Go Back

A Method For Finding Equilibrium Points of a Non-Linear Dynamic Model

Yaman Barlas and Hakan Yaşarcan
Boğaziçi University, Dept. of Industrial Eng.
80815 Bebek, İstanbul - Turkey
Tel: ++(90) 212 263 15 40/1407-1408, Fax: ++(90) 212 265 18 00
ybarlas@boun.edu.tr, yasarcan@boun.edu.tr

An algorithm for finding all the equilibrium points of a given non-linear dynamic model is proposed.. Such an algorithm would necessitate the general solution of a set of N non-linear algebraic equations. It is well known that no such method exists in general. Our method aims to work for a rather general subset of non-linear systems, namely when all non-linearities are expressed in polynomial terms. The significance of the method is that; i- it can greatly speed up model analysis by providing the equilibrium information prior to simulation, and ii- it can help verify the results obtained from simulations (numerical simulation may skip an existing equilibrium and "create" spurious equilibria). The method is explained and demonstrated on two examples. The algorithm works well, except when there are infinite number of equilibria on an N-dimensional plane. Current work focuses on this sub-problem. Finally, there are some issues of speed and numerical accuracy, the other two main topics of current and future research.

Key words: equilibrium point, analysis, dynamic, non-linear model.

INTRODUCTION

Given the following set of dynamic equations:

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{Bmatrix} = \begin{Bmatrix} f_{1,1}(x_1, x_2, \ldots, x_n) \\ f_{1,2}(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_{1,n}(x_1, x_2, \ldots, x_n) \end{Bmatrix} \qquad \text{(equation set 1)}$$

$$\left(\textit{Note that} \quad \dot{x}_i = \frac{dx_i}{dt} \right)$$

it is a well known fact that there is no general method to solve this dynamical system of equations when $f_{1,i}$'s are non-linear. Furthermore there is no general method to find even the equilibrium points of such a non-linear system. (Equilibrium points are constant solutions of (1), such that if x_i start on them, they stay on them forever. i.e. $\dot{x}_i(t) = f_{1,i}(x_1, x_2, ..., x_n) = 0$ for all i and $t > 0$). The non-linear problem is well summarized by Press and Flannery; "We can make an extreme, but wholly defensible statement: There is no good general method for solving systems of more than one nonlinear equations. Furthermore it is not hard to see why (very likely) there never will be any good general methods..." (Press and Flannery, 1986).

Although system dynamics literature acknowledges the importance of equilibria and stability information (Aracil and Toro, 1988. Barlas and Çivi 1994), there is not much concrete research done on the topic. In more general literature there exist some numerical methods for solving non-linear systems of algebraic equations (Woodford, 1992). These methods need an initial vector and starting with this vector they gradually converge to a solution by some search techniques. But, these methods cannot tell how many equilibrium solutions exist. So with trial and error, with different initial vectors, the procedure must be repeated. At the end, there is no knowledge as to what whether all possible equilibria are covered or not. This is similar to running a simulation program with different initial values to find the equilibrium points. There are also some analytic/symbolic methods that work for certain restricted, polynomial non-linearities. (Barlas and Çivi 1994. Jenner, 1963. Rayna, 1987). The problem is that these methods face too many implementation problems (numeric or symbolic) when applied to realistic models of even moderate size.

The method that we propose is not a numerical method. It is essentially an analytical and symbolic method, supported with some numerical sub-procedures when needed. The main difference of our method and the ones in the literature is that our method does not need an initial vector and it aims to find all the solutions of the given system of equations. In this research we hope to develop a method that will find all equilibrium points of n dimensional system of non-linear dynamic models for a very wide range of cases. The only assumption made is that $f_{1,i}$'s can be any polynomial and any combination of polynomials using the four basic operators (addition, subtraction, multiplication and division) for all *i*.

THE MAIN STEPS OF THE PROPOSED METHOD

Step 1. Set all the equations in *equation set1* equal to zero.

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{Bmatrix} = \begin{Bmatrix} f_{1,1}(x_1, x_2, \ldots, x_n) \\ f_{1,2}(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_{1,n}(x_1, x_2, \ldots, x_n) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix}$$

Step 2. Get rid of the division operator in above equation set with necessary multiplications by denominators. [See Appendix-2 for a technical note].

$$\begin{Bmatrix} f_{2,1}(x_1, x_2, \ldots, x_n) \\ f_{2,2}(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_{2,n}(x_1, x_2, \ldots, x_n) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix} \qquad \text{(equation set 2)}$$

Step 3. Derive one-dimensional polynomials in terms of each variable by a "reduction technique" applied to the above system of *n* equations. Since our reduction technique cannot work with division operators, the *equation set 2* will be used. At the end, the reduction technique will produce distinct one-dimensional polynomials for each of the n variables.

Steps of The Reduction Algorithm

Step 3.1. Select a variable that is going to be kept and call it x_k. The resulting polynomial will be only in terms of x_k. The initial equation set consists of $\{f_{2,1}, f_{2,2}, \ldots, f_{2,n}\}$, which will be updated by the "elimination procedure".

Step 3.2. Select a variable x_i to be eliminated such that $i \neq k$. Eliminate x_i from the equations, using the "elimination procedure" [see Appendix-1]. As a result of this procedure, the number of the equations in the equation set is also reduced by one.

Step 3.3. If the equation set is reduced to a single polynomial in terms of x_k only, then end the reduction algorithm, otherwise go back to *step 3.2* to select another variable to eliminate.

The above *reduction algorithm* is applied for *each* of the variables to obtain the following polynomials:

$$\begin{Bmatrix} f_{3,1}(x_1) \\ f_{3,2}(x_2) \\ \vdots \\ f_{3,n}(x_n) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix} \qquad \text{(equation set 3)}$$

Step 4. Find the real roots of each equation in *set 3*. In the literature there are different methods to find the roots of a given one-dimensional polynomial (Santina and D'Carpio, 1991).

$$\begin{Bmatrix} rootset_1 \\ rootset_2 \\ \vdots \\ rootset_n \end{Bmatrix} \equiv \begin{Bmatrix} \left\{ \overset{*}{X}_{1,1}, \overset{*}{X}_{1,2}, \ldots, \overset{*}{X}_{1,m_1} \right\} \\ \left\{ \overset{*}{X}_{2,1}, \overset{*}{X}_{2,2}, \ldots, \overset{*}{X}_{2,m_2} \right\} \\ \vdots \\ \left\{ \overset{*}{X}_{n,1}, \overset{*}{X}_{n,2}, \ldots, \overset{*}{X}_{n,m_n} \right\} \end{Bmatrix}$$

Step 5. After *step 4*, there are $m_1 * m_2 * \cdots * m_n$ potential equilibrium points. But not all combinations of roots make all the equations in *equation set 1* equal to zero, so they must be tested. If a root combination makes all the derivatives in *equation set 1* zero, then it is an equilibrium point of this system. All the equilibrium points are thus determined.

EXAMPLE 1

Assume the following non-linear dynamic model of order three:

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{Bmatrix} x_1^2 - x_2^2 - x_1 x_3 \\ \dfrac{x_2^2 - 2x_1 x_2}{x_3} - 3x_2 - x_3 \\ \dfrac{x_3}{2x_2 - x_1 + 1} + 2 \end{Bmatrix} \qquad \text{(equation set 1)}$$

Step 1. Set all the equations in *equation set 1* to zero.

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{Bmatrix} x_1^2 - x_2^2 - x_1 x_3 \\ \dfrac{x_2^2 - 2x_1 x_2}{x_3} - 3x_2 - x_3 \\ \dfrac{x_3}{2x_2 - x_1 + 1} + 2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$$

Step 2. Get rid of the division operator in the above equation set with necessary multiplications by denominators. [Appendix-2]

$$\begin{Bmatrix} x_1^2 - x_2^2 - x_1 x_3 \\ x_2^2 - 2x_1 x_2 - 3x_2 x_3 - x_3^2 \\ x_3 + 4x_2 - 2x_1 + 2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \qquad \text{(equation set 2)}$$

Step 3. Find one-dimensional polynomials for each variable by the reduction technique:

The Reduction Algorithm (Iteration #1)

Step 3.1. Select x_1 as the variable that is going to be kept. The initial equation set consists of $\{f_{2,1},\ f_{2,2},\ f_{2,3}\}$.

Step 3.2. Select variable x_2 to be eliminated. (We do not currently have a rule for variable selection for elimination, so we select the variable with lower index). Eliminate x_2 using the elimination procedure:

Elimination Procedure

Step 3.2.1. The maximum powers of x_2 in each equation in the equation set 2 are 2, 2 and 1 respectively.

Step 3.2.2. Among these maximum powers, 1 is the minimum, so the last equation ($f_{2,3}$) is selected.

Step 3.2.3. From $f_{2,3}$, $x_2 = 0.5x_1 - 0.25x_3 - 0.5$ is obtained.

Step 3.2.4. Insert x_2 in the other two equations of equation set.

$$\begin{Bmatrix} x_1^2 - (0.5x_1 - 0.25x_3 - 0.5)^2 - x_1x_3 \\ (0.5x_1 - 0.25x_3 - 0.5)^2 - 2x_1(0.5x_1 - 0.25x_3 - 0.5) - 3(0.5x_1 - 0.25x_3 - 0.5)x_3 - x_3^2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

Do necessary simplifications and multiplications to get rid of the division operator.

$$\begin{Bmatrix} -12x_1^2 - x_3^2 - 12x_1x_3 - 4x_3 + 8x_1 - 4 \\ -3x_3^2 - 20x_1x_3 + 28x_3 - 12x_1^2 + 8x_1 + 4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \qquad \text{(equation set 2.1)}$$

Step 3.2.5. Since variable x_2 is eliminated completely, the new equation set is the above equation set and this is the end of the elimination procedure.

Step 3.3. There is more than one variable left in 2.1, so go back to *step 3.2* to select another variable to eliminate.

Step 3.2. (Second time in *Iteration #1*) This time eliminate x_3 using the elimination procedure:

Elimination Procedure

Step 3.2.1. The maximum powers of x_3 in each equation in the *equation set 2.1* are 2 and 2 respectively.

Step 3.2.2. The maximum powers are equal, so we break the tie arbitrarily and select the equation with smaller index, i.e. the first equation in the *equation set 2.1*.

Step 3.2.3. From $f_{2.1,1}$, $x_3^2 = -12x_1^2 - 12x_1x_3 - 4x_3 + 8x_1 - 4$ is obtained.

Step 3.2.4. Insert x_3^2 in the second equation of *equation set 2.1*, yielding

$$\{-3(-12x_1^2 - 12x_1x_3 - 4x_3 + 8x_1 - 4) - 20x_1x_3 + 28x_3 - 12x_1^2 + 8x_1 + 4\} = \{0\}$$

and do necessary simplifications to obtain the following:

$$\{6x_1^2 - 2x_1x_3 - 5x_3 + 2x_1 - 2\} = \{0\}$$

Step 3.2.5. Variable x_3 is not eliminated completely, so a new equation set is formed with the first equation of the *equation set 2.1* and the above equation. The elimination procedure is thus repeated on the following equation set.

$$\begin{Bmatrix} -12x_1^2 - x_3^2 - 12x_1x_3 - 4x_3 + 8x_1 - 4 \\ 6x_1^2 - 2x_1x_3 - 5x_3 + 2x_1 - 2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \qquad \text{(equation set 2.2)}$$

Step 3.2.1. (Second time) The maximum powers of x_3 in each equation in the equation set 2.2 are 2 and 1 respectively.

Step 3.2.2. Among these maximum powers, 1 is the minimum, so the last equation is selected.

Step 3.2.3. From $f_{2.2,2}$, $x_3 = \dfrac{6x_1^2 + 2x_1 - 2}{2x_1 + 5}$ is obtained.

Step 3.2.4. Insert x_3 to the first equation in *equation set 2.2*, yielding

$$\left\{-12x_1^2 - \left(\frac{6x_1^2 + 2x_1 - 2}{2x_1 + 5}\right)^2 - (12x_1 + 4) * \left(\frac{6x_1^2 + 2x_1 - 2}{2x_1 + 5}\right) + 8x_1 - 4\right\} = \{0\}$$

and do necessary simplifications to obtain the following:

$$\{33x_1^4 + 52x_1^3 - 64x_1^2 - 56x_1 + 16\} = \{0\} \qquad \text{(equation set 2.3)}$$

Step 3.2.5. Since variable x_3 is eliminated completely, the above is the new equation set and this is also the end of the elimination procedure.

Step 3.3. End the reduction algorithm since there is just one equation left in the equation set and in terms of x_1 only. This equation *2.3* will be the first equation of *equation set 3*. (See below, the very end of this example).

The Reduction Algorithm (Iteration #2)

Step 3.1. Select x_2 as the variable that is going to be kept. The initial equation set consists of $\{f_{2,1}, f_{2,2}, f_{2,3}\}$.

Step 3.2. Select variable x_1 to be eliminated. Eliminate x_1 with the elimination procedure:

Elimination Procedure

Step 3.2.1. The maximum powers of x_1 in each equation in the *equation set 2* are 2, 1 and 1 respectively.

Step 3.2.2. Among these maximum powers, 1 is the minimum, in both the second and third equations. Arbitrarily break the tie and select the second equation that has smaller index.

Step 3.2.3. From $f_{2,2}$, $x_1 = \dfrac{x_2^2 - 3x_2x_3 - x_3^2}{2x_2}$ is obtained.

Step 3.2.4. Insert x_1 in the other equations in the equation set, and do necessary simplifications and multiplications to get rid of the division operator.

$$\begin{Bmatrix} 3x_2^4 + 8x_2^3x_3 - 13x_2^2x_3^2 - 8x_2x_3^3 - x_3^4 \\ 8x_2x_3 + x_2^2 + 2x_3^2 + 4x_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \qquad \text{(equation set 2.4)}$$

Step 3.2.5. Since variable x_1 is eliminated completely, the new equation set is the above set 2.4 and this is the end of the elimination procedure.

Step 3.3. Since there is more than one variable in *equation set 2.4* go back to *step 3.2* to select another variable to eliminate.

Step 3.2. (Second time for *Iteration #2*) This time eliminate x_3 using the elimination procedure.

Elimination Procedure

Step 3.2.1. The maximum powers of x_3 in each equation in the *equation set 2.4* are 4 and 2 respectively.

Step 3.2.2. Among these maximum powers, 2 is the minimum, so the last equation is selected.

Step 3.2.3. From $f_{2.4,2}$, $x_3^2 = -4x_2x_3 - 3x_2^2 - 2x_2$ is obtained.

Step 3.2.4. Insert x_3^2 in the first equation of the *equation set 2.4*, and do necessary simplifications

$$\begin{Bmatrix} 3x_2^4 + 8x_2^3x_3 + 52x_2^3x_3 + 39x_2^4 + 26x_2^3 + 32x_2^2x_3^2 + 24x_2^3x_3 + 16x_2^2x_3 \\ -\left(16x_2^2x_3^2 + 9x_2^4 + 24x_2^3x_3 + 16x_2^2x_3 + 12x_2^3 + 4x_2^2\right) \end{Bmatrix} = \{0\}$$

after simplifying the above equations we obtain the following, which again includes an x_3^2 term:

$$\left\{33x_2^4 + 60x_2^3x_3 + 14x_2^3 + 16x_2^2x_3^2 - 4x_2^2\right\} = \{0\}$$

According to the elimination procedure, x_3^2 must again be inserted in the above equation, yielding:

$$\left\{33x_2^4 + 60x_2^3x_3 + 14x_2^3 + 16x_2^2\left(-4x_2x_3 - 3x_2^2 - 2x_2\right) - 4x_2^2\right\} = \{0\}$$

again simplifications are made:

$$\left\{33x_2^4 + 60x_2^3x_3 + 14x_2^3 - 64x_2^3x_3 - 48x_2^4 - 32x_2^3 - 4x_2^2\right\} = \{0\}$$

and we obtain the following:

$$\left\{15x_2^4 + 4x_2^3x_3 + 18x_2^3 + 4x_2^2\right\} = \{0\}$$

Step 3.2.5. Since variable x_3 is not eliminated completely, the elimination procedure is repeated on the new equation set which is formed by the above equation and the second equation of the *equation set 2.4*.

$$\left\{\begin{matrix} 15x_2^4 + 4x_2^3x_3 + 18x_2^3 + 4x_2^2 \\ 8x_2x_3 + x_2^2 + 2x_3^2 + 4x_2 \end{matrix}\right\} = \{0\} \qquad \text{(equation set 2.5)}$$

Step 3.2.1. (Second time) The maximum powers of x_3 in each equation in the *equation set 2.5* are 1 and 2 respectively.

Step 3.2.2. Among these maximum powers, 1 is the minimum so the first equation of *equation set 2.5* is selected.

Step 3.2.3. From $f_{2.5,1}$, $x_3 = \dfrac{-15x_2^4 - 18x_2^3 - 4x_2^2}{4x_2^3}$ is obtained.

Step 3.2.4. Insert x_3 in the second equation of *equation set 2.5*, and do necessary simplifications to obtain the following:

$$\left\{33x_2^8 + 284x_2^7 + 380x_2^6 + 144x_2^5 + 16x_2^4\right\} = \{0\} \qquad \text{(equation set 2.6)}$$

Step 3.2.5. Since variable x_3 is eliminated completely, the new equation set is the above set 2.6 and this is the end of the elimination procedure.

Step 3.3. End the reduction algorithm since there is just one equation left in the equation set and in terms of x_2 only. This equation *2.6* will be the second equation of *equation set 3*. (See below).

Iteration #3 (The Reduction Algorithm)

In this iteration, the reduction algorithm and the elimination procedure are applied once again to eliminate x_1 and x_2 exactly as discussed above to obtain a single polynomial in terms of x_3. We skip all the steps and directly present the end result:

$$\left\{11x^8 + 256x_3^7 + 280x_3^6 + 16x_3^4\right\} = \{0\}$$

The above equation is the third equation of *equation set 3* below.

The *reduction algorithm* thus gives the following polynomial equation set in terms of each variable:

$$\begin{Bmatrix} 33x_1^4 + 52x_1^3 - 64x_1^2 - 56x_1 + 16 \\ 33x_2^8 + 284x_2^7 + 380x_2^6 + 144x_2^5 + 16x_2^4 \\ 11x_3^8 + 256x_3^7 + 280x_3^6 + 16x_3^4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \qquad \text{(equation set 3)}$$

Step 4. These polynomials are solved numerically, yielding:

$$\begin{Bmatrix} root\ set_1 \\ root\ set_2 \\ root\ set_n \end{Bmatrix} \equiv \begin{Bmatrix} \{-2.061881, -0.880820, 0.236070, 1.130873\} \\ \{-7.061498, -1.009382, -0.327439, -0.207741, 0.000000\} \\ \{-0.218102, 0.000000, 0.275888, 1.092711, 22.122230\} \end{Bmatrix}$$

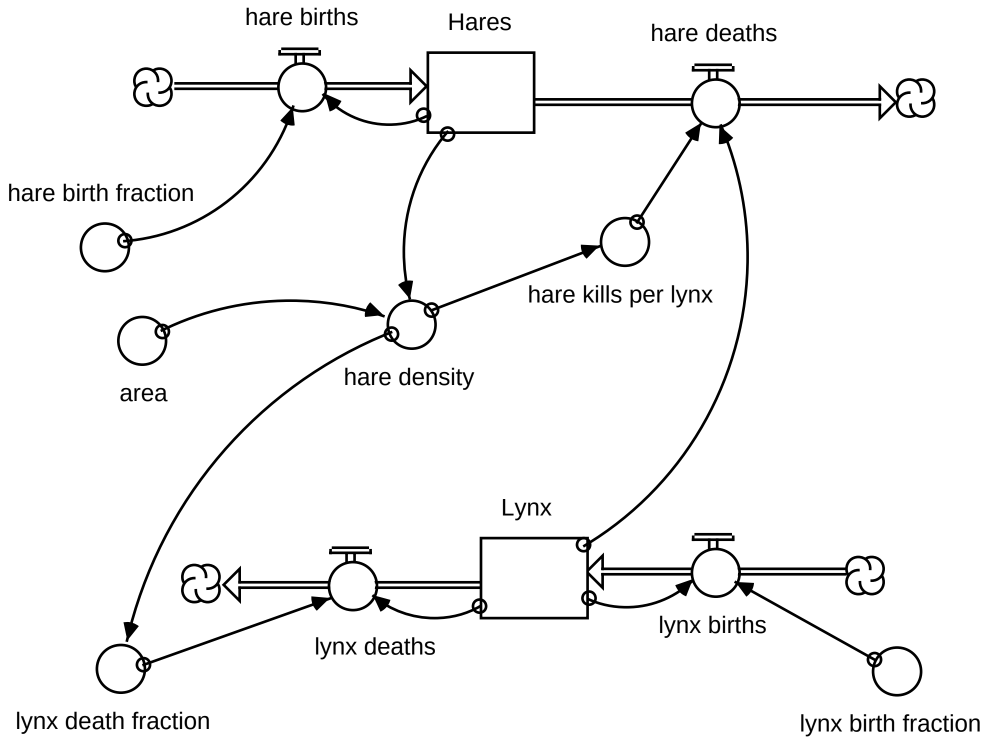
Step 5. *Step 4* implies 100 (4*5*5) potential roots. The combinations of roots that make all the equations in *equation set 1* equal zero are the equilibrium points of the given system, which are found to be as follows:

$$\begin{Bmatrix} 1st \text{ equilibrium point} \\ 2nd \text{ equilibrium point} \\ 3rd \text{ equilibrium point} \\ 4th \text{ equilibrium point} \end{Bmatrix} = \begin{Bmatrix} (x_1 = -2.061881, x_2 = -7.061498, x_3 = 22.122230) \\ (x_1 = -0.880820, x_2 = -1.009382, x_3 = 0.275888) \\ (x_1 = 0.236070, x_2 = -0.327439, x_3 = -0.218102) \\ (x_1 = 1.130873, x_2 = -0.207741, x_3 = 1.092711) \end{Bmatrix}$$

When each equilibrium point is plugged back in the given *equation set 1*, we verify that they are indeed the equilibria of the model.

EXAMPLE 2 (PREDATOR-PREY DYNAMICS)

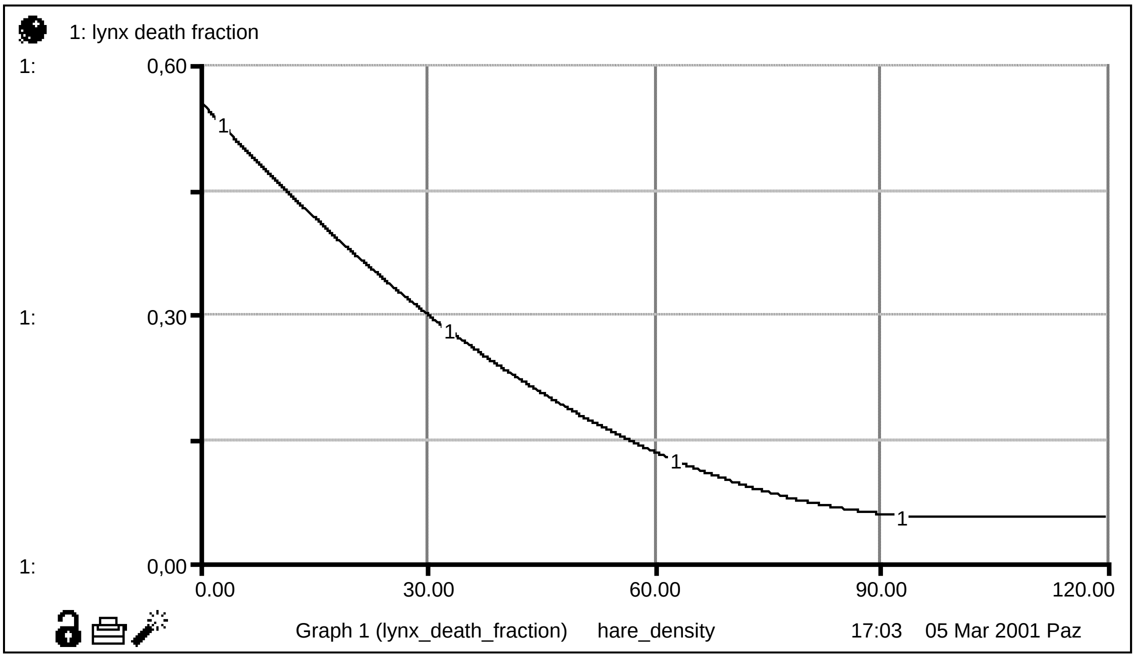
The following diagram belongs to one of the sample models packaged with the STELLA software:

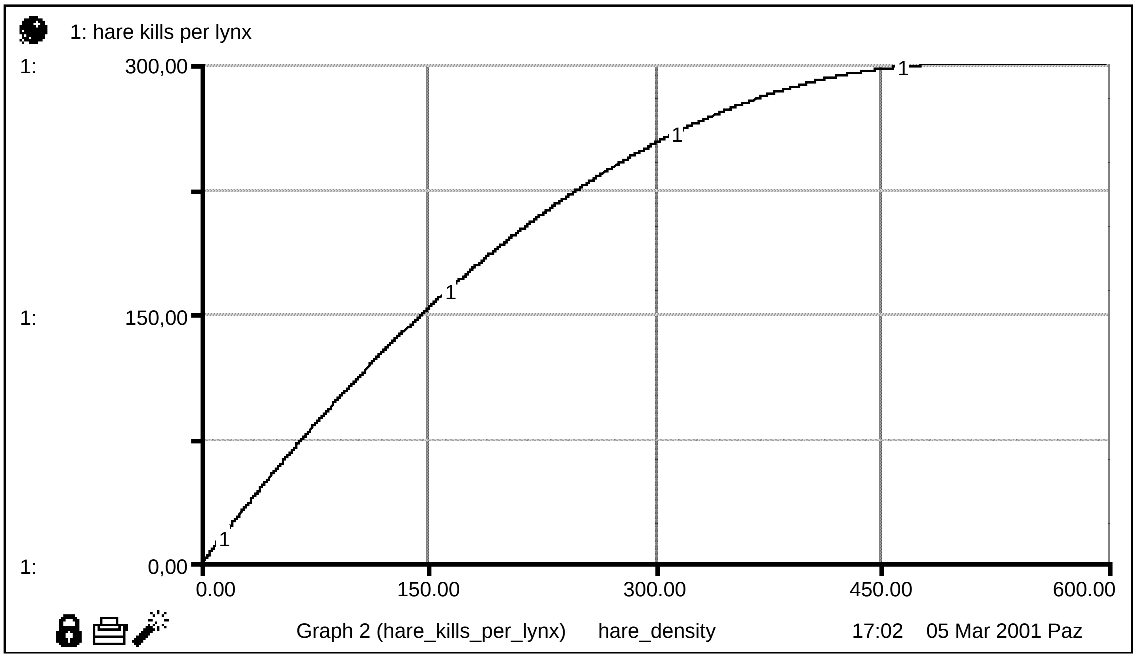


The equations of the above model are as follows:

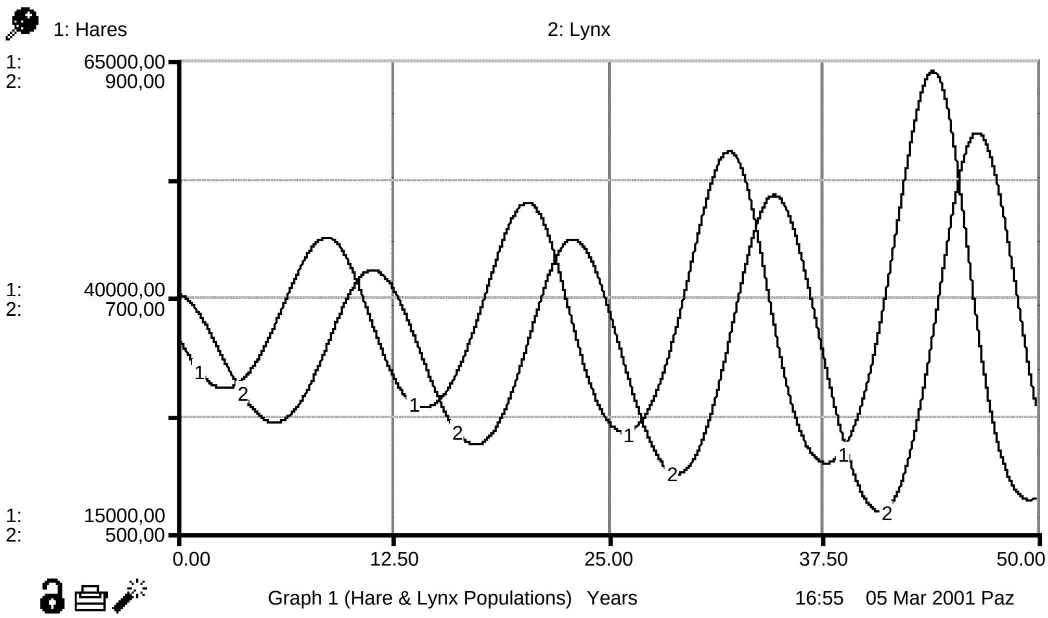
```
Hares(t) = Hares(t - dt) + (hare_births - hare_deaths) * dt
    INIT Hares = 35000
    INFLOWS:
        hare_births = Hares*hare_birth_fraction
    OUTFLOWS:
        hare_deaths = Lynx*hare_kills_per_lynx
Lynx(t) = Lynx(t - dt) + (lynx_births - lynx_deaths) * dt
    INIT Lynx = 700
    INFLOWS:
        lynx_births = Lynx*lynx_birth_fraction
    OUTFLOWS:
        lynx_deaths = Lynx*lynx_death_fraction
area = 1E3
hare_birth_fraction = 1.25
hare_density = Hares/area
hare_kills_per_lynx = if (hare_density<500) then -0.002*hare_density^2+2*hare_density else 500
hare_kills_per_lynx_2 = if (time<500) then -0.002*time^2+2*time else 500
lynx_birth_fraction = .25
lynx_death_fraction = if (hare_density<100) then 5e-5*hare_density^2-0.01*hare_density+0.55 else
0.05
lynx_death_fraction_2 = if (time<100) then 5e-5*time^2-0.01*time+0.55 else 0.05
```

In the original model, the variables “hare_kills_per_lynx” and “lynx_death_fraction” are graphical functions. For our purpose, we model them as polynomial functions of the variable “hare density”. (After certain levels of hare density, they are assumed constants). The following graphs show how these variables depend on hare density:





This model, with the given equations, basically produces unstable growing oscillations as seen in the following figure:



Our purpose is to find all the equilibrium points of this unstable non-linear model. After making the necessary initial simplifications, the equations of the model yields the following three systems of equations:

$$\begin{Bmatrix} \dot{h} \\ \dot{l} \end{Bmatrix} = \begin{Bmatrix} 1.25h + 2*10^{-9}h^2l - 0.002hl \\ -0.3l - 5*10^{-11}h^2l + 0.00001hl \end{Bmatrix} \qquad for \quad l > 0 \quad \& \quad 100{,}000 > h > 0$$

(equation set 1.1)

$$\begin{Bmatrix} \dot{h} \\ \dot{l} \end{Bmatrix} = \begin{Bmatrix} 1.25h + 2*10^{-9}h^2l - 0.002hl \\ 0.2l \end{Bmatrix} \qquad for \quad l > 0 \quad \& \quad 500{,}000 > h > 100{,}000$$

(equation set 1.2)

$$\begin{Bmatrix} \dot{h} \\ \dot{l} \end{Bmatrix} = \begin{Bmatrix} 1.25h - 500l \\ 0.2l \end{Bmatrix} \qquad for \quad l > 0 \quad \& \quad h > 500{,}000$$

(equation set 1.3)

Step 1. Set all the equations in *equation set 1.1* equal to zero.

$$\begin{Bmatrix} \dot{h} \\ \dot{l} \end{Bmatrix} = \begin{Bmatrix} 1.25h + 2*10^{-9}h^2l - 0.002hl \\ -0.3l - 5*10^{-11}h^2l + 0.00001hl \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

Step 2. There are no division operators in the above equation set, so *equation 2* is obtained without any further effort.

$$\begin{Bmatrix} 1.25h + 2*10^{-9}h^2l - 0.002hl \\ -0.3l - 5*10^{-11}h^2l + 0.00001hl \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

(equation set 2)

Step 3. Find one-dimensional polynomials for each variable, by the reduction technique.

Iteration #1 (The Reduction Algorithm)

Step 3.1. Select *h* as the variable that is going to be kept. The initial equation set is $\{f_{2,1}, f_{2,2}\}$.

Step 3.2. Select variable *l* to be eliminated. Eliminate *l* with the elimination procedure:

Elimination Procedure

Step 3.2.1. The maximum powers of *l* in each equation in the equation set are 1 and 1 respectively.

Step 3.2.2. The maximum powers are equal, there are two minima. Break the tie arbitrarily and select the equation with smaller index, i.e. the first equation in the *equation set*.

Step 3.2.3. From $f_{2,1}$, $l = \dfrac{125h}{-2*10^{-7}h^2 + 0.2h}$ is obtained.

Step 3.2.4. Insert *l* to the second equation in *equation set 2*,

$$\left\{-0.3\left(\frac{125h}{-2*10^{-7}h^2+0.2h}\right) - 5*10^{-11}h^2\left(\frac{125h}{-2*10^{-7}h^2+0.2h}\right) + 0.00001h\left(\frac{125h}{-2*10^{-7}h^2+0.2h}\right)\right\} = \{0\}$$

do necessary simplifications and multiplications to get rid of the division operator.

$$\left\{6.25*10^{-8}h^3 - 0.0125h^2 + 375h\right\} = \{0\} \qquad \text{(equation set 2.1)}$$

Step 3.2.5. End the elimination procedure since variable *l* is eliminated completely. Also eliminate the first equation so that the new equation set becomes *equation set 2.1*.

Step 3.3. End the reduction algorithm since there is just one equation left in the equation set and in terms of h only. This equation *2.1* will be the first equation of *equation set 3*. (See below, the very end of this example).

Iteration #2 (The Reduction Algorithm)

Step 3.1. Select *l* as the variable that is going to be kept. The initial equation set is again $\{f_{2,1}, f_{2,2}\}$.

Step 3.2. Select variable h to be eliminated. Eliminate h with the elimination procedure:

Elimination Procedure

Step 3.2.1. The maximum powers of h in each equation in the equation set are 2 and 2 respectively.

Step 3.2.2. The maximum powers are equal, there are two minima. Break the tie arbitrarily and select the equation with smaller index, i.e. the first equation in the *equation set*.

Step 3.2.3. From $f_{2,1}$, $h^2 = \frac{-1.25h + 0.002hl}{2*10^{-9}l}$ is obtained.

Step 3.2.4. Insert h^2 to the second equation in *equation set 2*,

$$\left\{-0.3l - 5*10^{-11}\frac{-1.25h + 0.002hl}{2*10^{-9}l}l + 0.00001hl\right\} = \{0\}$$

do necessary simplifications and multiplications to get rid of the division operator.

$$\{-6l^2 + 0.625hl - 0.0008hl^2\} = \{0\}$$

Step 3.2.5. Since variable h is not eliminated completely, the elimination procedure is repeated on the new equation set which is formed by the above equation and the first equation of the *equation set 2*.

$$\begin{Bmatrix} 1.25h + 2*10^{-9}h^2l - 0.002hl \\ -6l^2 + 0.625hl - 0.0008hl^2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \qquad \text{(equation set 2.2)}$$

Step 3.2.1. (Second time) The maximum powers of h in each equation in the equation set are 2 and 1 respectively.

Step 3.2.2. Among these maximum powers, 1 is the minimum, so the last equation is selected.

Step 3.2.3. From $f_{2.2,2}$, $h = \frac{-6l^2}{0.625l - 0.0008l^2}$ is obtained.

Step 3.2.4. Insert h to the first equation in *equation set 2.2*,

$$\left\{1.25\frac{-6l^2}{0.625l - 0.0008l^2} + 2*10^{-9}\left(\frac{-6l^2}{0.625l - 0.0008l^2}\right)^2 l - 0.002\frac{-6l^2}{0.625l - 0.0008l^2}l\right\} = \{0\}$$

do necessary simplifications and multiplications to get rid of the division operator.

$$\left\{0.09672l^5 - 135l^4 + 46875l^3\right\} = \left\{0\right\} \qquad \text{(equation set 2.3)}$$

Step 3.2.5. End the elimination procedure since variable *h* is eliminated completely. Also eliminate the second equation so that the new equation set becomes the *equation 2.3*.

Step 3.3. End the reduction algorithm since there is just one equation left in the equation set and in terms of *l* only. This equation in set *2.3* will be the second equation of *equation set 3* below.

The *reduction algorithm* thus gives the following polynomial equation set in terms of each variable:

$$\begin{Bmatrix} 6.25*10^{-8}h^3 - 0.0125h^2 + 375h \\ 0.09672l^5 - 135l^4 + 46875l^3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \qquad \text{(equation set 3)}$$

Step 4. These polynomials are solved numerically, yielding:

$$\begin{Bmatrix} \textit{root set}_h \\ \textit{root set}_l \end{Bmatrix} \equiv \begin{Bmatrix} \{0, 36754.45, 163245.55\} \\ \{0, 648.848, 746.934\} \end{Bmatrix}$$

Step 5. *Step 4* implies 9 (3*3) potential roots. The combinations of these roots that make all the equations in *equation set 1.1* equal zero are as follows:

$$\begin{Bmatrix} 1st\ \text{point} \\ 2nd\ \text{point} \\ 3rd\ \text{point} \end{Bmatrix} = \begin{Bmatrix} (\mathrm{h} = 0, \mathrm{l} = 0) \\ (\mathrm{h} = 36754.45, \mathrm{l} = 648.848) \\ (\mathrm{h} = 163245.55, \mathrm{l} = 746.934) \end{Bmatrix}$$

But the 3*rd* point is outside the range of the *equation set 1.1*. So it can not be an equilibrium point. The *equation set 1.1* thus has two equilibrium points.

Since in this example the model consists of three different equation sets, we repeat the above steps for *equation set 1.2* and *equation set 1.3* and in each case we obtain (h=0, l=0) as the only point that makes the equations zero. But this point is outside the range of both equation sets. So, the second and third equation sets of the model provide no additional equilibrium points.

If we combine the above results obtained from the three equation sets of the model, we obtain the following equilibrium points:

$$\begin{Bmatrix} 1st\ \text{equilibrium point} \\ 2nd\ \text{equilibrium point} \end{Bmatrix} = \begin{Bmatrix} (\mathrm{h} = 0, \mathrm{l} = 0) \\ (\mathrm{h} = 36754.45, \mathrm{l} = 648.848) \end{Bmatrix}$$

When each equilibrium point is plugged back in the given *equation sets*, we verify that they are indeed the equilibria of the model.

CONCLUSION

An algorithm to find all the equilibrium points of a given non-linear dynamic model is discussed. The method aims to work for a rather general subset of non-linear systems, provided that all non-linearities are/can be expressed in polynomial terms. The significance of the method is that; i- it can greatly speed up model analysis by providing the equilibrium information prior to simulation, and ii- it can help verify the results obtained from simulations (numerical simulation may skip an existing equilibrium and "create" spurious equilibria). The method is demonstrated on two examples. The algorithm works well, except when there are infinite number of equilibria on an N-dimensional plane. Future work will focus on this problematic case. Finally, there are some issues of speed and numerical accuracy, the other two main topics of current and future research. The method can be coded in the future as an integral part of the existing System Dynamics software.

APPENDIX 1. THE ELIMINATION PROCEDURE

The elimination procedure starts with *n* variables and *n* equations. At the end it gives *n-1* equations with *n-1* variables.

Steps of The Elimination Procedure

Step 1. Find the maximum powers of the variable x_i in each equation from the given equation set. (Note that x_i is the variable that is selected for elimination).

Step 2. Among these maximum powers, find the minimum power (*p*) that is non zero. Select this equation (that gives the minimum of the maximum powers).

Step 3. From selected equation obtain

$$x_i^p = \frac{f_{numerator}(x_i, x_i^2, x_i^3, \ldots, x_i^{p-1}, other\ x_j\text{'s}\ where\ j \in V)}{f_{denominator}(other\ x_j\text{'s}\ where\ j \in V)}$$

V is the variable set including all the variables of the given equation set except x_i.

Step 4. Insert x_i^p in the other equations of the given equation set and do the necessary multiplications to get rid of the division operator [Appendix-2]. Check if there is no $\{x_i^k : k \geq p\}$ term left. Else, do the same insertion recursively until no $\{x_i^k : k \geq p\}$ is left in these equations.

Step 5. If variable x_i is eliminated completely, also eliminate the selected equation and end the procedure. Otherwise update the equation set using the selected equation, return back to *step 1* of the elimination procedure and apply it on the updated equation set.

(Note that x_i and the given equation set were determined by the reduction algorithm.)

APPENDIX 2. GETTING RID OF THE DIVISION OPERATOR

Assume that the following is the given equation:

$$f(x_1, x_2) = \frac{x_1 * x_2}{x_2} + 1 = 0$$

The wrong way of getting rid of the division operator would be a division cancellation to yield the following:

$$f(x_1, x_2) = x_1 + 1 = 0$$

The method presented does not allow division cancellations, because such cancellations may result in loosing some of the roots. The proper way of getting rid of divisions is to multiply the equation by the necessary terms. So after multiplying all terms by x_2, the given equation becomes:

$$f(x_1, x_2) = x_1 * x_2 + x_2 = 0$$

REFERENCES

Aracil, J and Toro M., 1988. Qualitative Analysis of System Dynamics Ecological Models, System Dynamics Review 4: 56-60.

Barlas, Y and Çivi, H., 1994. Designing General-Purpose Algorithms For Equilibria and Stability Analysis of Non-Linear Dynamic Models, Boğaziçi University Research Paper, Istanbul.

Jenner, W.E., 1963. Rudiments of Algebraic Geometry, Oxford University Press: 6-26.

Press W.H., Flannery B.P. 1986. *Numerical Recipes.* Cambridge University Press.

Rayna, G., 1987. Reduce Software for Algebraic Computation, Springer-Verlag.

Santina Mohammed S. and D'Carpio-Montalvo Paul. 1991. *Analytical, Numerical, And Computational Methods For Science and Engineering.* Prentice-Hall International, Inc.

Woodford, Chris. 1992. *Solving Linear and Non-linear Equations.* Ellis Horward Limited.