Technical Report No. 138

SIMCOMP VERSION 2.0 USER'S MANUAL

Jon Gustafson and George Innis

Natural Resource Ecology Laboratory

Colorado State University

Fort Collins, Colorado

GRASSLAND BIOME

U.S. International Biological Program

January 1972

TABLE OF CONTENTS

Abstract

SIMCOMP is a computer programming system which is designed to aid biologists with a limited knowledge of FORTRAN programming to design and execute compartmental-flow simulations. The system is designed to minimize the programming overhead required by any computer language while maintaining sufficient flexibility to solve most problems. A simulation is defined by specifying the flow rates between compartments and may be in either difference or differential equation form. Tabular and graphical output may be requested. The design and mathematical formulation of a simulation is described. The syntactical rules for writing SIMCOMP programs and the solution technique is explained.

Introduction

A basic requirement recognized by most integrated research programs in ecology, which utilize systems analysis and computer technology, is a methodology of applying the system sciences to biological problems. SIMCOMP is a programming system designed to ease this application. SIMCOMP accepts as input FORTRAN-like expressions which describe the mathematical computation of the flows between compartments in a system and generates a FORTRAN program which solves this system of flow equations and provides output of both printed and graphic information upon request. SIMCOMP compiles and organizes the flows defined by the user, supplying a standard solution scheme. By inspecting the flows and parameters defined by the user, SIMCOMP generates input-output routines which minimize the number of commands required by the user to direct, execute, and display his results.

By recognizing the basic similarities and requirements for the computer solution of any compartmental flow simulation, SIMCOMP is designed as a programming language tailored to the needs of ecologists involved in simulation as a basic tool of their research. The version of SIMCOMP described in this report represents an attempt to bring a technique of systems analysis into the increasing array of methodologies available to the ecologist. As such, ecologists can in turn contribute significantly to the expansion and refinement of the language so that future versions can be designed to further satisfy the ecologists' needs.

1. Compartmental Flow Models.

A broad variety of techniques have been developed to model and simulate many systems. Differential equations and difference equations have been most used in the development of ecological models. As greater reality and resultant complexity are introduced, the solutions have become more intractable and computers have been employed. Computer simulation in this context has come to mean the numerical solution of a simultaneous set of differential or difference equations.

Although many solution schemes are available, one of the most versatile approaches is to view the simulation as an initial value problem. The initial value problem for first order difference equations takes the following form. Let the amount of material or energy in the i^{th} compartment at time t be represented by $x_i(t)$. For a system of n compartments, the state of the system at any time t can be expressed as a vector

$$\underset{\sim}{x}(t) = \left\langle x_1(t),\ x_2(t),\ \ldots,\ x_n(t) \right\rangle .$$

Let a change in the state of the system over some time interval, say Δt from time t to time $t + \Delta t$, be represented by

$$\Delta\underset{\sim}{x}(t) = \left\langle \Delta x_1(t),\ \Delta x_2(t),\ \ldots,\ \Delta x_n(t) \right\rangle .$$

In general, the $\Delta x_i(t)$ are functions which may depend upon

(a) the values of the state variables at time t, $\underset{\sim}{x}(t)$,

(b) the values of a set of informational variables, say $v_j(t)$ for $j = 1, \ldots, m$, which, in general, vary with time (these in general include driving variables),

(c) the values of a set of parameters or constants, say p_k, $k = 1, \ldots, s$, which do not vary with time,

(d) and time itself.

The change in the ith state variable $\Delta x_i(t)$ at time t over the time interval Δt may be functionally written as

$$\Delta x_i(t) = F_i[\underset{\sim}{x}(t), \underset{\sim}{v}(t), \underset{\sim}{p}, t] \cdot \Delta t,$$

where the function F_i is the change per unit time in the state variable x_i.

Given the initial values of the state variables at time $t = t_o$, that is $\underset{\sim}{x}(t_o)$, and the changes in the state variables $\Delta\underset{\sim}{x}(t)$, we can find the state of the system at any time $t_m = t_o + m\Delta t$ for $m = 0, 1, 2, \ldots, M$. The state of the system at any time t_M is recursively computed as

$$\underset{\sim}{x}(t_M) = \sum_{m=0}^{M-1} \left\{ F_i[\underset{\sim}{x}(t_m), \underset{\sim}{v}(t_m), \underset{\sim}{p}, t_m] \cdot \Delta t \right\} + \underset{\sim}{x}(t_o)$$

In order to simulate biological systems, we postulate the following three principles.

(a) A biological system can be viewed as a collection of smaller subsystems. (Indeed some systems might consist of a single subsystem.)

(b) A change of state in any subsystem must result from the flow of material or energy between compartments contained in that subsystem.

(c) The identity of the material or energy flowing in any subsystem must remain constant throughout the subsystem.

As a result of the second postulate, we have further required that the change of state of any particular compartment be expressed as the algebraic sum of the flows to or from that compartment. Let the *net* flow per unit time from compartment i to compartment j be represented by

$$f_{ij} = f_{ij}[\underset{\sim}{x}(t),\ \underset{\sim}{v}(t),\ \underset{\sim}{p},\ t].$$

Note that $f_{ij} = -f_{ji}$; that is, the net flow into compartment j is reflected by a corresponding loss from compartment i, and by necessity the identity of the material flowing must remain unique. Therefore, expanding upon our formulation of the solution of the initial value problem, we find that the rate of change of material in some compartment i, above expressed as $F_i[\underset{\sim}{x}(t),\ \underset{\sim}{v}(t),\ \underset{\sim}{p},\ t]$, is the sum of the net flows from each of the associated compartments. Formally this requires that

$F_i = \sum_{j \in S} f_{ij}$, where S is the set of compartments which are coupled to compartment i by flows.

A nine compartment system comprised of two subsystems is illustrated in Fig. 1. The compartments are represented by the boxes. Material or energy flows are represented by solid arrows between the compartments. A flow which is always in one direction is represented by a single-headed arrow, such as the flow from compartment 1 to compartment 3. Flows in which the net flow may be in either direction are represented by a double-headed arrow, such as the flow between compartment 2 and compartment 3. Note that there are no material flows between compartments in separate subsystems. Informational flows are represented by dotted arrows. The rate of flow between compartment 5 and compartment 7, for example, is controlled by the amount of material in compartment 3. These informational flows are represented by $\underset{\sim}{v}(t)$ in our mathematical formulation. We may further identify compartment 1 as a source, provided the flow from 1 to 3 does not depend upon the quantity of material in 1, and likewise identify compartment 8 as a sink.

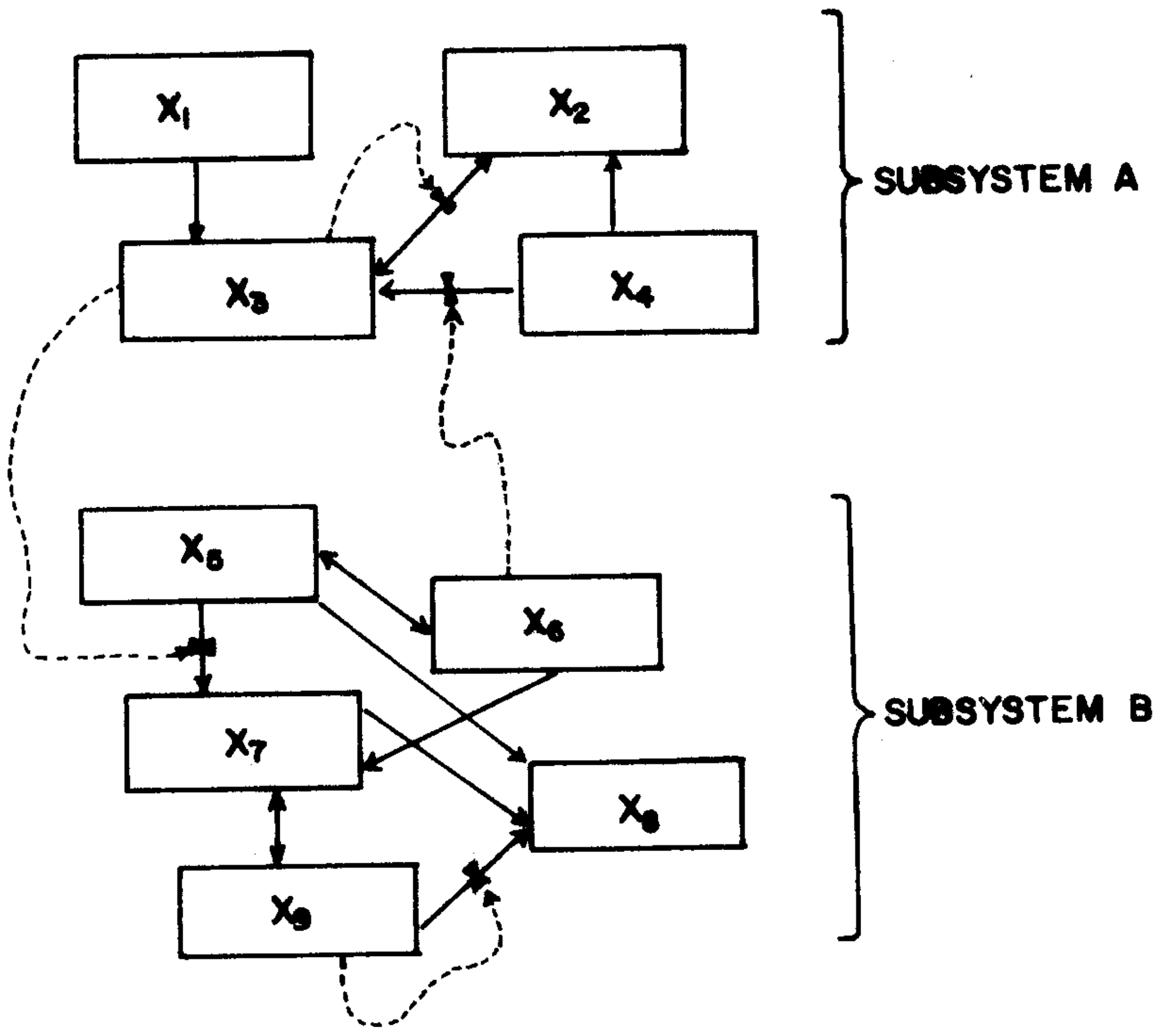

Fig. 1. A nine compartment system comprised of two subsystems illustrating actual flows (solid arrows) and informational flows (dotted arrows).

We have now identified the elements necessary to specify a computer simulation of a compartmental flow model. We require

(a) initial values for the state variables,

(b) mathematical expressions which calculate the net flows between compartments,

(c) mathematical formulas which calculate the informational flows,

(d) the identification and values for parameters and constants used in (b) and (c), and

(e) the starting and final times over which the simulation is to be run and the time step for the numerical solution of the initial value problem.

SIMCOMP is designed so that these five items are easily specified by the user. SIMCOMP organizes the flow expressions into difference equations and provides the solution. SIMCOMP further requires the user to specify what information is to be printed and plotted. The syntactical definitions for writing SIMCOMP programs are explained in the next section. The mathematical formulation of any compartmental flow simulation by first specifying the above five items, perhaps with the aid of a flow diagram, should precede the formulation of a SIMCOMP program.

2. SIMCOMP Programming.

SIMCOMP programming requires a basic knowledge of FORTRAN programming. This includes knowledge of integer and real variable modes, FORTRAN arithmetic expressions, and conditional and unconditional branching. The more sophisticated FORTRAN programming techniques such as input/output and subprograms are suggested but are not essential. The user is referred to any good instructional FORTRAN manual such as *Computer Programming - FORTRAN IV* by Decima M. Anderson.

SIMCOMP programs are divided into two sections: the source program section and the data section. SIMCOMP source program card format in general conforms to FORTRAN card format conventions. SIMCOMP compiler directives do not conform to FORTRAN card format conventions and must begin in or before column 5 and must be terminated by a period. Blanks are ignored. SIMCOMP data section statements are free format in columns 1 through 80, and blanks are ignored.

SIMCOMP reserves, as attributes of the system, certain variable names. Any of these variables may be used (or altered at the user's discretion) in the computation at any time. These variables and their meaning are shown in Table 1.

Table 1. Reserved variable names.

Variable	Meaning
X(i) where $1 \leq i \leq 99$.	The current amount of material in compartment i (variable x_i in chapter 1)
TIME	The current simulated time
TSTART	The starting time of the simulation
TEND	The ending time of the simulation
DT	The integration step-size (variable Δt in chapter 1)
DTPR	Time step between print-outs
DTFL	Time step between flow print-outs
DTPL	Time step between plotted values
F	The value of the currently computed flow (variable f_{ij} in chapter 1)

2.1 Source Program Section.

SIMCOMP source program statements are compiled into a FORTRAN program under the direction of SIMCOMP compiler directives. The physical location of the source section in a SIMCOMP deck is described in section 2.4.

2.1.1 Parameter declarations.

Variables and arrays are declared by the use of FORTRAN labeled common blocks. Variables and arrays, referred to as parameters, are specified by the statement form:

```
COMMON/cnm/v,v,...,v
```

The statement must begin in or after column 7 and may not extend past column 72. The statement may be continued on successive cards by placing a non-blank or non-zero character in column 6. Up to 19 continuations may be included. The common block name "cnm" may be from 1 to 5 alphanumeric characters in length. The parameters "v" may be from 1 to 7 alphanumeric characters, starting with a letter but not the letter "X". It is recommended that variable names be limited to 5 or fewer characters. A parameter may be subscripted with up to 3 subscripts if an array is being declared. Up to 50 common blocks, each containing up to 50 variables, may be declared. Certain variable names in addition to those starting with "X" are reserved by the system and may not be declared as parameters. These variables are those displayed in Table 1.

COMMON statements can each appear only once in a SIMCOMP program and may not appear in user-defined subprograms. COMMON statements which appear anywhere prior to a subprogram are automatically included in that subprogram.

Variables which fall in the following catagories need be declared in a labeled common statement.

(a) Values for variables which are to be set via a data assignment statement.

(b) Values for variables which are to be printed or plotted via a PRINT.* or PLOT. request.

(c) Variables which are computed in flows and are used in subprograms and vice-versa.

(d) Variables which are subscripted.

Example:

```
COMMON/RAPSV/A,T(3,3),PC35,PV3,R(6)
```

2.1.2 <u>Flow definitions</u>.

A flow between compartments in a subsystem is computationally defined by FORTRAN statement(s). A flow definition between compartments i and j is initiated by a SIMCOMP compiler directive of the form (i,j). and must begin in or before card column 5. A maximum of 99 compartments is allowed so that $1 \leq i \leq 99$ and $1 \leq j \leq 99$ for i and j integers. FORTRAN executable statements follow on the same or subsequent cards with the result of the computation being set equal to the reserved variables "F". A flow definition is terminated by the occurrence of another SIMCOMP compiler directive, a FORTRAN COMMON statement, a user-defined subprogram, or the end of the SIMCOMP source section.

* Note that the period is part of the command verb.

Flow definitions are specified by the following statement forms:

```
(i,j). F = Fortran expression
            or
(i,j).  Fortran executable statement
            .
            .
            .
        F = Fortran expression
            .
            .
            .
```

The FORTRAN statement "F = Fortran expression" may appear anywhere within the flow definition, but it must appear at least once and "F" must be equal to the net amount of material flowing between compartments i and j per unit time.

SIMCOMP will currently accept up to 500 flows. Upon encountering a flow definition, SIMCOMP generates the state variable compartments (reserved variables) "X(i)" and "X(j)". A state variable or compartment does not exist unless a flow to or from it is defined. SIMCOMP currently allows up to 99 compartments, therefore $1 \le i \le 99$ and $1 \le j \le 99$. A particular net flow may be defined only once.