

A Package for the Analysis of Ecosystem Flow Networks

Robert E. Ulanowicz

Chesapeake Biological Laboratory, University of Maryland, Solomons, Maryland 20688-0038, USA

James J. Kay

Environment and Resource Studies, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

ABSTRACT

In a single software package NETWRK affords an ecosystem researcher or manager with several new avenues for extracting heretofore unavailable information about the flows of material or energy in the ecosystem. Clues about how an ecosystem may be reacting to a perturbation are often manifested first as changes in the relationships among the constituent taxa. Systematic analyses that quantify these configurational changes are only now beginning to appear as ecological software.

NETWRK consolidates four analytical techniques: (1) The study of indirect trophic effects allows the user to infer how a given taxonomic group affects or is affected by other compartments with which it may not be directly connected. (2) The myriad of trophic interactions is simplified into a linear chain of trophic transfers (*sensu* Lindeman), revealing how efficiently the system is processing the medium in question. (3) The full structure of pathways of recycle is elaborated, possibly depicting domains of control within the system. (4) The overall trophic status of the system is assessed by several indices deriving from thermodynamics and information theory. A before-after comparison of these system indices allows the user to render quantitative judgements about the extent to which the community in question has been impacted.

This nine year old program has been tested extensively and is available for use in both MS-DOS and Macintosh environments in either FORTRAN or Pascal. Copies are available from the authors.

KEYWORDS: flows, networks, foodwebs, ecosystems, trophic

INTRODUCTION

One of the popular ways of representing the activities taking place in an ecosystem is to depict them as networks of flows of material and/or energy. More often than not these network schematics resemble "spaghetti" or "bird's nest" diagrams that are difficult to interpret without recourse to systematic mathematical tools. (See Figure 1 for example) The software package NETWRK is a collection of such tools which allows the user to uncover the wealth of information contained in quantified networks of ecosystem flow networks. These tools consist of measures of various aspects of the flow configurations and several analyses of the pattern of connection among the compartments, both direct and indirect.

Knowing the direct interactions that occur in an ecological community provides information on the indirect influences that members, lacking immediate contact, have on one another. It is possible to trace along concatenations of direct exchanges to estimate the

magnitudes of such "influences at a distance." The mathematical tools for elaborating indirect trophic effects is called input-output analysis and was introduced into ecology by Hannon [1] and refined by Patten [2] and many others. Using the input-output analysis subroutine within NETWRK, for example, one can tell just how much of each lower-level resource goes to support a specific top carnivore. Alternatively, one can delimit the influence of any primary input to the system.

The observed configuration of food transfers usually bears more resemblance to a convoluted web than it does to the food chains of early ecological theory. However, using mathematical techniques developed by Ulanowicz [3,4], it becomes possible to "untangle" the jumble of pathways and recombine them into a concatenation of straight line trophic transfers that is representative of the status of trophic interactions of the system as a whole. The "Lindeman trophic analysis" subroutine in NETWRK provides a canonical representation of trophic transfers and efficiencies that can be compared with like transformations of completely disparate ecosystem networks.

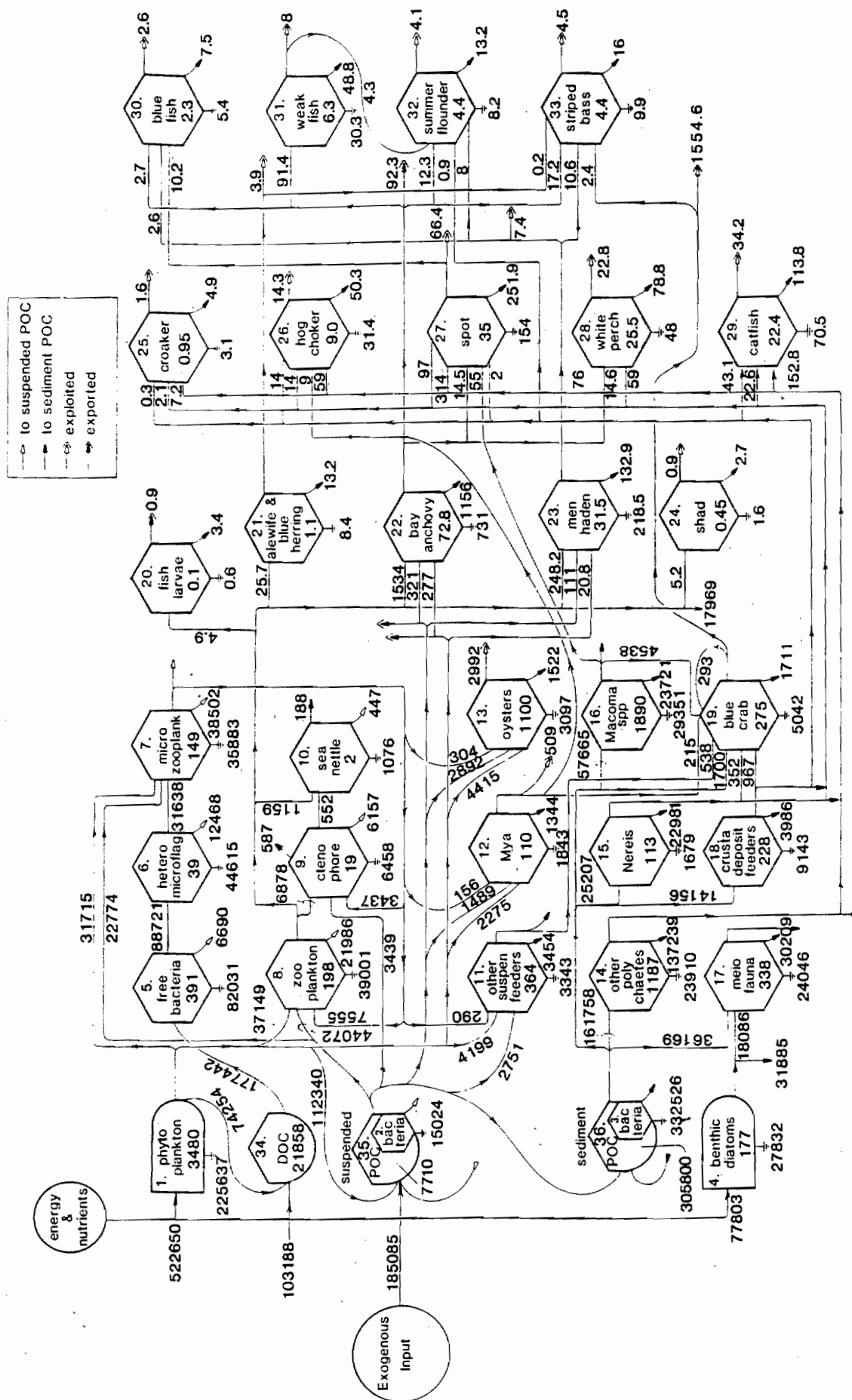


Figure 1: Carbon Exchanges in the Chesapeake Bay Ecosystem [25].

Flows in mg Carbon/m²/y; Stocks in mg Carbon/m²

One of the activities of an ecosystem with great diagnostic potential is the pattern of recycle of material (and, to a lesser extent, bound energy) within the system. NETWRK identifies, enumerates and conveniently reports as functional groupings, called "nexuses", all the simple pathways for recycle inherent in the given network. Furthermore, the cycles are all extracted from the starting configuration, and the network is separated into an acyclic network of dissipative flows and a purely cyclic pattern of recycled medium.

Finally, it is possible to quantify the overall characteristics of the given ecosystem as a set of mathematical measures on its exchange network. Information theory can be employed to quantify how well "organized" the trophic web is, how much functional redundancy it possesses, what potential for development it possesses, and how much its autonomy is encumbered by the necessary exchanges with the external world. These indices are often useful for documenting the effects of perturbations at the level of the whole system, or for identifying unequivocally when eutrophication has occurred. [5]

To summarize, four types of analyses are performed by NETWRK. Each analysis focuses on a different hierarchical level of system aggregation. The outputs are presented in ascending degree of aggregation. First, input-output structure matrices are calculated, allowing one to look in detail at the effects any particular flow or transformation might have on any other given species or flow. Next, the graph is mapped into a concatenated trophic chain (after Lindeman [6]). Then all the simple, directed biogeochemical cycles are identified and separated from their supporting dissipative flows. Finally, global variables describing the state of development of the network are presented.

HISTORY OF THE PROGRAM.

The original impetus for emphasizing the analysis of flow structure came from the field of economics, where success in elucidating indirect economic effects had been achieved by manipulations on matrices of economic flows [7,1]. Thereafter followed a number of other topological treatments of the underlying flow graph [8,9,10,11]. The intention in writing NETWRK has been to collect input-output analysis and various other analytical techniques into a single large program which allows an investigator to choose the analysis that best serves his needs. It is also hoped that by comparing the results of different methods (sometimes operating at different levels of aggregation) new insights will appear which would not have been possible by confining one's attention to a single approach.

The first version of NETWRK was written in 1982 in FORTRAN IV on a mainframe computer. A second version followed in 1985 and in 1986 NETWRK 3 was written. NETWRK 3 was the first version converted to run on MS-DOS micro computers.

In 1986 a Pascal version (Turbo Pascal) of the program was created. The Pascal version takes advantage of two features of that language, the ease and flexibility of I/O, and pointers and linked list data structures. The I/O capabilities of Pascal allowed the construction of a friendly user interface. Pointers and linked list structures are well

suited for programming the graph theoretic methods used in the analysis of cycles in ecosystems.

In 1986, Ulanowicz published a book [3] which provides the theoretical background for the analysis carried out by NETWRK. The book also provides detailed examples of the calculations done by NETWRK. In December 1986, a workshop was held to examine the application of flow analysis programs (including NETWRK) to marine ecosystems. The results of this workshop, and further analysis of ecosystems by the participants, have been published as a book [12].

In 1988 NETWRK 4 appeared. A Pascal version (Lightspeed) was written for the Macintosh in 1989. This version was presented at the annual meeting of ISEM [23]. NETWRK is now a mature and robust package which has been tested by many users in different environments. The program will continue to be upgraded to reflect developments in the theory of flow analysis of ecosystems.

DATA REQUIREMENTS

All of the routines in NETWRK require data on the entire network of exchanges of a particular medium (energy or some form of matter) in the ecosystem. Sometime prior to data collection assumptions had to have been made on how the ecosystem 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 inputs to other compartments, (4) all exports of useful medium outside the system, and (5) all rates of dissipation of medium (usually respiration losses).

A sample of a simple energy flow network is the five compartment ecosystem of Cone Spring (Williams and Crouthamel, unpublished manuscript) as depicted in Figure 2. The "ground symbols" represent dissipative respirations leaving the system.

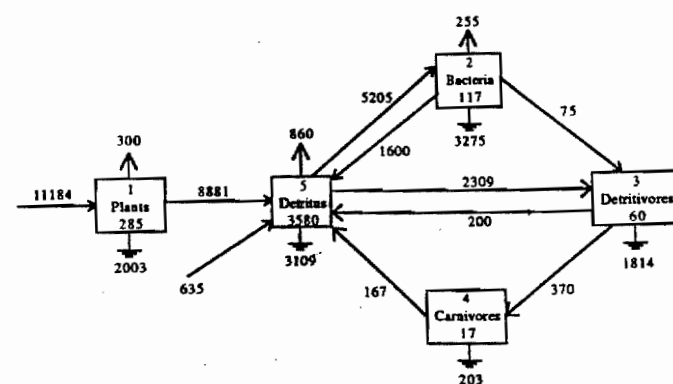


Figure 2: Energy Flows ($Kcal/m^2/yr$) and densities ($Kcal/m^2$) among the 5 compartments of the Cone Spring ecosystem

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. As can be seen from the results, this distinction is an important one and should be made wherever possible.

The data input format used in NETWRK 4 is the standard data format adopted by *Scientific Committee for Oceanic Research (UNESCO) Working Group 73*. The authors feel that a standard format for storing ecosystem flow data would contribute greatly to the sharing of data sets and programs by researchers. An example of this data format is shown in Table I, and consists of the data set associated with Figure 2. (All lines must not exceed 80 characters.)

Table I
The following is the input file corresponding to Figure 1.

```

CONE SPRING; KC/M-2/YR
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 line of the data set is a header or title which serves to identify the data and the resultant output. It is passed on as 78 characters 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 last two characters (i.e. 79 and 80) indicate the accuracy of the data. (This specification is optional, and is described at the end of this section.)

The first entry on the second line 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 37 compartments in a flow web, although this restriction can be relaxed by straightforward changes in the dimension statements. The second entry (called *NL*) tells how many of the compartments represent living, producing populations. It is necessary for the trophic analysis.

The network dimension line is followed by *N* lines of alphanumeric titles, each title describing its respective compartment. (e.g. line 2 of the titles corresponds to the second compartment: Bacteria.) Compartment descriptors are simply transcribed onto the output file for ease of reference in interpreting the printout of results. The ordering of the compartments must be such that the living compartments appear first and the non-living compartments are grouped at the end.

The last compartment name is followed by a series of values for the biomasses (standing stock), imports (input from outside the system), exports (output outside the system) and dissipation (non-usable respiration) vectors. (Note that the biomasses are not used in NETWRK, but are part of the standard SCOR format. The *i*th element of each vector is the value for *i*th compartment.) The default value for each element of the vectors is 0. Thus, only the non-zero elements need be specified as part of the data set. Since these vectors are usually sparse, this saves considerably on the storage required for each data set. Each non-zero element of the vector is stored as one line in the file, the compartment number appearing first and the value of the flow second. The end of each vector is signified by a line containing a negative compartment number.

Following these vectors is the matrix of flows exchanged between compartments. Each line corresponds to one entry of the exchange matrix, the first number defines the donor compartment number, the second the recipient and the third the value of the flow. This matrix is terminated in the same way as the vectors.

It should be noted that 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, when performing input-output analysis, use the transpose of this convention.)

An option to specify the accuracy of the data is provided. This is because even though the network under investigation is balanced, the input data, by virtue of its finite accuracy (e.g., 2 or 3 significant digits), may yield round-off error that the program would interpret as an imbalance. If one wishes to suppress the spurious diagnostic which would result from such round-off error,

an integer should be placed after column 78 of the header line 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 after column 80 of the header line will ignore any imbalances between 0.01 and -0.01. Suppression of the diagnostic is optional, as the subsequent calculations will not be affected by the warning diagnostic. (However, if the network is actually unbalanced, interpretation of some of the results may be ambiguous.)

THE ANALYSIS AND ITS RESULTS

The output resulting from the analysis of the Cone Spring example is listed in Table II. It is divided into five segments, each segment corresponding to a group of related analyses.

The initial segment begins with echoes of the input data; the header record, the total number of compartments, the number of living compartments, the names of the various compartments both living and non-living, the import vector, export vector, respiration vector and matrix of exchanges.

Next is the *THROUGHPUT* of a compartment. It is the total amount of medium flowing through each compartment in a steady-state. In input-output analysis throughput is a measure of the importance of that particular entity. The sum of these throughputs is called the total system throughput. It is a flow measure of the total system activity or size. In the case of a non-steady state system, the *IMBALANCE* between inputs and outputs appears next. (Throughput has no meaning in the unbalanced case.)

Structure Analysis

The second section of output presents an ecological variation of economic input-output analysis. Leontief [7] developed economic input-output analysis as a means for quantifying the amount of raw materials and industrial services required for producing a quantity of consumer goods. He introduced the backward case analysis, wherein outputs or demands are traced back to inputs. Augustinovic [13] later developed the forward case economic analysis that traces the fate of inputs through the system until they become outputs.

Hannon [1] was the first to apply Leontief's analysis to ecological systems. He was interested in determining the direct and indirect contributions of the system to a unit of output, and therefore modified Leontief's analysis to solve the equations differently. His success spawned a flurry of efforts among systems ecologists. Finn [8] later developed a forward case analysis for analyzing ecological networks. In addition he introduced several measures of ecosystem function, including cycling, that aid in the ecological interpretation of the results of input output analysis. Ulanowicz [3] (Chapter 3) has documented these analytical techniques in textbook fashion.

The *TOTAL CONTRIBUTION COEFFICIENTS* describe exactly what fraction of the total amount leaving compartment *i* (row designation) eventually enters compartment *j* (column designation) over all real pathways, direct and indirect [14]. For example, of all the gross production by compartment 1 (the plants), about 3% eventually enters compartment 4 (the carnivores). More

interestingly, 10.9% of the gross carnivore production winds up in its host, the detritivores (the 4-3 entry.) The rows of the contribution matrix usually sum to more than one, because output from a given compartment can visit several other compartments before exiting the system. Each diagonal entry indicates how much self-stimulation occurs in that particular compartment.

The *TOTAL DEPENDENCY COEFFICIENTS* portray the obverse to the perspective afforded by the previous matrix. Namely, the *i*-*j*th entry is the fraction of the total ingestion by *j* which passed through compartment *i* along its way to *j*. Hence, 17.2% of the carnivore diet (4) was mediated by the bacteria (2). The columns of this matrix are particularly useful in that they portray the extended diet of each species. Thus, although the carnivores (4) receive direct sustenance only from the detritivores (3); we see from column 4 in the dependency matrix that the carnivores are also wholly dependent upon the detritus and derive 93.3% of their ration ultimately from the plants, 17.2% from the bacteria (as noted above), and 1.72% from their own output.

Such indirect rations should provide valuable information to those managing a particular species. A Chesapeake Bay example would be the striped bass (*Morone saxatilis*) and the bluefish (*Pomatomus saltatrix*). Both appear as pelagic carnivores. Their extended diets, however, reveal significant niche separation. The striped bass is a recipient of energy and material that was passed up the pelagic grazing chain, whereas the medium reaching the bluefish had arrived mostly via the benthic community. This disparity in the extended diets provided a partial explanation of why the pesticide Kepone (a contaminant of the James River sediments) was appearing in high concentrations in the flesh of the bluefish, but was not prominent in the striped bass.

The next set of results is one of the most useful applications of input-output analysis. It is a decomposition of the graph according to each input [8,10]. For each of the non-zero inputs one assumes in turn that a unit of medium enters the network via that input and that no other inputs are present. The consequences of that single input are then displayed. Not only does this decomposition portray the isolated effects of the various inputs, but these sub-networks can be linearly recombined to recreate the effects of any combination of inputs, if the flow structure were kept the same. (Of course, the structure will probably change, but that does not render the exercise useless.) For example, when 11,184 units enter the plants and 635 enter the detritus, 75 units flow from the bacteria to the detritivores. If instead, 9,000 units entered the plants and 2,000 flowed as input to the detritus from outside the system, then $(9000 \times .00626 + 2000 \times .00788) = 72.1$ units would flow from compartment 3 to compartment 4 under the same flow structure.

Lindeman Trophic Aggregations

The third section of output is the result of analysing the given network according to the trophic concepts of Lindeman [6]. Of course, it is impossible to relegate many heterotrophs entirely to a single trophic level, but Ulanowicz and Kemp [11] indicated how input-output

techniques could be used to apportion the activities of omnivores among a series of integer trophic levels. This method has been expanded to include the effects of biogeochemical cycles by Ulanowicz [15], and it is that later formulation which is implemented by the software.

In order for the analytical results of this section to be meaningful, it is necessary to have specified the number of compartments that represent living populations, say NL ($<N$), and to have ordered the species so that the $(N-NL)$ abiotic compartments appeared last in the series. The trophic aggregation algorithm also requires that no cycles consist entirely of living compartments (i.e. there must be at least one abiotic element in each simple cycle). Pimm [16] states that such cycles are rare in ecosystem networks, and it is assumed that the magnitude of any such cycles are quantitatively insignificant.

The reader may feel uneasy about the requirements stated in the previous paragraph. It often happens that given networks do contain several small cycles among the heterotrophs. In that case the program will remove these cycles using an abbreviated form of the cycles algorithm described in the next section. The key feature to watch, when heterotrophic cycling is present, is that the Finn cycling index for heterotrophic cycling (see below) is sufficiently small -- below two percent or so.

If the number of living compartments (NL) is not given in the input, then a default value of N is assumed, i. e. , everything is alive. In this event, all the cycles will be removed, and the entire trophic analysis will be conducted on the acyclic residual network. Users are urged to make the distinction between living and abiotic compartments!

The first item to appear in this section is the **LINDEMAN TRANSFORMATION MATRIX**. It has dimension $NL \times NL$ and may contain rows of zeros towards the bottom of the matrix. Its columns represent the apportionment of the corresponding species among the integer trophic levels. Hence, the carnivores have been assigned approximately 97% to trophic level 3 and 3% to level 4. The columns of this matrix should always sum to one. Reading across a row gives the composition of a trophic level. Whence, trophic level 3 consists of 3% of the detritivore activity and 97% of the carnivore activity. All abiotic activity is condensed into the N th compartment and assigned a trophic level of one.

When the figures in each column of the transformation matrix are weighted by the value of the trophic level and the results are summed, one arrives at the **EFFECTIVE TROPHIC LEVEL FOR THAT SPECIES**. These trophic position values have been defined by Levine [9] as measures of the average trophic levels at which each compartment is receiving medium. For example, if a species or compartment is receiving 15 units of medium along a pathway of length 2 and 5 units along a pathway of length 3, then it is acting 75% as a herbivore (trophic level=2) and 25% as a carnivore (level=3). The effective trophic position becomes $(75 \times 2) + (25 \times 3) = 1.75$. This value can be compared with the corresponding trophic positions of other members of the community for purposes of ranking. More interestingly, any change in this value as the external conditions or community composition change could be used to help assess how well the members of the associated compartment are adapting to

the new conditions. (A lowered trophic status is usually indicative of stress on the population in question.). The average trophic levels in the Cone Spring network are rather unexciting, but the same analysis on more complex networks often yields interesting surprises.

The "**CANONICAL**" **EXPORTS AND RESPIRATIONS** are the amounts leaving the system from each integer trophic level. Again, the N th or last compartment represents the aggregated detrital pool. The elements of the **GRAZING CHAIN** represent the inputs to each integer trophic level from the preceding level. (The first value is the aggregate of the exogenous inputs. The second entry represents herbivorous grazing, of which none is present in the Cone Spring model.) The **RETURNS** from each level of the trophic chain to the detrital pool are listed in the next vector. **DETRITIVORY** refers to the flow from the detrital pool to the second trophic level. The ratio of this value to that for herbivorous grazing (second entry in the grazing chain) measures the relative importance of recycle to the community. Although recycle constitutes less than 10% of the total activity in Cone Spring, its dominance over herbivorous grazing makes it all-important to the existence of the higher trophic levels.

The exogenous **INPUTS TO THE DETRITAL POOL** total 635 units in Cone Spring, and there is no **INTERNAL CIRCULATION** within the pool. (Any positive value here would be represented by a "self loop" on the detrital compartment.) The configuration of the data presented thus far can be depicted as in Figure 3.

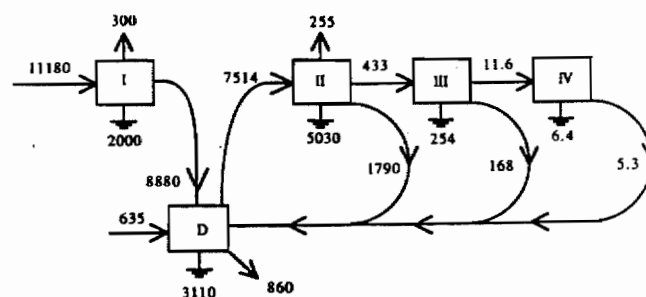


Figure 3: The trophic aggregation of the Cone Spring network with the autotrophs and detritus separated. The trophic levels are designated by roman numerals, and D represents the detritus pool.

That the grazing chain for Cone Spring was not a monotonically decreasing series can be ascribed to the effects of recycle. However, when the detrital pool is merged with the autotrophs, the resultant "**LINDEMAN SPINE**" necessarily will form a decreasing sequence of flows. One can speak of **TROPHIC EFFICIENCY** as the ratio of the input to a trophic level to the amount that level passes on to the next. Thus, of the 7,510 units entering level 2, only 433 are passed on to level 3, a 5.8% trophic efficiency. The trophic representation of Cone Spring after the merger of the autotrophs with the detrital pool is shown in Figure 4

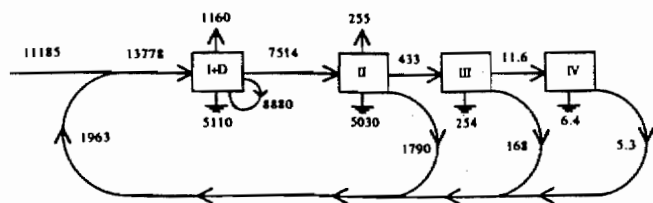


Figure 4: The trophic aggregation of the Cone Springs network with the autotrophs and detritus pools merged. (I+D)

Biogeochemical Cycle Analysis

Most ecosystem graphs contain biogeochemical cycles of material or energy, and it is the structure of these cycles that is fully described by the fourth section of the output. Positive feedback is a critical phenomenon in determining overall system structure. Positive feedback occurs in ecosystems when flows cycle among compartments. Compartments engaged in positive cycling will have an increase in output relative to those compartments which do not. This could result in a strong competitive advantage over non cycling compartments during growth phases. As well, positive feedback can be a stabilizing factor in systems with a time lag. Cycles in ecosystems are an important factor contributing to their autonomous behaviour [3,17]. Furthermore, it has been argued [18] that the organization of flows in ecosystems reflects a tendency to degrade the exergy content of the flowing energy while conserving materials flowing within the system. Exergy degradation is related to straight-through flows, while material conservation is related to the cycles in the ecosystem. These are reasons for studying the cycling in ecosystem flows and in particular for separating the flow network into two components, the cycles and the straight-through flows.

The process of separating cycles from straight-through flows has two distinct aspects. The first involves identifying which components are connected together to form a cycle. This is a straightforward topological problem whose solution can be found in most elementary graph theory texts. The second aspect involves partitioning the flows among compartments into those associated with cycles and those associated with straight-through flows. This is a problem which has not been dealt with extensively and which still requires further investigation.

What follows is a brief description of the process of identifying cycles. More detailed descriptions can be found in [3,17].

Topological Cycles and their Enumeration

The identification of the structure of the cycle, that is, what is topologically connected to what to make up the cycle, begins with a depth-first search (See Ch. 6 of [19]). The purpose of the search is to identify the cycle arcs. Cycle arcs are connections between compartments whose direction is from a higher compartment number to a lower compartmental number. In the Cone Spring example the arc from detritus to bacteria (from 5 to 3) is a cycle arc. If one searches for cycles starting only from those compartments which have cycle arcs into them, one is

assured of finding all the cycles. This significantly cuts down on the time for searching out and enumerating all the cycles.

The next step is a backtracking search. The idea of a backtracking search is to start with one compartment and see if it is connected to the next higher numbered compartment (i.e., is 3 connected to 4?). If it is connected, then we see if this next compartment (4) is connected to the lowest numbered compartment not already visited whose number is as least as great as the starting compartment number (i.e., is 4 connected to 5?). The process continues until the starting compartment is reached or until we find a compartment which is not connected to any compartment whose number is higher than the starting compartment number and which has not been visited. In the former case a list of the visited compartments constitutes a simple cycle. In the latter case we back up one compartment in our list and see if it is connected to the second next lowest numbered compartment not already visited (i.e., is 3 connected to 5 given that 3 is connected to 4 and 4 is not connected to any compartment whose number is greater than 3 nor to 3) and so on.

This is an algorithm which is best described by pictures rather than words. The idea is to treat each compartment as a dot. The dots are numbered. Compartments i and j are connected if the ij th element of the exchange matrix is non-zero. One tries to find a path, along the arcs connecting the dots, which starts and finishes with the same dot and which contains no other dot more than once. The trick is that from any dot you may only try to follow a path either to the lowest numbered dot (whose number is greater than the original dot) which you have not already tried, or to the original dot.

Table II:

The output from NETWRK for the data in Table I.

CONE SPRING; 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

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

COMPARTMENTAL THROUGHPUTS

	1	2	3	4	5
1	.112E+05	.521E+04	.238E+04	.370E+03	.115E+05

*** STRUCTURE ANALYSES ***

TOTAL CONTRIBUTION COEFFICIENTS

	1	2	3	4	5
1	.000E+00	.371E+00	.192E+00	.303E-01	.794E+00
2	.000E+00	.145E+00	.886E-01	.140E-01	.310E+00
3	.000E+00	.720E-01	.371E-01	.158E+00	.154E+00
4	.000E+00	.211E+00	.109E+00	.172E-01	.451E+00
5	.000E+00	.468E+00	.241E+00	.382E-01	.171E+00

TOTAL DEPENDENCY COEFFICIENTS

	1	2	3	4	5
1	.000E+00	.933E+00	.933E+00	.933E+00	.933E+00
2	.000E+00	.145E+00	.172E+00	.172E+00	.145E+00
3	.000E+00	.371E-01	.371E-01	.100E+01	.371E-01
4	.000E+00	.172E-01	.172E-01	.172E-01	.172E-01
5	.000E+00	.100E+01	.100E+01	.100E+01	.171E+00

ANALYSIS OF INPUT 1

INPUT VECTOR

	1	2	3	4	5
1	.100E+01	.000E+00	.000E+00	.000E+00	.000E+00

EXPORT VECTOR

	1	2	3	4	5
1	.268E-01	.213E-01	.000E+00	.000E+00	.718E-01

RESPIRATIONS

	1	2	3	4	5
1	.179E+00	.273E+00	.151E+00	.169E-01	.259E+00

EXCHANGE MATRIX

	1	2	3	4	5
1	.000E+00	.000E+00	.000E+00	.000E+00	.794E+00
2	.000E+00	.000E+00	.626E-02	.000E+00	.134E+00
3	.000E+00	.000E+00	.000E+00	.309E-01	.167E-01
4	.000E+00	.000E+00	.000E+00	.000E+00	.139E-01
5	.000E+00	.434E+00	.193E+00	.000E+00	.000E+00

ANALYSIS OF INPUT 5

INPUT VECTOR

	1	2	3	4	5
1	.000E+00	.000E+00	.000E+00	.000E+00	.100E+01

EXPORT VECTOR

	1	2	3	4	5
1	.000E+00	.268E-01	.000E+00	.000E+00	.904E-01

RESPIRATIONS

	1	2	3	4	5
1	.000E+00	.344E+00	.191E+00	.213E-01	.327E+00

EXCHANGE MATRIX

	1	2	3	4	5
1	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
2	.000E+00	.000E+00	.788E-02	.000E+00	.168E+00
3	.000E+00	.000E+00	.000E+00	.389E-01	.210E-01
4	.000E+00	.000E+00	.000E+00	.000E+00	.175E-01
5	.000E+00	.547E+00	.243E+00	.000E+00	.000E+00

*** LINDEMAN TROPHIC AGGREGATIONS ***

LINDEMAN TRANSFORMATION MATRIX

	1	2	3	4
1	.100E+01	.000E+00	.000E+00	.000E+00
2	.000E+00	.100E+01	.969E+00	.000E+00
3	.000E+00	.000E+00	.315E-01	.969E+00
4	.000E+00	.000E+00	.000E+00	.315E-01

EFFECTIVE TROPHIC LEVELS OF EACH SPECIES

	1	2	3	4	5
1	.100E+01	.200E+01	.203E+01	.303E+01	.100E+01

CANONICAL EXPORTS

	1	2	3	4	5
1	.300E+03	.255E+03	.000E+00	.000E+00	.860E+03

CANONICAL RESPIRATIONS

	1	2	3	4	5
1	.200E+04	.503E+04	.254E+03	.639E+01	.311E+04

THE GRAZING CHAIN

	1	2	3	4
1	.112E+05	.000E+00	.433E+03	.116E+02

RETURNS TO DETRITAL POOL

	1	2	3	4
1	.888E+04	.179E+04	.168E+03	.525E+01

DETRITIVORY = .75140E+04

INPUT TO DETRITAL POOL = .63500E+03

CIRCULATION WITHIN DETRITAL POOL = .00000E+00

LINDEMAN SPINE

	1	2	3	4
1	.138E+05	.751E+04	.433E+03	.116E+02

TROPHIC EFFICIENCIES

	1	2	3	4
1	.545E+00	.577E-01	.269E-01	.000E+00

*** BIOGEOCHEMICAL CYCLE ANALYSIS ***

2-CYCLE NEXUS WITH WEAK ARC (2, 3) = 75.000

1. 5-2-3-5-

2. 5-2-3-4-5-

1-CYCLE NEXUS WITH WEAK ARC (4, 5) = 167.000

3. 5-3-4-5-

1-CYCLE NEXUS WITH WEAK ARC (3, 5) = 200.000

4. 5-3-5-

1-CYCLE NEXUS WITH WEAK ARC (2, 5) = 1600.000

5. 5-2-5-

CYCLE DISTRIBUTIONS

	1	2	3	4	5
1	.000E+00	.352E+04	.521E+03	.137E+03	.000E+00

NORMALIZED DISTRIBUTION

1	2	3	4	5
.000E+00	.829E-01	.123E-01	.322E-02	.000E+00

FINN CYCLING INDEX IS .9839E-01

RESIDUAL FLOWS

	1	2	3	4	5
1	.000E+00	.000E+00	.000E+00	.000E+00	.888E+04
2	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
3	.000E+00	.000E+00	.000E+00	.203E+03	.000E+00
4	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
5	.000E+00	.353E+04	.202E+04	.000E+00	.000E+00

AGGREGATED BIOGEOCHEMICAL CYCLES

	1	2	3	4	5
1	.000E+00	.000E+00	.000E+00	.000E+00	.000E+00
2	.000E+00	.000E+00	.750E+02	.000E+00	.160E+04
3	.000E+00	.000E+00	.000E+00	.167E+03	.200E+03
4	.000E+00	.000E+00	.000E+00	.000E+00	.167E+03
5	.000E+00	.168E+04	.292E+03	.000E+00	.000E+00

*** INFORMATION INDICIES ***

TOTAL SYSTEM THROUGHPUT = .42445E+05
 DEVELOPMENT CAPACITY = .13586E+06
 ASCENDENCY = .56725E+05 (.418)
 OVERHEAD ON IMPORTS = .62218E+04 (.046)
 OVERHEAD ON EXPORTS = .78108E+04 (.057)
 DISSIPATIVE OVERHEAD = .35274E+05 (.260)
 REDUNDANCY = .29832E+05 (.220)

INTERNAL CAPACITY = .59164E+05
 INTERNAL ASCENDENCY = .29332E+05 (.496)
 REDUNDANCY = .29832E+05 (.504)

*** CONNECTANCE INDICES ***

OVERALL CONNECTANCE = 1.841
 INTERCOMPARTMENTAL CONNECTANCE = 1.375
 FOODWEB CONNECTANCE = 1.000

In terms of the output in Table II, first to appear in this section is an *ENUMERATION OF ALL THE SIMPLE CYCLES* in the given exchange matrix (only 5 in this case). The simple cycles are grouped into "nexuses" of cycles which share the same "weak arc". (See below for definition) The nexuses are always listed in ascending order of the magnitudes of each weak arc (the identity and magnitude of which is printed as a header to each nexus). The order of the cycles within any nexus is of no particular significance, but corresponds to the order in which the cycles were identified by the routine CYCLES. The number of cycles in the sample network is quite sparse, and the grouping by weak arc is almost trivial. However the Crystal River ecosystem of 17 compartments (see [3]) has 119 cycles. The number of possible cycles goes up as approximately the factorial of the number of compartments. Luckily, most ecosystems are very sparse, that is, very few of all possible connections are actually made.

Functional Cycles and their Removal

Once the structure of the cycles has been identified and recorded, the next step is to separate the flow network into 1) arcs associated with cycles and 2) arcs associated with straight-through flow. This process begins by identifying

the weak arc in each cycle. In each cycle, there is one arc which has the smallest flow through it. If this arc is removed and the value of the flow through it is subtracted from the flow in each of the other arcs in the cycle, then the cycle is effectively removed from the flow network. There is no other arc in the cycle whose flow rate could be set to zero (thus breaking the cycle) without some flows in some cycle arcs becoming negative. Thus, the weak arc is the cycle arc with the smallest flow through it.

It is most likely that an arc is part of more than one cycle. All the cycles which share the same weak arc are called collectively a nexus. When a weak arc is removed from the network, all the cycles in its nexus are broken. That portion of the flow through the weak arc which is subtracted from each of the arcs in each of the cycles, is determined by calculating the circuit probabilities. This is done by multiplying the fractional inflows for each cycle arc to obtain a weight for the whole cycle. The weight for each cycle is then summed for all the cycles in the nexus to give the nexus weight. The portion of the flow through the weak arc which is subtracted from each arc of a particular cycle is that cycle's weight divided by the nexus weight.

The process of removing the cycles from the network begins by finding the smallest weak arc and its nexus. The cycles in that nexus are then eliminated. This is done by subtracting the appropriate amount from each entry in the exchange matrix associated with a cycle arc from the nexus. Once a nexus has been eliminated, the updated exchange matrix is again searched for the smallest weak arc. This arc's nexus is then removed, and so on until all the cycles have been eliminated. The exchange matrix which remains at the end of this process represents the network of straight-through flows. The matrix which represents the cycles is found by subtracting the straight-through flow exchange matrix from the whole network exchange matrix.

As a cycle with m links is removed from the graph, the flow associated with that cycle is accumulated in the m -th component of an n -dimensional array. This *CYCLE DISTRIBUTION* array, therefore, shows how much flow is cycling in loops of various sizes. Such a profile might be useful in assessing system response to perturbation; for example, where cycling via larger loops might be more sensitive to disturbance. When the cycle distribution is normalized by the total system throughput, the result is the *NORMALIZED DISTRIBUTION*. Summing the normalized distribution yields the *FINN CYCLING INDEX*, or the fraction of all the flow in the system which is being cycled [8].

The results of subtracting all the cycled flow from the exchange matrix are reported as the *RESIDUAL EXCHANGES*, or straight-through (acyclic) flows listed after the Finn index. This residual exchange matrix can be used along with the original inputs, exports and respirations (which were unaffected by the cycle extraction technique) to construct the total graph of acyclic flows.

The last output pertaining to the cycle structure is the matrix of *AGGREGATED BIOGEOCHEMICAL CYCLES*. The corresponding row sums and column sums of this matrix will always balance; no further reference to exogenous exchanges is necessary. When

depicted as a network the aggregated cycles form one (or more) closed nexus(es). The visual structure of this network when computed from more complicated input arrays very often reveals the domains of control in the network. It should be noted that the starting network has been decomposed into an acyclic "tree" of dissipative flows and a wholly conservative nexus of cycled flows.

INFORMATION INDICES

The final short segment of output provides values for global attributes of the network as defined by Ulanowicz in his theory of pattern development.[3,4,20,21,22]. The total system throughput has already been defined and serves as a measure of the size of the system. Multiplying the total throughput by the diversity (according to the Shannon Wiener formula) of the individual flows yields the development capacity. This quantity is a measure of the network's potential for competitive advantage over other real or putative network configurations. Ascendency is the product of a factor of size (total system throughput) times a factor representing the coherence of the flows (the average mutual information of the flow structure.) Its upper bound is the development capacity. It measures the degree of articulation in the network.

The difference between the realized structure and its upper bound is the overhead and occurs for any of four reasons -- there is overhead due to uncertainty about imports, exports and dissipations (respirations), and the flows are proceeding along parallel pathways (redundancy). The fractions of the development capacity encumbered by each of its five components are given in parentheses following the values.

The foregoing indices were descriptors of the entire system. Ulanowicz [3,20] shows how growth and development in networks are best characterized by components of the internal capacity (as calculated over only internal exchanges). There are two -- the internal ascendency and the internal redundancy.

Finally, these global indices are recast so as to describe the effective number of connections between any two arbitrary compartments. The overall connectance includes the effects of exogenous transfers, while the intercompartmental value characterizes only the endogenous exchanges. [24] The foodweb connectance pertains only to transfers among the living compartments

PROGRAMMING CONSIDERATIONS

The software itself is made up of an input module, four separate computational modules and an output module. Input is assumed to be in the form specified and considerable checking is done to make sure the data is sensible. The authors have written several routines to convert other data formats to the required format. One of these routines allows the user to input data from the terminal device. (These are available from the authors upon request.)

Except for the cycle routine, the calculations are all simple matrix operations, the most complex being matrix inversion. The only routine, except for cycle analysis, which is time consuming is "the analysis of inputs" in the analysis of structure. The issue here is the number of matrix operations required and the vast amount of output

generated.

The cycle routine consists of a highly recursive set of procedures and was the greatest programming challenge. This analysis is well handled by a linked list data structure such as those in Pascal, but can also be accomplished in FORTRAN, albeit less efficiently. Because of the number of recursive calls made to the procedures, stack and heap size can restrict the size of the ecosystem which can be analysed. The FORTRAN version is restricted to a maximum nexus size of 300 cycles. The Pascal version with 256K of stack/heap available can deal with about 1,300 cycles for a 32 compartment system. (The smaller the number of compartments, the larger the number of cycles which can be handled.) With a 2Mg Macintosh or mainframe, the number of cycles which can be handled is much larger (about 10,000). Users of NETWRK in an MS-DOS environment are the most heavily restricted.

To deal with this problem Ulanowicz has written a procedure which decreases the number of cycles to be dealt with by redefining what constitutes a weak arc. This is done by setting an upper limit on the size of the flow through an arc, above which it will not be considered a weak arc. The upper limit is iteratively adjusted downward until all the eligible cycles fit into core. All cycles with a flow value for their weak arc less than the upper limit are removed. The upper limit is then increased and more cycles are removed. This process continues until all cycles are removed. This procedure increases CPU time substantially.

In general, execution time is quite fast, in the order of seconds to 2 minutes on micro computers. The largest percentage of the execution time is used by the cycle analysis. The biggest real ecosystem the authors have tested took 10 minutes to run. The 1,300 cycle hypothetical test data set took 1 1/2 hours to run on a Macintosh.

The basic package assumes one data set to be analysed. It is relatively easy to extend this to sets of data run in batch mode, especially in the FORTRAN version. The Pascal version takes advantage of Pascal's ability to control events, to allow the user to interactively change the data and rerun the analysis. Thus, the user can play "what-if" games. This is important if the user wishes to evaluate the utility of some of the measures.

The output generated by NETWRK can be daunting. The analysis of a real 32 compartment ecosystem results in a 256K output file. Most of the output is due to the "analysis of inputs". A future version of NETWRK will allow the user to select which analyses are to be performed.

All of the output from NETWRK is in matrix form. It takes some labour on the part of the user to convert this numerical information into a pictorial representation of the flows in the ecosystem. Such pictorial information, for example a picture of the cycles, is usually required as an aid in ecosystem flow analysis. This bottleneck is not peculiar to NETWRK. It is a problem for all analysis of flow networks currently available. The authors are now working on a module to automate the preparation of graphic results.

The reader may wonder why both FORTRAN and Pascal versions of NETWRK exist. From a programmers perspective, Pascal is a better vehicle for this package in a

micro computer environment. However, it has been the authors experience that most users (i.e. biologists) are more comfortable with FORTRAN. Therefore, to make the program accessible to the intended audience, a FORTRAN version has been retained.

SUMMARY

For most of its history, quantitative ecology has relied on either statistics or simulation modeling as tools for the analysis of ecosystems. However, statistics has only rarely been applied to entire ecosystems, while simulation modeling of entire communities has yielded few reliable predictions. Ecologists and managers interested in the structure and behavior of whole ecosystems require more useful and reliable techniques. NETWRK affords these investigators new and promising methods of analysis. [25]

The data required by NETWRK and the format in which information is supplied to the program guide the user in gathering complete and systematic sets of data. The format also allows for compactness of data storage, ease of editing, and simplicity in manual interpretation.

The analyses executed in NETWRK cover the hierarchical gamut from emphasis on individual taxa to the description of the properties of the entire community. Compartmental indices, such as the throughput and average trophic level of each taxon, show how each population stands with respect to the others in the community. The input-output analyses allow the user to assess the magnitude and type of any bilateral trophic relationship, including indirect trophic influences. The Lindeman trophic aggregation scheme is like an x-ray of the backbone of the trophic dynamics of a complicated network. Community responses to perturbations will appear in this context as changes in the amounts reaching the higher trophic levels and differences in trophic efficiencies along the aggregated chain. Similarly, it may be possible to read the pattern of controls in the system from the configuration of the recycle pathways inherent in the network. Finally, theoretically defensible indices of whole-system trophic status allow the user to determine when the system as a whole has been significantly impacted. Ratios and relationships among these indices permit one to address quantitatively such heretofore ambiguous notions as the "health", "integrity" or "well-being" of the ecosystem. [26, 27]

The calculations have been tested extensively over the last nine years, and once the traps for errors in the input have been negotiated, it is rare that problems with the execution of the program are encountered. A wealth of quantitative output results from running the program, and abundant literature exists to help the user interpret the various elements of the output. Work is underway to allow the user to select particular sections of the program for execution and reporting. Furthermore, the authors intend to incorporate graphical routines that will render the output in a form that facilitates diagnosis of the system behavior.

The methods employed in NETWRK involve a minimum of a priori assumptions. Therefore, the results represent interpretations of the input information; and one may issue pronouncements about the ecosystem based on the output from NETWRK with much the same level of

confidence that one has in the input data itself. NETWRK is a reliable tool that possibly will revive the now-sagging interest in quantitative ecosystems analysis.

The authors invite any and all user comments about the program. Copies of NETWRK are available from either author by sending him an appropriate diskette on which to return the program.

REFERENCES

1. Hannon, B. 1973. The structure of ecosystems. *J. theor. Biol.* 41:535-546.
2. Patten, B.C., (ed.) 1976. *Systems Analysis and Simulation in Ecology*. (4 volumes) Academic Press, New York.
3. Ulanowicz, R.E., 1986. *Growth and Development: Ecosystems Phenomenology*. Springer-Verlag, New York. 203p.
4. Kay, J.J., L. Graham, and R.E. Ulanowicz. 1989. "A Detailed Guide to Network Analysis", in *Network Analysis in Marine Ecology: Methods and Applications*, F. Wulff, J. Field, K. Mann (Eds.), Springer Verlag, Heidelberg.
5. Ulanowicz, R.E., 1986. "A Phenomenological Perspective of Ecological Development" pp. 73-81 in T.M. Poston and R. Purdy (eds.) *Aquatic Toxicology and Environmental Fate: Ninth Volume*. ASTM STP 921, American Society for Testing and Materials, Philadelphia.
6. Lindeman, R.L. 1942. The trophic-dynamic aspect of ecology. *Ecology* 23:399-418.
7. Leontief, W. 1951. *The Structure of the American Economy, 1919-1939*. 2nd edn. Oxford University Press, New York.
8. Finn, J.T. 1976. Measures of ecosystem structure and function derived from analysis of flows. *J. theor. Biol.* 56:363-380.
9. Levine, S.H. 1980. Several measures of trophic structure applicable to complex food webs. *J. theor. Biol.* 83:195-207.
10. Patten, B.C., R.W. Bosserman, J.T. Finn and W.G. Cale. 1976. "Propagation of cause in ecosystems" Pages 457-479 in B.C. Patten, (ed), *Systems Analysis and Simulation in Ecology*.
11. Ulanowicz, R.E., and W.M. Kemp. 1979. Toward canonical trophic aggregations. *Am. Nat.* 114:871-883.
12. F. Wulff, J. Field, K. Mann (Eds.), *Network Analysis in Marine Ecology: Methods and Applications*, Springer Verlag, Heidelberg 1989
13. Augustinovic, M. 1970. "Methods of international and intertemporal comparison of structure". In A.P. Carter and A. Brody (eds.): *Contributions to Input-Output Analysis*, Vol. 1. North Holland: Amsterdam.
14. Szyrmer, J. and R.E. Ulanowicz. 1986. "Total flow in ecosystems", *Ecol. Mod* 35:123-136.
15. Ulanowicz, R.E. 1990. "Ecosystem trophic foundations: Lindeman exonerata", To appear in B.C. Patten and S.E. Jorgensen (eds.): *Complex Ecology: The Part-whole Relationship in Ecosystems*, Vol. 2, Prentice-Hall: New York.
16. Pimm, S.L. 1982. *Food Webs*. Chapman and Hall, London. 219p
17. Ulanowicz, R. 1983 "Identifying the structure of cycling in ecosystems", *Math. Biosci.* 65: 219-237.
18. Kay, J.J. 1984. *Self-organization in living systems*, Ph.D. Thesis, Department of Systems Design Engineering. University of Waterloo: Waterloo, Ontario.
19. Horowitz, E., and S. Sahni 1984. *Fundamentals of data structures in Pascal*, Computer Science Press.

20. Hirata, H. and R.E. Ulanowicz. 1984. Information theoretical analysis of ecological networks. *Int. J. Systems Sci.* 15(3):261-270.
21. Ulanowicz, R.E. 1980. An hypothesis on the development of natural communities. *J. theor. Biol.* 85:223-245.
22. Ulanowicz, R.E. and J.S. Norden. 1989. Symmetrical network overhead. *Int. J. Systems Science* 21:429-437.
23. Kay, J., 1989, "NETWRK 4" presented at the annual meeting of the International Society for Ecological Modelling (North America), August 8, University of Toronto.
24. Ulanowicz, R.E. 1989. "A Window of Vitality: Bounds on the Complexity of Ecosystem Flow Networks. Ref. No. 89-140 CBL, Chesapeake Biological Laboratory.
25. Baird, D., Ulanowicz, R.E., 1989. The Seasonal Dynamics of the Chesapeake Bay Ecosystem, *Ecol. Mono.*, 59(4):329-364.
26. Kay, J.J., 1991. "A Non-equilibrium Thermodynamic Framework for Discussing Ecosystem Integrity", *Environmental Management*, (in press)
27. Kay, J.J., Schneider, E.D., 1991. "Thermodynamics and Measures of Ecological Integrity" in *Proceedings of the International Symposium on Ecological Indicators*, Fort Lauderdale, Florida, Elsevier.(in press)