**AUTOMOD 2.2**

Automatic Modelling of Ecosystem Flows

Robert E. Ulanowicz

University of Maryland

Chesapeake Biological Laboratory

Solomons, Maryland 20688-0038 USA

Internet: ulan@cbl.umd.edu

BACKGROUND:

The simulation modeling of entire ecosystems is fraught with many

difficulties. Prominent among them is the virtual failure of such models to provide reliable forecasts of system behavior into the future.

SCOR Working Group 59 (Platt et al., 1981) cited this problem with

marine ecosystem models and suggested that an alternative way to study

whole ecosystems might be to quantify the networks of material or energy

flows among the various compartments of an ecosystem and to submit this

data to various topological analyses. Accordingly, Ulanowicz (1983, 1986, 1988, Ulanowicz & Kay 1991) has developed a suite of four analyses, each operating at a different hierarchical level of the system, that are

available as a package, NETWRK4, written in FORTRAN 77 to operate on

MS-DOS personal computers. In its turn, NETWRK is but one element of an

assemblage of software packages collected by SCOR Working Group 73 (Kay et al, 1988) that treat ecosystem flow data presented in a standard format.

Network flow analysis is proving to be an exciting new area of

research into the structure of ecosystems. With little presupposition it

is now possible to say much in a diagnostic vein about an ecosystem using

network methodologies. But network analysis is intended to be only

diagnostic. As such, it will not fully satisfy those asking questions

about the dynamics of the system. Managers who have been exposed to the

results of network analysis are generally quite enthusiastic about the

further understanding these methods provide. However, there remains the

necessity for them to ask "what if" questions, e.g., "What if we halved the biomass of the ctenophores", or "What if the catch bluefish were doubled."

Clues toward answering managerial "what if" questions are provided by network analysis, but there remain many applied ecologist who will not be satisfied until simulations of the responses to their proposed

perturbations are provided. Of course, all the data necessary to create a simple model is at hand in the network (and there is no reason at this time to believe that more sophisticated simulations will give any better

predictions.) Hence, there is a need for an algorithm that will take the

data in standard SCOR format and create from it a generic model of

predator-prey dynamics.

AUTOMOD was written in 1987 to address this need. AUTOMOD 2.2 is an

expansion and update of the earlier versions. The intended user probably

already has the network data in standard format. (Readers who don't should not despair. The format is quite simple and a data entry routine, DATAIN, is provided with this package for those who don't even want to bother learning the SCOR format.) With the entry of a few terminal instructions to specify the nature of the perturbation, AUTOMOD will return a simulated response. Interpreting the results of the simulation can often prove to be a valuable hueristic exercise.

THEORY:

Donor Control:

AUTOMOD allows the user to choose one of three simple dynamic

schemes. The simplest (but least realistic) response is that of linear,

donor control. Here the flow out of any compartment is proportional to

the amount of medium in the compartment. The constant of proportionality is calculated by dividing the observed flow (as read from the input data) by the initial (or nominal) biomass of the donor compartment. The resulting set of first order linear differential equations is then integrated using a fourth-order Runge-Kutta scheme (Press et al., 1986).

The primary advantage of linear, donor controlled kinetics is their

inherent stability. If one begins with all positive flows and stocks, one is guaranteed that the flows and stocks generated by this model will remain non-negative. Hence, there is a use for these simple kinetics when one has data on a network that is not quite balanced (and anyone who has assembled a network from real data knows how difficult balancing each compartment can be!). To complete the balance, one simply runs the linear model for an appropriate time and obtains a balanced network.

Predator Control:

Donor control will often provide an acceptable model of "passive"

processes, such as respiration, sedimentation, etc. But most feeding

relationships are strongly dependent upon the densities of the predators.

of course, one could just as easily program linear predator control

kinetics, however, such a scheme is inherently UNSTABLE. In formulating

the mathematical expression for the flow it becomes necessary to include

some factor to indicate when the prey stock becomes dangerously low. The

bilinear, or Lotka-Volterra assumption does decrease the predation to zero as the prey goes extinct, but the linear dependency on prey stocks appears to be insufficient to keep such models from radical instabilities.

Wiegert (1973) employed thresholds to keep a predator controlled model from becoming unstable. Below a certain threshold of low prey availability predation was assumed to cease, thereby providing a "refuge" to keep the prey from becoming extinct. Above a second threshold of high prey density the feeding per unit of predator was assumed to reach a saturation value.

In between the two critical prey densities the specific predation rose in linear fashion.

It is possible to mimic the elements of Wiegert's modified predator

control function in a continuous way. In AUTOMOD the two-parameter

function describing the predation of species i by predator j is:

Fij = K \* Mj \* EXP (1.-Xi)

where Fij = the flow from prey i to predator j

Mj = the stock (biomass) of predator j

K = the ration of i in j's diet

(how much i each unit of j consumes under

the nominal circumstances)

Xi = Mio/Mi (see below)

Mi = the stock of prey i

Mio = the stock of prey i under the nominal

circumstances (i.e., at the beginning.)

The key thing to note is that at the nominal conditions Xi=1., and the

exponential factor disappears, revealing the predator control nature of the flow. As the prey becomes scarce the argument of the exponent becomes a large negative number, so that the flow to j becomes exponentially small.

For example, when the prey population drops to 0.1 of its nominal value,

the predator is receiving only 1.23E-04 of its nominal ration. At the

other end of the scale, when the prey is very abundant, Xi becomes

insignificant and the prey saturates at 2.72 times it nominal ration.

Empirical tests of the model show it to be stable to small perturbations

around the nominal conditions, whenever those specified conditions are

balanced around each compartment.

Lotka - Volterra Dynamics:

Experience with the predator control model over the five years

since AUTOMOD first appeared has shown that its performance leaves much to be desired. The inherent instability of predator control seems to

transcend even the predator refuge scheme for imparting stability. Not

that the system blows up, but rather it meanders in what resembles chaotic fashion. It is possible that real systems behave somewhat in this manner, but it is unlikely that they respond so dramatically to small parameter changes. Furthermore, with donor-control influence can propagate only in a forward direction; with predator-control it propagates only down the trophic chain. Hence, one desires a generic model wherein propagation is bi-directional and which exhibits at least the rudiments of nonlinear

behavior.

The Lotka-Volterra, or bilinear assumption allows influence to

propagate in both directions simultaneously. The general form of the

Lotka-Volterra (L-V) model is

n

dMi/dt = Mi(Bi+Sum PijMj)

j=1

where t represents time, Bi an intrinsic growth parameter and Pij are the

trophic interaction constants. Cohen et al. (1990), while attempting to

relate trophic statics to dynamics, implied that it is possible to

associate a sign-stable L-V model with each conceivable static

configuration. (A sign-stable L-V model is inherently stable due to the

configuration of the signs of its parameters and independent of their

magnitudes.) Thus, it becomes possible to choose as a third option for

AUTOMOD -- a sign-stable variation of L-V interactions to represent the

interaction dynamics.

For all living species (1,2,...,NL), the form of L-V dynamics will be

n

dMi/dt = Mi(Bi + Sum PijMj) = Ii - EiMi - RiMi

j=1

where Ii is the exogenous input to i and Ei and Ri are, respectively, i's export and respiration coefficients per unit biomass. By analogy with mass-balance one may choose

Fij = PijMiMj

knowing the nominal (and static) values for Fij and Mi, all parameters

except Bi and Pii are readily estimated. All contributions to the non-

living compartments (j = NL+1, NL+2,...,N) are assumed to be linear,

donor-controlled, i.e.,

Fij = PijMi

for all j>NL.

One sees by exclusion that the terms BiMi + PiiMiMi do not correspond to any observed mass flow. At steady-state one wishes them to disappear, i.e.,

BiMio + PiiMioMio = 0.

Away from steady-state these terms will be used as non-conservative

"clamping" flows. For example, if Bi>0 and Pii<0, then for small values

of Mi, the positive first term will dominate and keep that compartment

from going extinct. Values of Mi much larger that Mio will result in a

fictitious negative flow to retard further growth by that compartment.

To some extent the values of Bi an Pii are arbitrary, but it helps to relate them to the system at hand, if for no other reason than to keep

them properly scaled. It is easy to show that the quadratic form has a

maximum value of -BiBi/4Pii. For this model it is convenient to equate

this quotient to the gross productivity of the compartment as diminished

by the respiration, i.e.,

n

Sum Fjio = Iio - RiMio + -BiBi/4Pii.

j=1

(Another way of saying the same thing is that the net productivity at the nominal conditions is optimal.) Calling the entire left-hand side of the last equation Gi and solving it along with the above requirement that the fictional flows vanish at the nominal conditions yields

Bi = 4Gi/Mio

and

Pii = - 4Gi/MioMio

The clamping parameters just chosen should result in only moderate

excursions of the biomasses from the nominal conditions for all but the

most severe perturbations. The strategy then is to gradually relax the

clamping constraints in the hope of "relaxing" the system toward a new

equilibrium as the "fictitious forces" are gradually decreased toward zero and mass balance is retrieved. The rate of relaxation of the diagonal members is set to one-half the rate of the slowest compartmental turnover. This allows to system to equilibrate to it's new "freedom" in due fashion.

USING THE MODEL:

Two groups of data are required to run the program. Immediately upon typing the command to run AUTOMOD a query will appear on the screen asking the user to give the name of input file 5. This file should contain the data on intercompartmental flows and biomasses given in the standard SCOR format. For those not wishing to learn the SCOR format as described below a routine, DATAIN, has been provided with this package. It will prompt the user for the required data and write it onto an output file in standard SCOR format.

Sometime prior to data collection, assumptions had to have been made on how the ecosystem (or other system) was to be aggregated into compartments.

For each compartment it is necessary to know: (1) all the inputs from

outside the system, (2) all the various inputs flowing from other compartments of the system, (3) all the outputs which flow as inuts to

other compartments, (4) all exports of useful medium outside the system,

and (5) all rates of dissipation of medium. The system need not to be at

steady-state (when the sum of all the inputs to each compartment is

balanced by the sum of all the outputs from the same compartment), although a diagnostic warning will be printed whenever any compartment does not balance. Each of the flows can be represented by a positive scalar element of a matrix or a vector; the absence of a flow is represented by a zero.

Negative magnitudes of flow are forbidden and should be recast as positive quantities. The distinction between useful exports and dissipated respirations is, unfortunately, not made by some authors and collectors of data. This distinction is an important one and should be made wherever possible.

A sample of a simple energy flow network is the five compartment

ecosystem of Cone Spring (Williams and Crouthamel, unpublished ms) as

depicted in Figure 1. The "ground symbols" represent dissipative

respirations leaving the system.

(Figure 1 is identical to Figure 2 in Ulanowicz[1983])

The following is the input file corresponding to Figure 1.

CONE SPRING; WILLIAMS & CROUTHAMEL (PATTEN VOL 1); KC/M-2/YR 2

5 4

PLANTS

BACTERIA

DETRITUS FEEDERS

CARNIVORES

DETRITUS

1 285.

2 116.6

3 60.

4 17.

5 357.4

-1 0.

1 11184.

5 635.

-1 0.

1 300.

2 255.

5 860.

-1 0.

1 2003.

2 3275.

3 1814.

4 203.

5 3109.

-1 0.

1 5 8881.

2 3 75.

2 5 1600.

3 4 370.

3 5 200.

4 5 167.

5 2 5205.

5 3 2309.

-1 -1 0.

The first record is a header or title which serves to identify the

data and the resultant output. It is passed on as 78 characters of alpha-numeric data (19A4, A2) to the output file. It may contain any information the user desires. In this instance a descriptive title, a citation for the data source, and the units of the flows (kilograms of carbon per square meter per year) have been transcribed. The 2 in column 80 signifies that the data is accurate to at least 0.01 units. (This specification is optional, as described below.)

The first entry on the second record is the number of compartments in the network. Call this value N. Practically everything else is

dimensioned by N. The program as written allows a maximum of 40 components in a flow web. The second entry tells how many of the compartments represent living, feeding populations. It is necessary for creating a realistic model. The record format is ( I3, I3).

The network dimension record is followed by N records of alphanumeric titles, each title describing its respective compartment. Compartment descriptors may be up to 25 characters long and are simply transcribed onto the output file for easy reference in interpreting the printout of results. The ordering of the compartments should be such that the living populations appear first and the non-living compartments are grouped at the end.

The last compartment name precedes a series of values for the

biomasses of each component. Each record consists of an integer (I3 format) followed by the value of the stock. (The value is read by the machine in E14.7 format, however one may override this designation during manual input by simply typing the value anywhere in the 14-character field as a decimal number.) After the last biomass has been entered, the end of the series is denoted by a negative integer in the first three spaces.

The biomasses are followed by the inputs (only two here), and the

inputs by the exports of useful medium. The fourth series consists of

respirations. As with the biomasses, the end of each vector is signified by a negative value as the species designation. The default value of any flow is zero, so that zeros do not need to be entered directly. This affords a great savings in specifying the exchanges that follow the respirations.

Here two integers (I3,I3) are used to define the donor and recipient

compartments of each exchange. The values of each flow immediately follow

the recipient designation (default format E14.7, as for the vectors above.). As with the vectors, the end of the exchanges is denoted by a record with a negative integer in the first three columns. Each given exchange will appear in the exchange matrix so that the j-th element of row i represents the flow originating from compartment i and terminating in compartment j. Some authors treating input-output analysis transpose this convention.

The header record of the next run may immediately follow the end

record of the exchanges, or a hard end-of-file may be used to terminate

execution. All flow values must be non-negative, or else a diagnostic and

termination of the run will result.

It is often the case that the network under investigation is balanced, but the input data, by virtue of it's finite accuracy (e.g., 2 or 3 significant digits), will yield round-off error that the program would interpret as an imbalance. If one wishes to suppress the spurious warning diagnostic which would result from such round-off error, he should place an integer in column 80 of the header record to signify the expected accuracy of the data. This integer should be the negative logarithm (base 10) of the least significant digit in the data. For example, a 2 in column 80 of the header record will ignore any imbalances between 0.01 and -0.01. Suppression of such a spurious diagnostic is optional, as the subsequent calculations will not be greatly affected, as long as the network is nearly in balance.

Once the input data specifying the nominal conditions has been read, they will be echoed on to the output file. The user should respond to the query for a file name for unit 6 with directions for where to write the output from the model. (This may take the form of a filename [FILE.EXT] or the designations PRN [printer] or CON [screen].) If there are any major problems with the input data file, a diagnostic will appear on the screen and the program will shut down. In the event the input file is not balanced to the designated precision, a summary of the balances around each compartment will be printed. Also, in the event the user may wish to use AUTOMOD to bring the network into balance, a set of instructions are given at this point as to how to proceed with the reminder of the run in order to obtain a balanced network (with which to restart the program). Starting a modelling run with a network that is significantly out of balance could lead to the spurious extinction of some compartments.

The second group of input data consists of answers the user supplies to queries on the console screen. Before the first query, the shortest and longest compartmental throughput times are printed out to guide the user in answering the questions that follow.

The first prompt asks the user to specify the type of model to be

used. The user may reply with a 1 to execute the generic predator control formulation. Option 2, the linear donor control, is reserved mostly for balancing a network that had been assembled form literature data. Type 3 is the generalized Lotka - Volterra scheme.

The START TIME will ordinarily be zero, but in the event a run is

being continued, the user will be told what the RESTART TIME is. The END

TIME refers to when the user wishes to terminate the integration. The

integration interval (difference between end time and start time)

usually should exceed the longest throughput time by severalfold.

The values of up to seven compartmental biomasses will be printed on the output device at intervals specified by the user's response to "PRINT INTERVAL?" The print interval should be short enough for the user to follow changes in the fastest compartment of interest. The shortest

throughput time appearing on the console may be useful in specifying the

print interval. Small print intervals are recommended for the

Lotka-Volterra scheme, because relaxation of the clamping variables occurs in stepwise fashion at each printout interval.

AUTOMOD allows the user to take a "snapshot" of the entire system at the interval given in response to "SNAPSHOT INTERVAL?" When a snapshot is taken, all the flows, biomasses and header information and parameter values are written to unit 18 in standard SCOR format. Any one (or all) of these snapshots can be used as input to one of the SCOR network programs (including AUTOMOD). If one wishes a snapshot only at the end of the integration, a response of 0 (zero) should be given. The snapshot interval should be at least as long as the print interval. (The user may notice the parameter values are written to the right of the normal SCOR data in file 18. These values are necessary to restart a program at the exact point where it was broken off and will be ignored by the input routines of any other SCOR program than a restart of AUTOMOD2.)

The user is now ready to specify the type of perturbation he/she

desires. Parameter changes are specified one at a time. One enters the

parameter change loop by giving a positive response to the query,

"DO YOU WISH TO PERTURB A PARAMETER?"

There are five kinds of parameters that may

be changed. Changes in biomasses and exogenous inputs replace the values

of these variables in the model. Changes in exports, respirations and

internal exchanges are made as new COEFFICIENTS. The model seems quite

robust with respect to changes in the first two variables, however

perturbations in the coefficients often result in the extinction of one or more compartments from the predator control model.

Right after the user specifies the type of change to make, the program requests that the user specify which element to alter. The present value of that element is printed out and the user is then asked to specify a new value.

Once all the alterations have been made (signified by a negative

response to the parameter change query), the program is ready to run. If

there are more than seven compartments in the system, the user will be

asked to choose exactly seven compartments whose biomasses he wishes to

follow. For models with 7 or fewer compartments, the biomasses of all the compartments are written to the output file at each print interval.

Just before the first "snapshot" is taken, the user will be asked

where to write the SCOR-formatted snapshot file (Unit 18?). One usually

specifies a file different than the one on which the primary results are

written. File 18 will also contain all the information necessary to

restart the integration at the very same point where it last left off.

Below are presented a sample terminal session followed by the output it generates to Unit 6. The reader will notice that the output begins with an echo of the input data from Unit 5. Then a record of all the parameter changes appears, and finally the time series of simulated biomasses completes the file.

If at any point during the integration one of the biomasses becomes

inordinately small (less than 1.0E-09), that compartment is declared

"extinct." Notification of the extinction is written both to the console

and to the primary output file along with the time at which the extinction occurred. The extinct compartment is thereafter decoupled from the rest of the system and its biomass is arbitrarily set to a small (but non-zero value). This allows integration to continue without encountering underflow errors or model instabilities.

SAMPLE TERMINAL SESSION

A>automod

File name missing or blank - Please enter name

UNIT 5? test.dat

UNIT 6? results.out

SHORTEST THROUGHPUT TIME IS .2240E-01 IN 2

LONGEST THROUGHPUT TIME IS .3117E+00 IN 5

WHICH TYPE MODEL DO YOU WANT?

1. GENERIC PREDATOR CONTROL

2. LINEAR DONOR CONTROL

3. DAMPED LOTKA-VOLTERRA.

1

START TIME?

0.0

END TIME?

3.0

PRINT INTERVAL?

0.1

SNAPSHOT INTERVAL? (DEFAULT [0] SNAPS ONLY AT END.)

0

DO YOU WISH TO PERTURB A PARAMETER? (Y/N)

y

WHICH KIND?

1. BIOMASS

2. INPUT

3. EXPORT

4. RESPIRATION

5. INTERNAL EXCHANGE

1

WHICH COMPARTMENT?

4

OLD VALUE = .1700E+02

NEW VALUE?

8.

DO YOU WISH TO PERTURB A PARAMETER? (Y/N)

y

WHICH KIND?

1. BIOMASS

2. INPUT

3. EXPORT

4. RESPIRATION

5. INTERNAL EXCHANGE

2

WHICH COMPARTMENT?

5

OLD VALUE = .6350E+03

NEW VALUE?

700.

DO YOU WISH TO PERTURB A PARAMETER? (Y/N)

N

UNIT 18? snap.out

Stop - Program terminated.

A>

SAMPLE OUTPUT FROM FOREGOING TERMINAL SESSION

CONE SPRING; WILLIAMS & CROUTHAMEL (PATTEN VOL. 1); KC/M-2/YR

NUMBER OF COMPARTMENTS IS 5

NUMBER OF LIVING COMPARTMENTS IS 4

LIVING COMPARTMENTS

1 PLANTS

2 BACTERIA

3 DETRITUS FEEDERS

4 CARNIVORES

NON-LIVING COMPARTMENTS

5 DETRITUS

THE BIOMASSES

1 2 3 4 5

.285E+03 .117E+03 .600E+02 .170E+02 .358E+04

INPUT VECTOR

1 2 3 4 5

.112E+05 .000E+00 .000E+00 .000E+00 .635E+03

EXPORT VECTOR

1 2 3 4 5

.300E+03 .255E+03 .000E+00 .000E+00 .860E+03

RESPIRATION VECTOR

1 2 3 4 5

.200E+04 .328E+04 .181E+04 .203E+03 .311E+04

EXCHANGE MATRIX

1 2 3 4 5

1 .000E+00 .000E+00 .000E+00 .000E+00 .888E+04

2 .000E+00 .000E+00 .750E+02 .000E+00 .160E+04

3 .000E+00 .000E+00 .000E+00 .370E+03 .200E+03

4 .000E+00 .000E+00 .000E+00 .000E+00 .167E+03

5 .000E+00 .521E+04 .231E+04 .000E+00 .000E+00

THE FOLLOWING COEFFICIENTS HAVE BEEN CHANGED:

BIOMASS IN 4 = .8000E+01

INPUT TO 5 = .7000E+03

TIME 1 2 3 4 5

1.000 .285E+03 .124E+03 .606E+02 .113E+02 .359E+04

1.100 .285E+03 .122E+03 .720E+02 .143E+02 .355E+04

1.200 .285E+03 .114E+03 .679E+02 .209E+02 .353E+04

1.300 .285E+03 .109E+03 .541E+02 .213E+02 .357E+04

1.400 .285E+03 .112E+03 .523E+02 .156E+02 .362E+04

1.500 .285E+03 .119E+03 .612E+02 .135E+02 .363E+04

1.600 .285E+03 .123E+03 .692E+02 .165E+02 .358E+04

1.700 .285E+03 .120E+03 .637E+02 .214E+02 .356E+04

1.800 .285E+03 .117E+03 .536E+02 .199E+02 .357E+04

1.900 .285E+03 .120E+03 .538E+02 .151E+02 .360E+04

2.000 .285E+03 .123E+03 .616E+02 .136E+02 .359E+04

2.100 .285E+03 .122E+03 .673E+02 .163E+02 .356E+04

2.200 .285E+03 .117E+03 .627E+02 .199E+02 .355E+04

2.300 .285E+03 .114E+03 .553E+02 .189E+02 .357E+04

2.400 .285E+03 .116E+03 .558E+02 .155E+02 .360E+04

2.500 .285E+03 .120E+03 .620E+02 .148E+02 .360E+04

2.600 .285E+03 .120E+03 .657E+02 .172E+02 .357E+04

2.700 .285E+03 .118E+03 .611E+02 .198E+02 .356E+04

2.800 .285E+03 .117E+03 .556E+02 .183E+02 .358E+04

2.900 .285E+03 .119E+03 .567E+02 .156E+02 .360E+04

3.000 .285E+03 .121E+03 .619E+02 .151E+02 .359E+04

3.100 .285E+03 .120E+03 .643E+02 .172E+02 .357E+04

Program Listings:

AUTOMOD is written in FORTRAN 77, but with only minor changes should run on any decent FORTRAN IV compiler. The source code can be obtained by contacting the author. It includes abundant comments to help the user interpret what is happening at any point in the program.

Literature Cited

Cohen, J.E., T. Luczak, C.M. Newman and Z.-M. Zhou. 1990. Stochastic

structure and nonlinear dynamics of food webs: qualitative stability

in a Lotka-Volterra cascade model. Proc. Roy. Soc. Lond. B 240: 607-

627.

Kay, J.J., L.A. Graham and R.E. Ulanowicz. 1990. A detailled guide to

network analysis. Pp. 15-61 in F. Wulff, J.G. Field and K.H. Mann

(eds.) Network Analysis in Marine Ecology: Methods and Applications.

Springer-Verlag, Berlin.

Platt, T., K.H. Mann and R.E. Ulanowicz. 1981. Mathematical Models in

Biological Oceanography. UNESCO Monographs on Oceanographic

Methodology 7. UNESCO Press, Paris. 157 p.

Press, W.H., B.P. Flannery, S.A. Teukolsky and W.T. Vetterling. 1986.

Numerical Recipies: The Art of Scientific Computing. Cambridge

University Press, Cambridge MASS. 818p.

Ulanowicz, R.E. 1983. Identifying the structure of cycling in ecosystems. Mathematical Bioscience 65:219-237.

Ulanowicz, R.E. 1986. Growth and Development: Ecosystems Phenomenology. Springer-Verlag, New York, New York. 203p.

Ulanowicz, R.E. and J.J. Kay 1991. A package for the analysis of

ecosystem flow networks. Environmental Software 6(3): 131-142.

Ulanowicz, R.E. 1988. Ecosystem trophic foundations: Lindeman exonerata. In: B.C. Patten and S.E. Jorgensen (eds.) Progress in Systems Ecology. Elsevier, Amsterdam (in press).

Wiegert, R.G. 1973. A general ecological model and its use in simulating algal-fly energetics in a thermal spring. pp. 85-102 in P.W. Geir, L.R. Clark, D.J. Anderson and H.A. Nix (eds.) Insects: Studies in

Population Management. Ecol. Soc. Australia, Canberra.

Wulff, F., J.G. Field and K.H. Mann. 1989. Network Analysis in Marine

Ecology: Methods and Applications. Springer-Verlag, Berlin. 284p.