COENOSES 6 (2): 71-78, 1991

PARAMETER-DEPENDENT EQUILIBRIA IN DYNAMIC MODELS OF ECOLOGICAL COMMUNITIES. A QUALITATIVE APPROACH WITH MACSYMA PROGRAMS

Fabio Cavallini, Osservatorio Geofisico Sperimentale - 34016 Trieste, Italy

Keywords: Ecological communities, Qualitative modelling, Symbolic computation, Equilibria, Qualitons

Abstract. The notion of qualitative model is pointed out and its role in theoretical ecology is discussed. Concepts and algorithms pertaining the stability of equilibria and their dependence from external input parameters are reviewed in the framework of qualitative modelling. The effectiveness and efficiency of symbolic computer languages in tackling these kinds of problems is shown through the detailed examination of case-studies borrowed from the literature and their solution via MACSYMA programs composed by case-specific input statements and general-purpose computing subroutines. It is also shown that a linear system cannot model globally a competitive or a predator-prey interaction if the state components are to be interpreted as absolute population levels. Hence the mathematical methods considered here are suitable only for a local analysis of the community dynamics in a neighborhood of an equilibrium state, nevertheless it appears that the use of symbolic computation has a much wider application range and can become an outstanding software tool in every branch of qualitative ecological modelling.

> Qualitative is nothing but poor quantitative.
> E. Rutherford

> What matters most for everyday use is almost always a qualitative result and not the precise value of a real number.
> R. Thom

1. Introduction

What is quality, as opposed to quantity? In a philosophical setting this is certainly a quite vague and challenging question, but in mathematics the notion of qualitative property can be given a rather narrow and well defined meaning. Indeed, elements in a set S are classified according to a qualitative property when in S an equivalence relation is defined such that the set of its equivalence classes is finite. In other words, a qualitative property can be expressed by a finite collection of words whereas a quantitative property requires using a numerical index varying in an infinite set of values. For example, monotonicity is a qualitative property since it can be expressed using the following five expressions: constant, strictly increasing, strictly decreasing, nonincreasing and nondecreasing. Analogously, (Lyapunov) stability is a qualitative property since it can be expressed using the following three expressions: asymptotically stable, neutrally stable and unstable.

Qualitative models are those in which only qualitative properties of the process parameters are known, but not their exact numerical values. For instance, in the case of constant parameters only the sign (positive, negative or zero) but not the magnitude is given. In other cases process parameters are functions of the state variables or of external inputs, but the functional dependence is not exactly known: only qualitative properties are given, such as monotonicity with respect to some argument. Of course if the process parameters are only qualitatively known, the goal of the modeller is the investigation of the qualitative properties of solutions (notably: stability and parameter sensitivity) since in this case it is impossible to compute exactly the time history of the system.

The role of qualitative modelling in the biological sciences and ecology in particular is a crucial one in so far as the numerical values of parameters governing these processes are usually very poorly known. This situation somewhat contrasts with that in "hard sciences" and technology where the process parameters can be measured accurately or they are a priori known as they are the outcome of the engineer's designing activity.

Traditionally, computers have been used in science and engineering mainly for quantitative modelling and this is largely due to the high performance of computers as number-crunching machines. However, symbolic manipulation can also be dealt with automatically, either by a skilful use of general-purpose or numeric-oriented languages (FORTRAN, PASCAL, C, ...) or by taking advantage of symbolic-oriented languages (LISP, PROLOG, ...) and of suitable high-level software tools (MACSYMA, MATHEMATICA, ...). Day and Puccia (1988) developed a large program in C for computing the table of predictions of equilibria (cf. next section) via the implementation of sophisticated graph-theoretic algorithms. In the present paper the qualitative study of parameter sensitivity and stability of equilibria is performed through the application of basic algebraic concepts and algorithms to case-studies borrowed from the literature: MACSYMA procedures are shown that completely solve the special problems considered and can be adapted in a straightforward way to handle similar situations. For a comprehensive introduction to qualitative modelling with emphasis on ecology, we refer to Puccia and Levins (1985) and for MACSYMA to Pavelle and Wang (1985).

2. Parameter-dependent equilibria of nonlinear systems

Let us consider a differentiable dynamical system given by a set of differential equations of the form:

$$\frac{dx_i(t)}{dt} = f_i(x_1(t), \ldots, x_n(t); p) \tag{2.1}$$

or, in vector notation:

$$\frac{dx(t)}{dt} = f(x(t); p) \tag{2.2}$$

where $x=(x_1, \ldots, x_n) \in R^n$ is the state of the system and $p \in R$ is the external input parameter, assumed constant. System (2.1) may also be represented pictorially by means of a so-called signed digraph in which a topological graph describes the whole system by associating a node to each state-variable x_i and by connecting node x_i to node x_j with an arrow when $f_j(.)$ is not constant with respect to x_i. Moreover, a + (—) sign on the arrow from x_i to x_j indicates that $f_j(.)$ is an increasing (decreasing) function of x_i. Analogously, an arrow from outside the system to node x_i means that $f_i(.)$ is not constant with respect to p; moreover, a + (—) sign on the arrow indicates that $f_i(.)$ is an increasing (decreasing) function of the input parameter.

Example 2.1. (Adapted from Puccia and Levins (1985), p. 45).

Let us consider a simple planktonic community composed by an algal species, an herbivore and a predator whose biomass concentrations are A, H and P, respectively. It is further assumed that nutrient concentration ν increases algal growth rate only. A reasonable dynamic interaction model might be:

$$\begin{aligned} \dot{A} &= f_A(A, H, \nu) \\ \dot{H} &= f_H(A, P) \\ \dot{P} &= f_P(H) \end{aligned} \tag{2.3}$$

where a dot indicates time-differentiation and the following assumptions are understood:

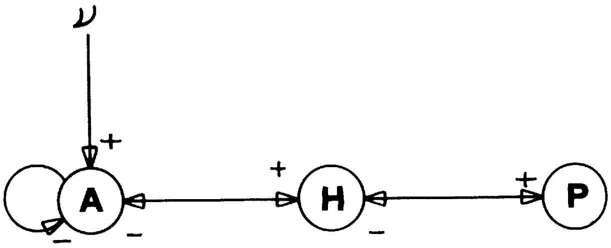

Fig. 1. Signed digraph representing the externally forced three-component planktonic community considered in Example 2.1: A = alga, H = herbivore, P = predator and ν = nutrient concentration.

$$\begin{aligned} &\frac{\partial f_A}{\partial A} < 0, \quad \frac{\partial f_A}{\partial H} < 0, \quad \frac{\partial f_A}{\partial \nu} > 0 \\ &\frac{\partial f_H}{\partial A} > 0, \quad \frac{\partial f_H}{\partial P} < 0 \\ &\frac{\partial f_P}{\partial H} > 0 \end{aligned} \tag{2.4}$$

Now, equations (2.3) with side conditions (2.4) are represented in an exactly equivalent way in Fig. 1.

An equilibrium for system (2.1) is defined to be a state x^* such that the constant function $x(t) = x^*$ is a solution of system (2.1). Hence x^* is an equilibrium if and only if:

$$f(x^*; p) = 0 \tag{2.5}$$

Then, by the implicit function theorem, every equilibrium state x^* results to be a function of the input parameter p:

$$x^* = x^*(p) \tag{2.6}$$

It is now natural to ask how the components $x_i^*(p)$ of an equilibrium are affected by a small change in p. Substituting (2.6) into (2.5) and differentiating yields:

$$A \frac{dx^*}{dp} + b = 0 \tag{2.7}$$

where the jacobian matrix A and the sensitivity vector b are given by:

$$A_{ij} = \frac{\partial f_i}{\partial x_j}(x^*(p); p) \text{ and } b_i = \frac{\partial f_i}{\partial p}(x^*(p); p) \tag{2.8}$$

From (2.7) we get:

$$\frac{dx^*}{dp} = -A^{-1} b \tag{2.9}$$

i.e.:

$$\frac{dx_i^*}{dp} = -\sum_j (A^{-1})_{ij}\, b_j \quad \forall\, i \tag{2.10}$$

If we now assume that only f_k (.), with k fixed, depends on parameter p, then we get that $b_k = \partial f_k/\partial p$ is the only nonzero component of b. Hence (2.10) yields:

$$\frac{dx_i^*}{dp} = -(A^{-1})_{ik}\, b_k \tag{2.11}$$

whence:

$$\text{signum}\,\Big(\frac{dx_i^*}{dp}\Big) = -\text{signum}\,(A^{-1})_{ik}\,.\ \text{signum}\,(b_k) \tag{2.12}$$

In agreement with Puccia and Levins (1982) we now introduce the table of predictions as the matrix T whose generic entry is given by:

$$T_{ij} = -\text{signum}\ [(A^{-1})_{ji}] \tag{2.13}$$

The sign of b_k is usually a datum of the problem: hence, by (2.12), the knowledge of $T_{k,\,i}$ allows one to predict whether x_i^*, the equilibrium level of the i-th state variable, will increase, decrease or be stationary with respect to changes in the input parameter p under the assumption that p affects the growth-rate of only the k-th state variable x_k .

Table 1. A MACSYMA subroutine for computing the table of predictions for an arbitrary jacobian matrix A of arbitrary order n.

```
prediction_table (a, n, pt):=
block (
a_inv: a^-1,
a_inv_tr: transpose (a_inv),
predic_tabl [i, j]:=
ratsimp (factor (expand (signum (-a_inv_tr [i, j])))),
pt: genmatrix (predic_tabl, n, n)
);
```

In table 1 a MACSYMA subroutine is shown which computes the table of predictions for an arbitrary jacobian matrix. As the following example will show, it may be assumed that only the signs, but not the magnitudes, of the jacobian matrix entries are known; if the sign of an expression cannot be determined, the whole symbolic expression is displayed, as will be illustrated in example 2.3. The codes presented here are almost self-explanatory, so we only remark that all computations were made after customizing the MACSYMA computational environment by the directive:

```
declare (signum, multiplicative);
```

which "teaches" the machine the mathematical rule:

$$\text{signum (a.b)} = \text{signum (a).signum (b)}$$

whose use is essential in order to automatically simplify expressions involving the signum (.) function.

Table 2. A complete MACSYMA program for computing the table of predictions for the jacobian matrix of the planktonic ecosystem considered in examples 2.1 and 2.2. Note that use is made of the subroutine shown in table 1.

```
assume_pos: true;
/* input the jacobian matrix */
n: 3;
r1: [-a11, -a12, 0];
r2: [a21, 0, -a23];
r3: [0, a32, 0];
a: matrix (r1, r2, r3);
/* ********************************************* */
prediction_table (a, n, pt);
```

Example 2.2. We consider again the situation of example 2.1 and compute the table of predictions using the MACSYMA program shown in table 2; the result is:

$$T = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & -1 & 1 \end{bmatrix}$$

Since it was assumed that only the growth rate of the first variable is parameter-sensitive, only the first row of T is to be taken into consideration. Remembering that $\partial f_A/\partial \nu$ is positive, we see that an increase in nutrient concentration will increase both A* and P* while leaving H* stationary.

The following example will show that the subroutine shown in table 2.1 can cope also with higher-order systems, even when some entries of the prediction table are not well-defined.

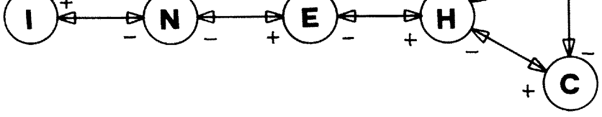

Fig. 2. Signed digraph representing the six-component planktonic community considered in Example 2.3: N = Nutrients, I = Inedible algae, E = Edible algae, H = herbivores, C = Carnivores and P = Planktivores.

Table 3. A complete MACSYMA program for computing the table of predictions for the jacobian matrix of the lake planktonic ecosystem considered in example 2.3. Note that again use is made of the subroutine shown in table 1.

```
assume_pos: true;
n: 6;
r1: [−a11, −a12, −a13, 0, 0, 0];
r2: [a21, 0, 0, 0, 0, 0];
r3: [a31, 0, 0, −a34, 0, 0];
r4: [0, 0, a43, 0, −a45, −a46];
r5: [ 0, 0, 0, a54, 0, −a56];
r6: [0, 0, 0, a64, a65, 0];
a: matrix (r1, r2, r3, r4, r5, r6);
/* ********************************************** */
pred_tabl: prediction_table (a, n, pt);
```

Example 2.3 (Adapted from Briand and Mc Cauley, 1978)

Let us consider a lake planktonic community described by the following six state variables: Nutrients (N), Inedible algae (I), Edible algae (E), Herbivores (H), Carnivores (C) and Planktivores (P). The inter-specific interactions are assumed to be as in figure 2 Using the MACSYMA program shown in Table 3, the following table of prediction is obtained:

$$T=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & \alpha & \beta & -1 & 1 & -1 \\ 0 & \beta & -\beta & 1 & -1 & 1 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & -1 & 0 \end{bmatrix}$$

where:

$$\alpha=\text{signum}\ ((A_{11}\ A_{34}\ A_{43}\ A_{56}+A_{13}\ A_{31}\ A_{46}\ A_{54})\ A_{65}-A_{13}\ A_{31}\ A_{45}\ A_{56}\ A_{64})$$

$$\beta=-\text{signum}\ (A_{46}\ A_{54}\ A_{65}-A_{45}\ A_{56}\ A_{64})$$

Looking at these equations, one can guess suitable assumptions under which the expressions α and β can acquire a definite numerical value. The computer implementation of such assumptions is straightforward. For example, assuming $\Delta=A_{46}\ A_{54}\ A_{65}-A_{45}\ A_{56}\ A_{64}=0$ amounts to adding the following statement at the end of the program in table 3:

```
ratsubst (A46*A54*A65, A45*A56*A64, pred_tabl);
```

and the result is:

$\alpha=1$, $\beta=0$

Alternatively, assuming $\Delta>0$ amounts to adding the following statement at the end of the program in table 3:

```
ratsubst (A46*A54*A65—delta, A45*A56*A64,

pred tabl);
```

and the result is:

$\alpha=1$, $\beta=-1$

Finally, assuming $\Delta<0$ amounts to adding the following statement at the end of the program in table 3:

```
ratsubst (A46*A54*A65+epsilon, A45*A56*A64,

pred_tabl);
```

and the result is:

$\alpha=-\text{signum}\ (A_{13}\ A_{31}\ \epsilon-A_{11}\ A_{34}\ A_{43}\ A_{56}\ A_{65})$, $\beta=1$

where $\epsilon=-\Delta$, so that in this case one ambiguity still remains (of the four we had initially).

3. The stability requirement and linearization

The results obtained in the preceding section apply to stable and unstable equilibria as well; however, the problem of distinguishing between these two kinds of equilibria is of crucial ecological importance in itself. Indeed, only asymptotically stable equilibria can be observed, since even an arbitrarily small external perturbation leading to a state very near, but different from, an equilibrium state non asymptotically stable would induce into the system a time-evolution that will never return to the original steady-state situation. Several criteria for detecting asymptotically stable equilibria are known since the end of the 19th century, but their computational implementation is still object of intensive research because stability algorithms are usually very time-consuming and have very severe memory requirements. Moreover, this problem becomes still much harder in the framework of qualitative modelling, since the standard numerical techniques cannot be implemented as such when only the signs, but not the magnitudes, of the jacobian matrix entries are known.

Puccia and Levins (1985) as well as Svirezhev and Logofet (1983) tackle this problem using fairly sophisticated graph-theoretic algorithms. The aim of the present section is to show through several case-studies that one of the most basic stability criteria, due to Hurwitz (1895), can be successfully applied in a straightforward way to the analysis and the synthesis of equilibria in the framework of the qualitative theory, using only elementary matrix-algebra and automatic symbolic computation. By analysis of equilibria we mean the determination of the stability properties of a given system; by synthesis we mean the determination of suitable conditions (e.g. in the form of inequalities) on the coefficients of the system in such a way that the system has a pre-assigned stability property, as will be illustrated in Example 3.1.

Since the word "stability" means different things to different people, we briefly point out the main concepts and theorems that will be used in the sequel, referring to Hirsch and Smale (1974) or Pontryagin (1962) for a general introduction to the subject and to Gantmacher (1964) for a detailed presentation of Hurwitz criterion.

An equilibrium state x^* is called asymptotically stable if for any neighbourood U of x^* there exists a neighbourood V of x^* such that any trajectory starting in V ends in U.

It is well known that an equilibrium state x^* of the dynamical system defined by:

$$\dot{x} = v\ (x)$$

is asymptotically stable if the origin is an asymptotically stable state of the linearized system:

$$\dot{y} = Ay \tag{3.1}$$

where A is the *jacobian matrix* of the system, whose entries are $A_{ij} = \frac{\partial v_i}{\partial x_j}(x^*)$.

In order to check that the origin is an asymptotically stable state of system (3.1), we shall use the following Hurwitz algorithm:

- compute the coefficients c_1, ..., c_n of the characteristic polynomial of matrix A:

$$\det(\lambda\ I - A) = \lambda^n + c_1\ \lambda^{n-1} + \ldots + c_n$$

- consider the Hurwitz matrices:

$$H_k = \begin{bmatrix} c_1 & c_3 & c_5 & \cdots & 0 \\ 1 & c_2 & c_4 & \cdots & 0 \\ 0 & c_1 & c_3 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & c_{k-2} & c_k \end{bmatrix} \quad ; \quad k = 1, \ldots, n$$

- compute the Hurwitz multi-index:

$$(\text{signum}\ (\det H_1), \ldots, \text{signum}\ (\det H_n))$$

- note that asymptotical stability is equivalent to:

$$\det H_k > 0 \quad \forall k = 1, \ldots, n.$$

In table 4 a MACSYMA subroutine is shown which computes the Hurwitz multi-index of an arbitrary matrix with symbolic entries.

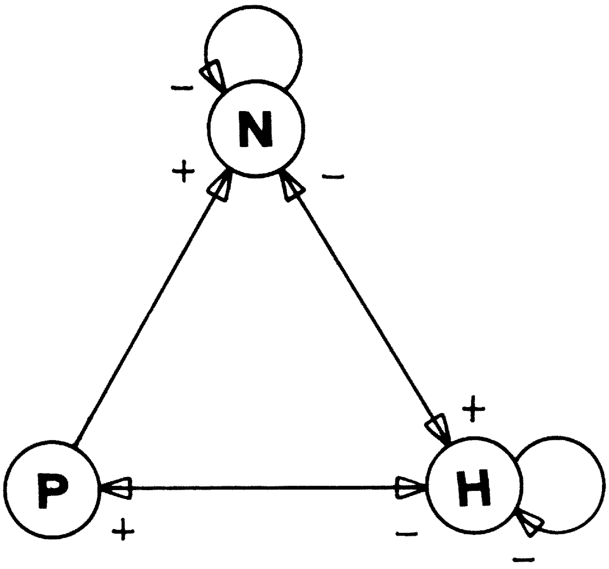

Fig. 3. Signed digraph representing the three-component community considered in example 3.1: P = predator, H = herbivore and N = nutrient.

Table 4. A MACSYMA subroutine for computing the Hurwitz multi-index of an arbitrary matrix A of arbitrary order n.

```
hurwitz_multi_index (a, n, hmi):=
block (
char_pol: expand ((—1)∧n*charpoly (a, x)),
/**/
hurwitz_array_2 [i, j]:=
   if n—2*j+i>=0 and n—2*j+1<=n
   then coeff (char_pol, x, n—2*j+i)
   else 0,
hurwitz matrix: genmatrix (hurwitz_array_2, n, n),
/**/
hurwitz_array_1 [i]:=
      determinant (genmatrix (hurwitz_array_2, i, i)),
hurwitz_vector: genvector (hurwitz_array 1, n),
/**/
hurwitz_index [i]:=
factor (expand (signum (hurwitz_array_1 [i]))),
hmi: genvector (hurwitz_index, n));
```

Example 3.1 (Adapted from Puccia and Levins, 1985, p. 26)

Let us consider a simple ecosystem in which a predator (P) secretes a nutrient (N) consumed by an herbivore (H), as illustrated in figure 3. Using the program shown in table 5 we obtain that the Hurwitz multi-index is given by:

$$[1,\ 1,\ \text{signum}\ (A_{13}\ A_{22} - A_{12}\ A_{23})]$$

Table 5. A complete MACSYMA program for investigating the stability of equilibria in the predator-prey-nutrient ecosystem considered in example 3.1. Note that use is made of the subroutine shown in table 4.

```
/* input matrix */
n: 3;
assume_pos: true;
row 1: [—a11, a12, —a13];
row 2: [—a21, —a22, a23];
row 3: [a31, 0, 0];
a: matrix (row 1, row 2, row 3);
/* ---stability analysis--- */
hurwitz_multi_index (a, n, hmi);
```

Looking at the last expression, one immediately sees that the system is asymptotically stable if and only if the following conditions is fulfilled: $A_{13}\,A_{22} > A_{12}\,A_{23}$.

Example 3.2 (Adapted from Gatto and Rinaldi, 1989)

Let us consider the qualitative model of acid deposition in forests given by the following set of ordinary differential equations:

$$\dot{T}=[-m+e\ (S)\ h\ (T)]T$$

$$\dot{S}=\beta N-e\ (S)\ h\ (T)\ S$$

$$\dot{N}=W-\alpha N-\beta NT+\eta mTS$$

where

T is tree living biomass
S is nutrient concentration (per unit biomass) in the trees
N is nutrient concentration in the soil
e (S) is efficiency of production, an increasing function of S
h (T) is maximum production of new biomass per unit biomass, a decreasing function of T
m, β, α, η are constant process parameters
w is a constant external nutrient input

and all variables are positive-definite.

The authors chaim that whenever a nontrivial equilibrium exists, it is asymptotically stable. However they do not report the complete proof because it "... is straightforward but cumbersome". This is certainly correct if the calculations are to be worked out with paper and pencil, but, using the short program shown in Table 6, one obtains at a keystroke that the Hurwitz multi-index of the linearized system at a nontrivial equilibrium is (1, 1, 1), which ensures asymptotic stability.

4. The role of linear dynamical system in mathematical ecology

In the preceding sections we have used extensively the theory of linear autonomous dynamical systems, but only in a local sense: for investigating the change of equilibrium levels under an infinitesimal growth of an input parameter and for determining the system dynamics in a small neighbourhood of an equilibrium state. Since the theory of linear systems is very well developed, one is led to hope that this approach might be used to model, at least approximately, also the global behaviour of ecological communities. We shall soon see that in general this is not the case, at least if the state variables are to represent positive-definite quantities such as biomass concentrations and competition is to be taken into account. Indeed, an obvious consistency requirement of any ecological model is that negative population levels are excluded. Now, let us consider a smooth dynamical system described by:

$$\dot{x}=v\ (x) \tag{4.1}$$

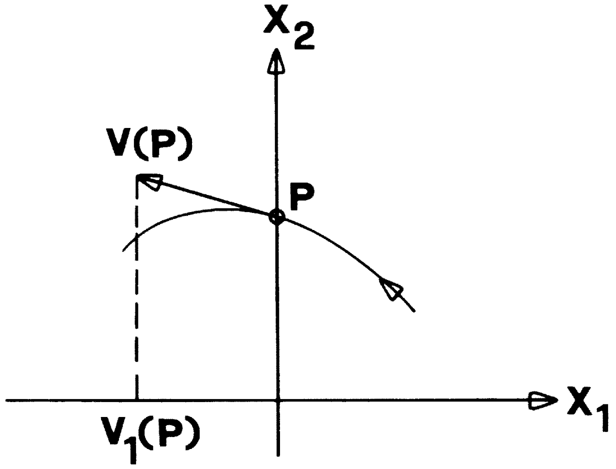

Fig. 4. The trajectory through $P\equiv(x_1=0,\ x_2)$ can leave the positive quadrant because at P the first component of velocity, v_1 (P), is negative.

Table 6. A complete MACSYMA program for investigating the stability of a nontrivial equilibrium in the qualitative model of acid deposition in forests considered in example 3.2. Note that use is made of the subroutine shown in table 4.

```
/* input matrix */
assume_pos: true;
m: e*h;
et: 1—etb;
a11:—e*hl*t;
a12: el*h*t;
a21:+e*hl*s;
a22:—h*(el*s+e);
a31:—(l—et)*m*s;
a32: et*m*t;
rl: [a11, a12,  0];
r2: [a21, a22,  b[;
r3: [a31, a32, —al—b*t];
a: matrix (rl, r2, r3);
n: 3;
/* *** stability analysis ****************************  */
hurwitz_multi_index (a, n, hmi);
```

System (4.1) will be called ecological if any trajectory starting at a point with non-negative components will always have non-negative components.

It is intuitive that a necessary and sufficient condition for system (4.1) to be ecological is the following:

$$v_i(x) \geq 0 \quad \text{for any } x=(x_1 , \ldots, x_n) \quad \text{such that} \quad x_i=0$$

Indeed, a trajectory x (t) exits the positive orthant at a boundary face of equation $x_i=0$ if and only if its velocity in the i-th direction is negative (cf. figure 4).

Example 4.1 (Volterra-Lotka equations)

A two-species community composed by a predator x_1 and a prey x_2 may be described by the classical Volterra-Lotka equations:

$$\dot{x}_1=x_1 (A\, x_2-B) \equiv v_1 (x_1 , x_2)$$

$$\dot{x}_2=x_2 (-C\, x_1+D) \equiv v_2 (x_1 , x_2)$$

Hence condition (4.2) is fulfilled since:

$$v_1 (0, x_2)=v_2 (x_1 , 0)=0; \quad \forall\, x_1, x_2$$

Example 4.3.

We consider again the situation of example 3.2 and note that condition (4.2) is fulfilled, since:

$$\dot{T} (T=0, S, N)=0$$

$$\dot{S} (T, S=0, N)=\beta N>0$$

$$\dot{N} (T, S, N=0)=W+\eta mTS>0$$

Assume now that system (4.1) is linear, i.e. v (x) = A x, where A is a constant matrix. Then (4.2) yields:

$$\sum_{j\neq i} A_{ij}\, x_j \geq 0; \quad \forall\, (x_1 , \ldots, x_n), \quad \forall\, i \tag{4.3}$$

Letting k ($\neq$i) be fixed and choosing $x_k=1$, $x_j=0$ for any $j\neq k$, we get from (4.3):

$$A_{ik} \geq 0, \quad \forall\, i\neq k \tag{4.4}$$

which implies that neither competitive nor predator-prey inter-specific interactions are allowed.

5. Discussion and concluding remarks

Through the case-studies considered above, we have seen that computer languages for symbolic computation such as MACSYMA are user-friendly and powerful tools for modelling the qualitative dynamic behaviour of ecological communities. Now it is perhaps worthwhile to discuss the limitations of the method as well as possible alternatives. Symbolic computations are time-consuming and memory-requiring; what is worse, the root of the problem is not on the computer side but in the very idea of symbolic manipulation. To illustrate this point, consider the operation of taking the determinant, to which all the algorithms of the present paper were ultimately bound. The determinant of a n×n full matrix with all different entries has n! terms; hence the determinant of a 14×14 such matrix has 87178291200 terms, which makes its very visualization impossible. In other words, large scale symbolic computations are unfeasible unless the matrices involved are very sparse.

Among symbolic computation languages we have referred to MACSYMA since it is the oldest, most widespread and most comprehensive package of this kind, but it is likely that similar results could be obtained with analogous products (such as MATHEMATICA) and perhaps with less computational costs, at least if the comparison table in Nagel (1990) can be taken as typical and not limited to the particular benchmarks used therein.

Table 7. Definition of algebraic operations between qualitons: a) addition; b) subtraction; c) multiplication; d) division, where a ? indicates an error condition.

+	P	N	Z	X
P	P	X	P	X
N	X	N	N	X
Z	P	N	Z	X
X	X	X	X	X

a)

—	P	N	Z	X
P	X	P	P	X
N	N	X	N	X
Z	P	N	Z	X
X	X	X	X	X

b)

x	P	N	Z	X
P	P	N	Z	X
N	N	P	Z	X
Z	Z	Z	Z	Z
X	X	X	Z	X

c)

/	P	N	Z	X
P	P	N	?	X
N	N	P	?	X
Z	Z	Z	?	Z
X	X	X	?	X

d)

Finally, we briefly sketch how the old friend FORTRAN could be used for tackling these kinds of problems. We first define qualitons as four pseudo-numbers labeled by P (for positive), N (for negative), Z (for zero), and X (for unknown) with which one can operate according to the four internal composition laws indicated in table 7 and called addition, subtraction, multiplication and division by analogy with arithmetics. It is easy to translate tables 7 a-d into FORTRAN subroutines. Algebraic expressions can then be constructed by successive application of the elementary operations. For example, saying that we know only the sign, but not the magnitude, of the entries of a matrix is equivalent to saying that the matrix entries are qualitons (different from X); then the determinant (as well as the cofactors) can be computed in a straightforward way by a recursive application of the well-known Laplace expansion theorems. We remark however that the application of this method is restricted to low-order systems because the computational complexity grows with the factorial of the order of the matrix (Friedberg, 1988). One could also hope to adapt faster numerical techniques to qualitons by just substituting elementary floating-point operations with elementary qualiton operations. This, however, is not always sufficient since often numerical algorithms make use not only of the algebraic structure of the real numbers but also of their order structure, whereas qualitons do not have a natural order structure (is X big or small?). This is the case, in particular, of the LU decomposition method as coded in Press et al. (1986, sec. 2.3) which could be used for computing determinants, but cannot be translated verbatim in terms of qualitons because of size comparisons between intermediate results. Moreover, contrarily to MACSYMA programs, qualitons are inadequate to resolve ambiguities through a synthesis procedure (cf. the discussion at the beginning of section 3 and Example 3.1).

The overall conclusion is then that symbolic languages are an effective and the most direct instrument for the qualitative modelling of (low-order or few-link) ecological communities.

Acknowledgements. Helpful discussions with professors E. Feoli and C.J. Puccia are gratefully acknowledged.

REFERENCES

BRIAND, F. and E. MC CAULEY. 1978. Cybernetic Mechanisms in Lake Plankton Systems: How to Control Indesirable Algae. Nature 237: 228-230.

DAY, R. and C.J. PUCCIA. 1988. Announcement. Coenoses 3: 148.

FRIEDBERG, S.H. 1988. A difference equation and operation counts in the computation of determinants. Mathematics Magazine 61: 295-297.

GANTMACHER, F.R. 1964. *The theory of matrices*. Vol. 2. Chelsea, Lewis, Michigan.

GATTO, M. and S. RINALDI. 1989. Acid deposition and catastrophes in forests: a tree-nutrient model. Coenoses 4: 81-90.

HIRSCH, M.W. and S. SMALE. 1974. *Differential Equations, Dynamical Systems and Linear Algebra*. Academic Press, New York.

HURWITZ, A. 1895. Uber die Bedingungen unter welchen eine Gelichung nur Wurzeln mit negativen reelen Theilen besitzt. Mathematische Annalen 46: 273-284.

NAGEL, Y. 1990. Review of PC MACSYMA. Notices of the American Mathematical Society 37: 11-14.

PAVELLE, R. and P.S. WANG. 1985. MACSYMA from F to G.J. of Symbolic Computation 1: 69-100.

PONTRYAGIN, L. 1962. Ordinary Differential Equations. Addison-Wesley, Reading, Massachusetts.

PRESS, W.H., B.P. FLANNERY, S.A. TEUKOLSKY and W.T. VETTERLING. 1986. *Numerical Recipes*. Cambridge University Press, Cambridge, Massachusetts.

PUCCIA, C.J. and R. LEVINS. 1985. *Qualitative Modeling of Complex Systems*. Harvard University Press, Cambridge, Massachusetts.

SVIREZHEV, YU.M. and D.O. LOGOFET. 1983. *Stability of Biological Communities*. Mir, Moscow.

Manuscript received: May 1990