remember that the database file is updated when the database manipulation menu is entered. Thus, you can force

your changes to be saved by entering and immediately exiting this menu.

Rick King

The Next Issue

In the next NSR newsletter, look for Larry Weissman's report on changes in computer hardware over the past year and how that has changed the way we use our computers. Keith Kroll will write about an application using a single metabolite model in MMID4. Plus, look for more helpful information and updates on programs offered by NSR.

Editors Note

We would appreciate any comments or suggestions you might have about the content or form of this newsletter. Please call or send e-mail to Eric Lawson or Carson Lemmen at (206) 685-2010 (eric@nsr.bioeng.washington.edu or lemmen@nsr.bioeng.washington.edu). Contact us anytime with information, suggestions, or frivolous comments.