MAPS : AN EXPERT ADVISOR FOR THE QUALITATIVE ANALYSIS OF DYNAMICAL SYSTEMS

M. Andretta (+), M. Lugaresi(++), P. Mazzanti(+++), R. Serra (+)

(+) TEMA, Viale A. Moro 38, Bologna (Italy)
(++) Dept. of Physics, Bologna University
(+++) Dept. of Phisics, Bologna University, INFN, Sezione di Bologna (Italy)

Abstract. The analysis of the evolution of non linear dynamical systems is a complex task. The cases where: i) the model equations can be regarded as a careful and reliable representation of the real system and, therefore, need no revision or modification; ii) the parameter values are precisely known; iii) the initial conditions are precisely known, are rather rare.

At least one of the previous conditions is not fulfilled in most of the systems of interest for System Dynamicists. Therefore qualitative analysis of dynamical systems, i.e. the study and classification of their asymptotic behaviours, is of extreme importance, at least in long term models.

The methodologies of knowledge representation recently developed in the field of expert systems can be applied to this problem. We therefore developed MAPS, an expert advisor for the qualitative analysis of dynamical systems. MAPS takes the system equations as input, classifies them according to their features and performs the necessary calculations at each stage, sending appropriate messages to the modelist.

At present MAPS deals with autonomous second order systems of ordinary differential equations.Further developments are foreseen concerning the study of higher order sytems and the design of an " equations database " for comparison with previously analyzed equations.

INTRODUCTION

When dealing with models of complex systems, we are faced with many uncertainties (Serra et al. (1986a, 1986b), Sedehi et al. (1986)). It often happens that the basic model equations are essentially empirical, lacking a firm theoretical basis. This is almost always the case, with the following two major exceptions :

*) those physical systems which allow a direct application of "natural laws " (e.g., planetary motion);

*) those socio-economical systems which are completely artificial (e.g., accounting) and can therefore be perfectly known, at least in principle (Bartezzaghi and Mariotti(1983)).

Besides this very fundamental kind of uncertainty, which involves the model structure, there are also well known problems relating to the parameter estimation and, in the case of dynamical models, to the determination of the initial conditions.

It is therefore necessary not to consider a single dynamical system and its time evolution, but rather a family of dynamical systems, which may differ with respect to model structure, parameters' values or initial conditions, or any combination of the previous sources of uncertainty.

These remarks are of course familiar to every system dynamist ; the usual recipe to deal with such a situation is to resort to several computer simulations, evaluating the robustness of the model to different modifications. It is a very difficult and complicated task, although it may prove a way to learn some interesting features of the system under study. We already pointed out the opportunity to use reduced models in order to compress the dimensionality of the phase space in such analysis (Sedehi et al. (1984,1986)).

The use of reduced models seems interesting when we are interested in the long term behaviour of the system, neglecting the fast dynamical variables which are "slaved" by the slower variables, also called order parameters (Haken (1977), Serra et al. (1986a)). The elimination of the fast variables amounts to a kind of projection of the original model equations on the subspace of the order parameters. However, a price needs to be payed in order to achieve this reduction of complexity : noise is introduced, leading to a stochastic reduced model. This, in turn, implies that the system description must be given in terms of partial differential equations of the Fokker-Planck type. However, there exist several cases where the level of noise is low enough to allow the use of deterministic reduced models, without introducing appreciable errors.

A simple example is the following (Haken (1977), Serra et al. (1986a) : consider a stochastic differential equation of the following type :

$$\dot{x}_i = f_i(x) + A_i(t) \qquad i = 1, \dots M \qquad (1)$$

where A(t) is gaussian white noise. We also suppose for simplicity that the system is of the gradient type, i.e. :

$$f_i(x) = -\frac{\partial V(x)}{\partial x_i} \qquad (2)$$

The "solution" of the Langevin equation (Eq.1) involves the determination of the time evolution of the probability density of the order parameters, p(x,t). There exist standard recipes in order to determine the evolution equation for p(x,t), which is a partial differential equation named after Fokker and Planck. Its asymptotic solution can be given in analytical terms in the case of gradient systems (of course, that's why we have chosen this example) :

$$P_{\infty}(x) = Q e^{-V(x)/D} \quad (3)$$

$$\dot{x}_i = f_i(x) \quad (4)$$

where Q is a normalization constant and D a measure of the noise intensity. We can directly verify on Eq.(3) that the extrema of the asymptotic probability density are extrema also of the potential function V(x), and are therefore stationary solutions of the deterministic dynamical system (Eq.4). In particular, stable equilibrium points of the deterministic system (i.e., minima of V(x)) correspond to maxima of the asymptotic probability density, and their determination allows a semi-quantitative analysis of the stochastic system (Eq.1). Indeed, if the noise level is low, the stochastic system will be found almost always near the extrema of the probability density.

This conclusion applies also to a wider class of systems. We therefore conclude that, in the case of sufficiently low noise level, a deterministic analysis is sufficient to get a picture of the system's destiny, precise enough for most practical purposes, provided that we remember that only stable equilibrium points should be considered, and that transitions among different locally stable states are allowed (although their rate may be also very low).

We also remark that, qualitatively, the basic content of the well-known central manifold theorem in dynamical systems theory (Guckenheimer and Holmes (1983)) is that the asymptotic behaviour of a deterministic system of ordinary differential equations can be inferred from that of a lower dimensional one.

We are thus lead to the conclusion that the analysis of deterministic reduced models may be very informative. In particular, we are interested in a discussion of the different asymptotic behaviours and their stability properties ("qualitative" analysis of dynamical systems). A fortiori, we are interested in such analysis when low dimensional deterministic systems stem directly from our modelling hypotheses, without being the result of the application of a projection operator to a more detailed model.

THE GOALS OF MAPS

Some analytical results are available, concerning the analysis we are interested in. The usefulness of the analytical approach in dealing with uncertain systems cannot be overemphasized, since it allows us to avoid several computer trials whose interpretation may be very cumbersome. Such analytical results are particularily rich in the case of low dimensional systems (Guckenheimer and Holmes (1983), Serra et al. (1986a, 1986b)). The two dimensional case is particularily simple, since chaotic behaviour is excluded, while it may be observed in higher dimensional systems. Usually, the "qualitative" analysis of dynamical systems requires an interplay between analytical and numerical computation. While several routines can be found in the numeric domain, the analytical calculations are usually carried on "by hand".

The main idea behind the system MAPS (Mathematical Advising Production System) is that a meaningful subset of these analytical calculations can be carried on in an authomatic way, due to progresses in computer algebra systems and expert systems.

There exist presently several systems which are capable of analytical calculations. Among the general purpose systems the most famous is doubtless Macsyma (Buchberger et al. (1982)). It has many symbolic computing capabilities both to perform basic algebraic and analytical calculations (e.g. expressions semplifications and differentiation, computation of limits, definite and indefinite integrals, functions expansion in Taylor or Laurent series, analytical solutions of algebraic and differential equations) and to solve problems in some applied mathematics areas (e.g. general relativity, high energy physics). Since its first implementation, in 1972, Macsyma has been used in many areas, to explore problems, for example, in atomic scattering cross sections, antenna theory, maximum likehood estimation, economics.

Another powerful system is Scratchpad, originally implemeted by I.B.M. using an experimental System/360 LISP system at the begining of the seventies. Scratchpad II (Jenks (1985)) is presently used only within I.B.M.'s scientific centers. Despite its poor diffusion, Scratchpad II deserves to be mentioned because of the extensible language approach adopted in its design that can be considered a sort of mathematical data and operators abstraction. The user language, in fact, contains a set of basic syntactic constructors, described by notations similar to those in conventional mathematics. These basic constructs may be extended by the user, so that Scratchpad II may be considered a very-high-level non procedural language for mathematical manipulations.

Besides the general purpose systems, there are also many systems dedicated to the solution of specific problems . Applications of these systems include celestial mechanics (e.g. CAMAL and TRIGMAN), quantum electrodynamics (e.g. REDUCE and SCHOONSCHIP), general relativity (e.g. CAMAL and SHEEP), high energy physics (e.g. SCHOONSCHIP and ASHMEDAI), optics (e.g CAMAL), chemistry (e.g. FORMAC and PERTRAN) and electronics (e.g. REDUCE). Some of these systems have grown out to general purpose ones (van Hulzen and Calmet (1982), Bordoni and Miola (1985)).

The MAPS system has been implemented with the aid of another computer algebra system, muMath, developed by D.Stoutmeyer and co-workers (Stoutmeyer (1985)). This choice was due to the fact that the muMath package was designed for personal computers, while retaining several powerful features. Mumath is written in muSimp, a high level Structured IMPlementation language, especially designed for artificial intelligence applications on personal computers. It was designed at the end of the seventies by Stoutmeyer and Rich, in order to give the user the power of a functional language like Lisp but with a much easier syntax.

On the other hand, several progresses have been achieved in expert systems (Harmon and King (1985)). They can be seen as systems performing in a way similar to a human expert, in narrow domains. A major concept relating to this field is that of heuristic knowledge (Nilsson (1982), Winston (1984)), namely that kind of knowledge which is embedded in rules of thumb, mental habits, etc., rather than in books and formal teaching. The major reason for the use of such heuristic knowledge is the need to reduce large search spaces, "trying first" those solutions which can be regarded as most probable, although by no means sure, on the basis of previous experience. A large body of techniques aiming at formalizing these rules through direct interaction with experts is known as "knowledge engineering".

The qualitative analysis of dynamical systems usually involves a large part of formal mathematical knowledge, plus some heuristics: here too we have a large search space, where "rules of thumb" can be applied in order to keep the problem manageable.

We can so summarize the goals of MAPS : to assist the modelist in the qualitative analysis of deterministic dynamical systems, performing a parametric analysis of the nature and stability of the asymptotic states. The analysis cannot be made completely authomatic, but it needs an interaction between the system and the modelist. The system should incorporate not only some mathematical knowledge, but also some "knowledge of the mathematician",i.e. heuristics. MAPS is a mixed system, involving both analytical and numerical computations.

THE MAPS PROTOTYPE

There exists presently a first prototype of MAPS for PC IBM compatibles, which is already a useful tool in the qualitative analysis of dynamical systems and which has also proved useful in order to analyze the requirements for a more complete system. In its present version, the system supports the qualitative analysis of two-dimensional autonomous systems of the following kind :

$$\begin{cases} \dot{x} = f(x, y, p, \vec{c}) \\ \dot{y} = g(x, y, p, \vec{c}) \end{cases} \qquad (5)$$

Here x and y are the two dynamical variables, c denotes a set of constants (literal and/or numerical) and p denotes a scalar parameter : the asymptotic analysis is performed as a function of the values of this parameter. The analysis is performed within a domain of values of x,y, and p which is asked by the system. Further specifications include the sign of the literal constants.

The system performs a diagnosis on the proposed system, verifying whether it belongs to some peculiar class (e.g., linear, hamiltonian or gradient) in order to simplify the analysis. This diagnosis is performed basing upon the (symbolic) Jacobian matrix of the system, which will be useful also in the following, when stability of equilibria will be checked. The next step is the search for equilibrium points, i.e. solutions of the algebraic system :

$$\begin{cases} f(x, y, p, \vec{c}) = 0 \\ g(x, y, p, \vec{c}) = 0 \end{cases} \qquad (6)$$

These solutions are searched in order of increasing difficulty. First the existence of solutions like (0,0),(x,0),(0,y) is checked. Such solutions can eventually be used in order to reduce the complexity of the algebraic system through factorization, trying to reduce the order of, say, f by expressing it as

$$f(x, y) = (x - x_0)^a \, (y - y_0)^b \, f^*(x, y) \qquad (7)$$

where a or b may eventually vanish. Moreover, it is possible to look for solutions by direct substitution.

For every equlibrium point, a linear stability analysis is performed, as a function of the parameter p. This involves the determination of the nature (real or complex) of the eigenvalues of the Jacobian matrix, evaluated at the equlibrium point, and of the sign of the real eigenvalues and of the real part of the complex eigenvalues. The nature of the equilibrium points can be determined according to the information summarized in Table 1.

EIGENVALUES	CONFIGURATION
Real distinct positiv	Unstable node
Real distinct negative	Stable node
Real distinct opposite signs	Saddle
Real coincid. negative	Stable "star" node
Real coincid. positive	Unstable "star" node
Compl. conjug. zero real part	Center
Compl. conjug. neg. real. part	Stable focus
Compl. conjug. pos. real part	Unstable focus
1 Eigenvalue = 0	Degenerate point

TABLE 1. Classification of the equilibrium points.

There is also the possibility to perform numerical simulations, in order to complement the analytical results. In fact, MAPS includes an interface with a set of numerical integration routines ; presently, the definition of the parameter values to test is entirely left to the user.

SYSTEM STRUCTURE

The system is written in muSIMP, i.e. the host language of muMath. This choice obviously simplifies the interface between MAPS and muMath. Moreover, a functional language like muSimp is particularily well suited for the kind of problems we are dealing with. As far as hardware is concerned, an IBM PC compatible with at least 512 kbytes of RAM memory is required.

MAPS is organized as a production system (Nilsson (1982),Winston (1984), Harmon and King (1985)), i.e. a system composed by a global database, a set of "if...then..." production rules, a control system and a knowledge acquisition module. It presently incorporates about 80 rules. The antecedents of the rules have the following form:

if < Left Hand Side > = < Right Hand Side >

the consequent being always a muSIMP function. Each rule is triggered by a pattern matching mechanism between the two sides of its antecedents and the database. A rule may have two kinds of antecedents: askable and derivable. When an askable antecedent is met, and there are no informations corresponding to its LHS in the database, the control system of MAPS asks the user for additional information. On the contrary, the control system automatically computes the missing information corresponding to the LHS of a derivable antecedent. When all the antecedents of a rule are satisfied, the muSIMP function of the consequent may be applied.

One of the major problems in building a production system is the so called conflict set resolution, that is the problem of singling the proper rule to be applied out of those which might be activated at each step of the resolution process. We have tackled this problem organizing the rules into a hierarchical structure of contexts ("contesti"). A context is a subset of rules, which are involved in the solution of some specific subproblem; in every context the rules are ordered according to their priority. There is only one active context at a time, and the control system of MAPS analyzes the rules in order of decreasing priority. A special context, called directing context, controls the activation of the other contexts, depending on the status of the database. Figure 1 sketches the hierarchical structure of the contexts.

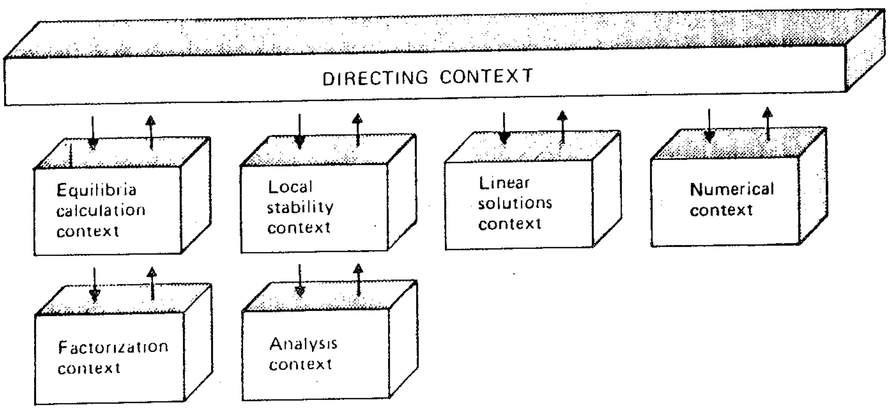

Fig. 1. The hierarchical structure of the contexts.

AN APPLICATION OF MAPS

Let us now describe an application of MAPS, in order to show the way how it works : we will perform the qualitative analysis of the equilibrium points of the following system, which can be regarded as a modified version of the well known Lotka-Volterra equations :

$$\begin{cases} \dot{x} = -ax^2 + pxy \\ \dot{y} = by - pcxy \end{cases} \qquad (8)$$

The corresponding output is shown in Appendix I. At present, the output of MAPS is written in Italian; anyway, a new English version is going to be developed. At first, the system asks the user the equations to be analyzed, the allowed domains of values of x, y and p and the signs of the constants. Then it computes the symbolic Jacobian matrix of the system and automatically analyzes the equations, verifying if they belong to some particular class (in this example the equations are polinomials, not linear ....). If the system is linear, the program algorithmically calculates the equilibrium point. If, as in this case, the system is not linear, MAPS searches for the equilibrium points using the above mentioned heuristics strategy. The program now displays the solutions it has found with this method and asks the user if he wants to perform the factorization of the equations. In the example we report MAPS factorizes the equations and it finds another equilibrium point. Now, the first part of the analysis is completed and a swap of the working memory is required. This is necessary because of the limited amount of RAM (320 kilobytes) muSIMP can address. When the loading of the second part of MAPS is done, the program performs a linear stability analysis for every equilibrium point, as a function of the parameter p. The final results of the analysis are displayed and the program asks if the user wants to activate the numeric context.

The reasons why we have introduced a numerical interface in the system are the following: 1) impossibiliy of getting exact results due to intrinsic complexity of the differential system. This complexity may reflect itself into an excessive computer overhead (excessive computing time or memory occupation). 2) The possibility of getting numerical results for various inital conditions and different values of the various parameters is an useful feature, also in the case when the main goal of the MAPS system succeeds. The numerical interface consists of a number of programs written in Basic which perform the following operations:

a) translation into Basic of the representation, in terms of lists, of the system of differential equations. Constant names and the parameter p are stored together with their ranges. The correspondence between the matematical functions in muSIMP and those in Basic is exact.

b) Explicit construction of the subprogram (in Basic) containinig the system to be integrated.

c) Numerical calculation by using suitable integration routines. The values chosen for the initial conditions, the literal constants and the parameter p are carefully controlled against the previously defined associated ranges. It is also possible to graphically represent the results obtained on the screen (high or low resolution) or to print them or both.

CONCLUSIONS

We have presented the first prototype of an expert advisor for the qualitative analysis of dynamical systems. This system integrates analytical, numerical and knowledge representation environments to study and classify the equilibrium points of a second order system of ordinary differential equations. The first benchmarks are positive. The three enviroments appear to be well integrated and the equilibrium points analysis is feasible also for relatively complex systems.

Some limitations arise from the characteristics of the functional programming environment and of the computer algebra system. In particular, the muSIMP impossibility to address more than 320 kbytes compelled us both to adopt a simple (not memory expensive) resolution startegy, and to segment the program into two parts. The Algol-like output of muMATH and the absence of general functions for the simplification of mathematical expressions, make the output from MAPS sometimes difficult to read. Anyway, these problems will be at least partially overcome with the new, already announced version of muSIMP muMATH.

Among the most important extensions of the system which we envisage, provided that enough memory is available (either on PCs or more powerful computers), we mention the diagnosis of possible limit cycles, the analysis of higher order systems and the development of an "equations database", with an efficient search strategy, for the comparison between already studied cases and new ones.

ACKNOWLEDGEMENTS

We gratefully acknowledge many helpful discussions with L.Gardini and G.Zanarini and the financial support of Enidata (grant N. 021ED86)

REFERENCES

Bartezzaghi,E. and Mariotti,S. (1983). Modelli per le decisioni strategiche aziendali. Angeli (Milano).

Bordoni,L. and Miola,A. (1985). Applicazione della manipolazione algebrica e simbolica. Rivista di informatica, 15, pp.143-166.

Buchberger,B., Collins,G.E., Loos,R. eds. (1982). Computer algebra; symbolic and algebraic computation. Springer, Wien.

Guckenheimer,J. and Holmes,P. (1983). Nonlinear oscillations, dynamical systems and bifurcations of vector fields. Springer, Berlin.

Haken,H. (1977). Synergetics. Springer, Berlin.

Harmon,P. and King,D (1985). Expert systems. Wiley, New York.

Jenks,R.D. (1985). A brief introduction to Scratchpad II. The Scratchpad II Newsletter, 1, p.1

Nilsson,N.J. (1982). Principles of artificial intelligence. Springer, Berlin.

Sedehi,H.A., Serra,R., Vassallo,S. (1984). Theoretical approach to long term company behaviour . Proc. Intl. Conf. System Dynamics, Oslo, p.351

Sedehi,H.A., Serra,R., Vassallo,S. (1986) . Adiabatic elimination of fast relaxing variables in socio-economic dynamical models. Eur. J. Oper. Res., 25, p.1

Serra,R., Zanarini,G., Andretta,M., Compiani,M., (1986a). Introduction to the physics of complex systems. Pergamon, Oxford.

Serra,R. and Zanarini,G. (1986b). Tra ordine e caos. CLUEB, Bologna.

Stoutmeyer,D.R. (1985). A preview of the next IBM-PC version of muMath. In B.Buchberger ed., Eurocal 85, Springer, Berlin, pp.33-44

van Hulzen,J.A. and Calmet,J. (1982). Computer algebra systems. In B.Buchberger et al. eds., Computer algebra,symbolic and algebraic computation, Springer, Wien, pp.221-244

Winston,P.H. (1984). Artificial intelligence. Addison-Wesley, Reading.

APPENDIX I . AN ANALYSIS WITH MAPS:THE LOTKA-VOLTERRA SYSTEM

```
? MAPS ()&

--- Mathematical Advising Production System (MAPS).   Rel. 1.1 ---

 INSERISCI LA PRIMA EQUAZIONE DEL SISTEMA
 EQ1 : DX/DT == < espr >_

1? EQ1 : DX/DT == A X - P X Y _

 INSERISCI LA SECONDA EQUAZIONE DEL SISTEMA
 EQ2 : DY/DT == < espr >_

1? EQ2 : DY/DT == -B Y^2 + C P Y X _

 INSERISCI IL PRIMO ESTREMO DEL RANGE DELLA VARIABILE X :
 ES1 : < estrinf >_

1? ES1:0_

 INSERISCI IL SECONDO ESTREMO DEL RANGE DELLA VARIABILE X :
 ES2 : < estrsup >_

1? _

 INSERISCI IL PRIMO ESTREMO DEL RANGE DELLA VARIABILE Y :
 ES1 : < estrinf >_

1? ES1:0_

 INSERISCI IL SECONDO ESTREMO DEL RANGE DELLA VARIABILE Y :
 ES2 : < estrsup >_

1? _

 INSERISCI IL PRIMO ESTREMO DEL RANGE DEL PARAMETRO P :
 ES1 : < estrinf >_

1? ES1:0_
```

```
INSERISCI IL SECONDO ESTREMO DEL RANGE DEL PARAMETRO P :

ES2 : < estrsup >_

1? _


SPECIFICA IL SEGNO DELLE COSTANTI
> 0 ;   or   < 0 ;   or   I ;   (-Indeterminato-)

A >0;

B >0;

C >0;


STO CALCOLANDO LO JACOBIANO DEL SISTEMA

STO VERIFICANDO SE L'ORIGINE E' UN PUNTO DI EQUILIBRIO

RIDUZIONE DEL SISTEMA ALL'ASSE X

RIDUZIONE DEL SISTEMA ALL'ASSE Y


I PUNTI DI EQUILIBRIO FINORA CALCOLATI SONO:

( 0 , 0 )

VUOI PROVARE A FATTORIZZARE LE EQUAZIONI

DX /DT == A X - Y P X

DY /DT == -B Y^2 + C Y P X

( S/N ;)

S;

STO PROVANDO A FATTORIZZARE LE EQUAZIONI

STO CALCOLANDO I PUNTI DI EQUILIBRIO DEL SISTEMA LINEARE


I PUNTI DI EQUILIBRIO CALCOLATI SONO :

( 0 , 0 )

( A B/(C P^2) , A/P )

I PUNTI DI EQUILIBRIO INTERNI AI RANGES PREFISSATI SONO:

( A B/(C P^2) , A/P )

( 0 , 0 )

LA FASE DI DETERMINAZIONE DEI PUNTI DI EQUILIBRIO
```

```
 E' TERMINATA

 SELEZIONA IL DRIVE PER IL SALVATAGGIO MOMENTANEO
 DELLA BASE DEI FATTI    (A / B / C / D)  C


 VIENE ESEGUITA LA FASE DI SALVATAGGIO

 AL TERMINE BATTERE : CONTINUE ( < drive > )&


? CONTINUE (C)&

@: FALSE
?@: BASEFATTI
?


 STO CALCOLANDO GLI AUTOVALORI DEL SISTEMA RELATIVI AL PUNTO DI EQUILIBRIO :

 ( A B/(C F^2) , A/F )

 GLI AUTOVALORI SONO:

-A B/(2 F) + A B^(1/2) (B - 4 F)^(1/2)/(2 F)

-A B/(2 F) - A B^(1/2) (B - 4 F)^(1/2)/(2 F)


 SE E' VERA LA RELAZIONE

(B - 4 F)/F^2 > 0

 IL SISTEMA AMMETTE DUE AUTOVALORI REALI DISTINTI :

-A B/(2 F) + A B^(1/2) (B - 4 F)^(1/2)/(2 F)
-A B/(2 F) - A B^(1/2) (B - 4 F)^(1/2)/(2 F)

 ESSI SONO ENTRAMBI DI SEGNO < 0.
 PERTANTO, IL PUNTO DI EQUILIBRIO :

 ( A B/(C F^2) , A/F )

 E' UN NODO STABILE


 STO CALCOLANDO GLI AUTOVALORI DEL SISTEMA RELATIVI AL PUNTO DI EQUILIBRIO :

 ( 0 , 0 )

 GLI AUTOVALORI SONO:

A

0
```

```
PER OGNI VALORE DI P INTERNO AL RANGE PREFISSATO

(0 PINF)

IL SISTEMA AMMETTE DUE AUTOVALORI REALI DISTINTI :

A
0

UNO DEI DUE AUTOVALORI E' UGUALE A ZERO, MENTRE L'ALTRO E' > 0

PERTANTO, IL PUNTO DI EQUILIBRIO : (0 0)

E' UN PUNTO DEGENERE DI TIPO INSTABILE

VUOI PASSARE AL CONTESTO NUMERICO ?  (S;/N;)

S;

VUOI CONTINUARE ? (S/N) N
@: TRUE
```