Differential elimination and biological modelling

François Boulier
Équipe calcul formel,
Laboratoire d'Informatique Fondamentale de Lille
Université Lille 1

IRISA, SYMBIOSE, 16 novembre 2006

1 Differential elimination

2 LÉPISME

3 Modelling ostreococcus tauri

4 The VKBL model

5 The Morant model

6 Conclusion

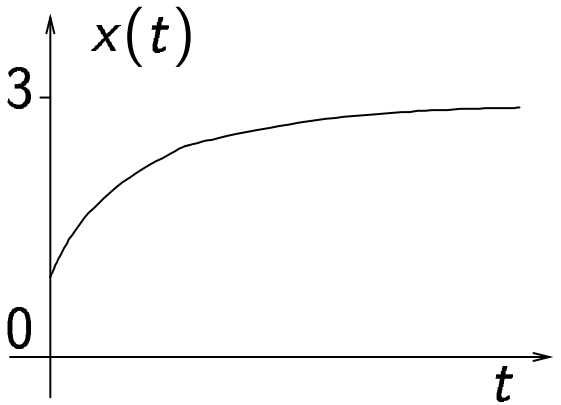
Differential equations : numerical integration

$$\begin{cases} \dot{x}(t) &= x(t)\,(3 - x(t)) \\ x(0) &= 1 \end{cases}$$

The unknown is a function $t \mapsto x(t)$.

Numerical integration consists in approximating the graph of the solution by a finite number of points. It is not computer algebra.

t	x(t)
0.	1.
0.1	1.20888
0.2	1.43019
0.3	1.65459
...	...
8.9	2.99999
9.	2.99999



Differential equations : close form integration

```
    |\^/|     Maple 9 (IBM INTEL LINUX)
._|\|   |/|_. Copyright (c) Maplesoft, a division of Waterloo Maple Inc
 \  MAPLE  /  All rights reserved. Maple is a trademark of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
> edo := diff (x(t),t) = x(t)*(3-x(t));
                      d
            edo := -- x(t) = x(t) (3 - x(t))
                      dt

> dsolve (edo, x(t));
                                3
                x(t) = ------------------
                       1 + 3 exp(-3 t) _C1
```

It is computer algebra.

It is usually not possible.

We do not do it in the computer algebra team of Lille.

Differential equations : simplification of systems

The following DAE (Hairer, Wanner II) has derivation index 2.
The unknowns are three functions $x(t)$, $y(t)$ and $z(t)$.

$$\begin{cases} \dot{x}(t) &= 0.7 \cdot y(t) + \sin(2.5 \cdot z(t)) \\ \dot{y}(t) &= 1.4 \cdot x(t) + \cos(2.5 \cdot z(t)) \\ 1 &= x^2(t) + y^2(t). \end{cases}$$

There is a missing ODE for numerical integration : $\dot{z}(t) = ???$.

Differential elimination software permit to extract it automatically

Differential elimination methods require polynomial systems

$$\begin{cases} \dot{x}(t) &= 0.7 \cdot y(t) + s(t) \\ \dot{y}(t) &= 1.4 \cdot x(t) + c(t) \\ 1 &= x^2(t) + y^2(t) \end{cases} \qquad \begin{cases} \dot{s}(t) &= 2.5 \cdot \dot{z}(t) \cdot c(t) \\ \dot{c}(t) &= -2.5 \cdot \dot{z}(t) \cdot s(t) \\ 1 &= s^2(t) + c^2(t). \end{cases}$$

A demo of Rosenfeld–Gröbner using the MAPLE diffalg package.

Differential equations : simplification of systems

The following DAE (Hairer, Wanner II) has derivation index 2.
The unknowns are three functions $x(t)$, $y(t)$ and $z(t)$.

$$\begin{cases} \dot{x}(t) &= 0.7 \cdot y(t) + \sin(2.5 \cdot z(t)) \\ \dot{y}(t) &= 1.4 \cdot x(t) + \cos(2.5 \cdot z(t)) \\ 1 &= x^2(t) + y^2(t). \end{cases}$$

There is a missing ODE for numerical integration : $\dot{z}(t) = ???$.

Differential elimination software permit to extract it automatically

Differential elimination methods require polynomial systems

$$\begin{cases} \dot{x}(t) &= 0.7 \cdot y(t) + s(t) \\ \dot{y}(t) &= 1.4 \cdot x(t) + c(t) \\ 1 &= x^2(t) + y^2(t) \end{cases} \qquad \begin{cases} \dot{s}(t) &= 2.5 \cdot \dot{z}(t) \cdot c(t) \\ \dot{c}(t) &= -2.5 \cdot \dot{z}(t) \cdot s(t) \\ 1 &= s^2(t) + c^2(t). \end{cases}$$

A demo of Rosenfeld–Gröbner using the MAPLE diffalg package.

diffalg versus BLAD

diffalg

I wrote the first version for MAPLE in 1995 while a postdoc at the university of Waterloo.
Much improved by E. Hubert, A. Wittkopf, F. Lemaire.
It is designed for interactive use. Not so easy to manipulate :

- how to choose rankings ?
- terrifying worst case complexity.

The *Bibliothèques Lilloises d'Algèbre Différentielle*

http ://www.lifl.fr/~boulier/BLAD.
Open source C libraries (LGPL). 36000 lines of C code.
Designed to be software components.
Provide functionnalities to overcome the above problems.
A less fancy demo of Rosenfeld–Gröbner using BLAD.

diffalg versus BLAD

diffalg

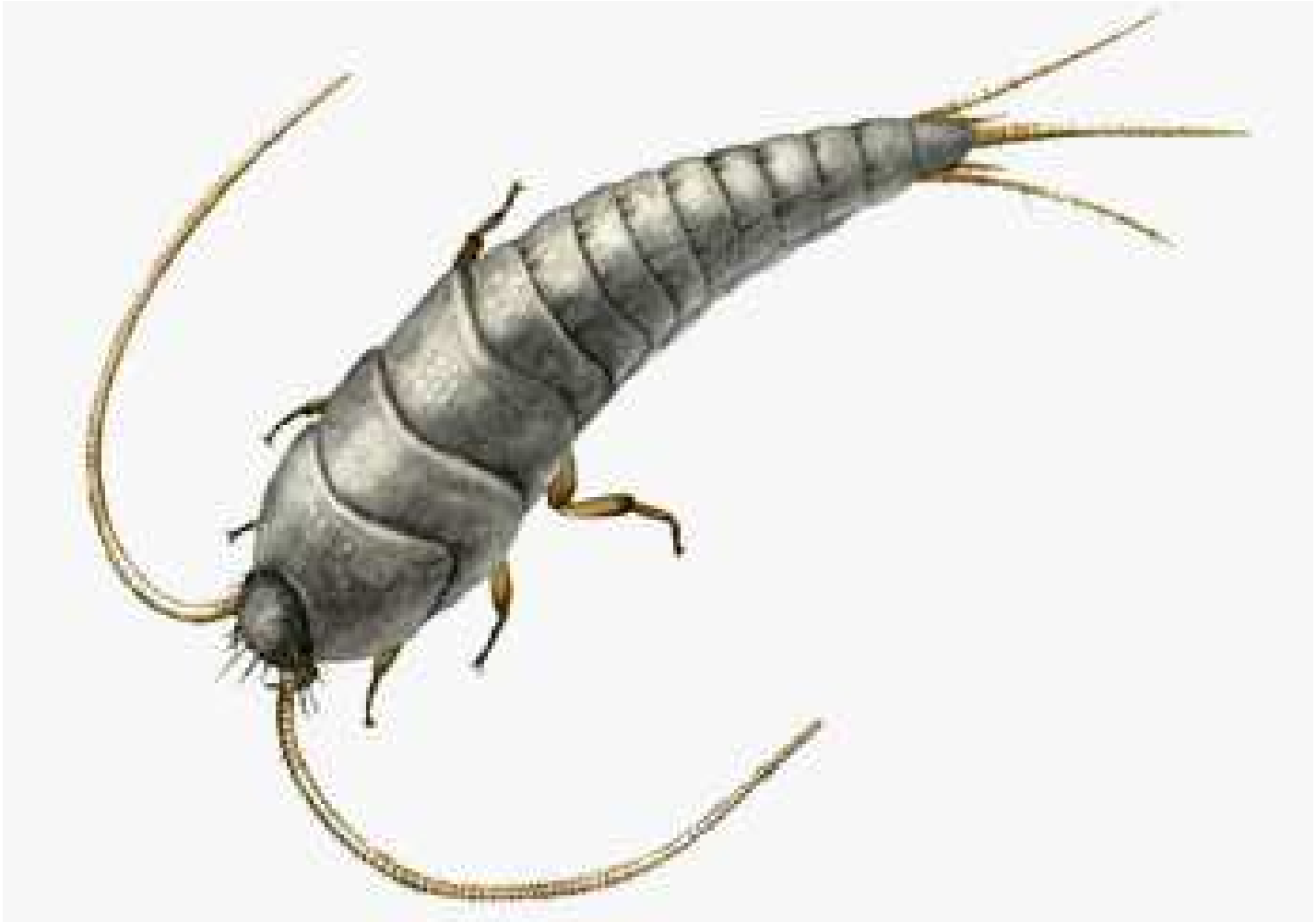
I wrote the first version for MAPLE in 1995 while a postdoc at the university of Waterloo.
Much improved by E. Hubert, A. Wittkopf, F. Lemaire.
It is designed for interactive use. Not so easy to manipulate :

- how to choose rankings ?
- terrifying worst case complexity.

The *Bibliothèques Lilloises d'Algèbre Différentielle*

http ://www.lifl.fr/~boulier/BLAD.
Open source C libraries (LGPL). 36000 lines of C code.
Designed to be software components.
Provide functionnalities to overcome the above problems.
A less fancy demo of Rosenfeld–Gröbner using BLAD.

The lepism (or silver fish)



Logiciels pour l'Estimation de Paramètres et l'Identification Systématique de Modèles

Statement of the problem

Given

- a parametric ODE system (four parameters k_e, V_e, k_{12}, k_{21}) :

$$\begin{aligned} \dot{x}_1(t) &= -k_{12}\, x_1(t) + k_{21}\, x_2(t) - \frac{V_e\, x_1(t)}{k_e + x_1(t)}, \\ \dot{x}_2(t) &= k_{12}\, x_1(t) - k_{21}\, x_2(t). \end{aligned}$$

- some measures :
  $x_1(t)$ is observed ;
  $x_2(t)$ is not observed

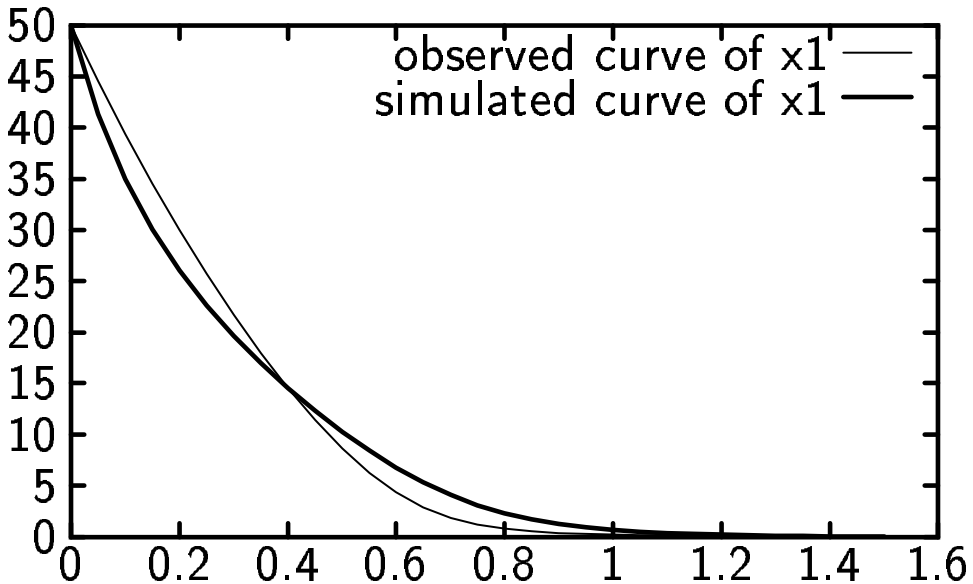
t	$x_1(t)$
$0.00000e-01$	$5.00000e+01$
$0.50000e-01$	$4.45078e+01$
...	
$1.50000e+00$	$4.95270e-02$

- possibly some extra information : $x_2(0) = 0$; $k_e = 7$.

Estimate the values of the unknown parameters V_e, k_{12}, k_{21}.

There exists a purely numerical method

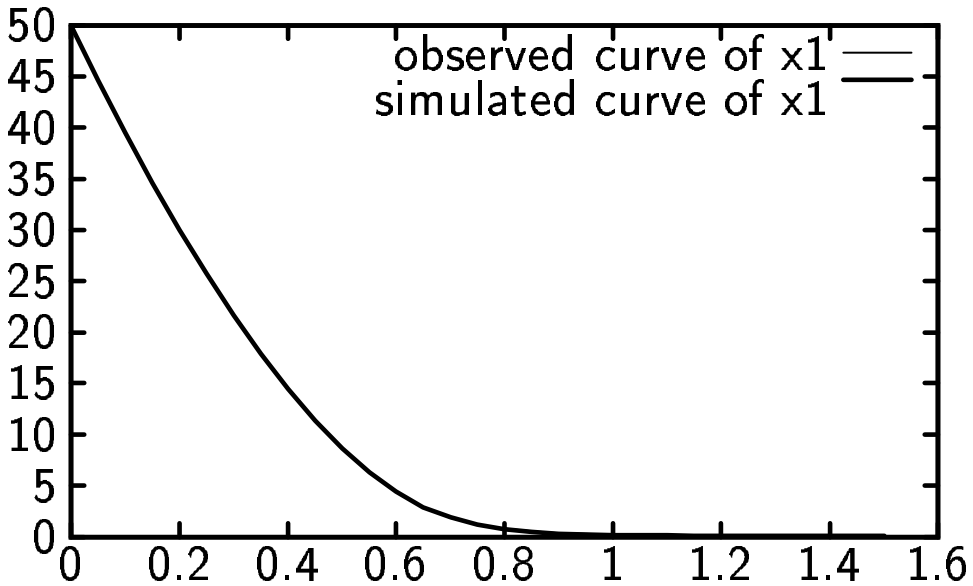
1. Give random values to k_{12}, k_{21}, V_e.
2. Integrate numerically the ODE and get a graph for $x_1(t)$.



3. If the error is too large, update k_{12}, k_{21}, V_e by the Levenberg–Marquardt method and goto step 2.

There exists a purely numerical method

1. Give random values to k_{12}, k_{21}, V_e.
2. Integrate numerically the ODE and get a graph for $x_1(t)$.



The Levenberg–Marquardt method ends in a wrong local minimum

$$k_{21} = .16, \quad k_{12} = .76, \quad V_e = 82.8.$$

Should be

$$k_{21} = .5, \quad k_{12} = 3, \quad V_e = 101.$$

Differential elimination for guessing good initial values

$$\begin{aligned}\dot{x}_1(t) &= -k_{12}\,x_1(t) + k_{21}\,x_2(t) - \frac{V_e\,x_1(t)}{k_e + x_1(t)},\\ \dot{x}_2(t) &= k_{12}\,x_1(t) - k_{21}\,x_2(t).\end{aligned}$$

1. Eliminate the non observed variable $x_2(t)$ using Rosenfeld–Gröbner or, better, PARDI.

$$\ddot{x}_1\,(x_1+k_e)^2+[k_{12}+k_{21}]\,\dot{x}_1\,(x_1+k_e)^2+[V_e]\,\dot{x}_1\,k_e+[k_{21}\;V_e]\,x_1\,(x_1+k_e).$$

2. Evaluate the ODE for many different values of t. By linear least squares, estimate the [parameters blocks].
3. Solve the parameters blocks w.r.t. parameters :

$$k_{12} = 0.45, \quad k_{21} = 1.65, \quad V_e = 87.29.$$

4. Run the optimization method starting from these values.

There are (numerical !) difficulties

Evaluation of the ODE

Difficult to numerically evaluate the derivatives.

Getting the parameters from the [blocks]

Over the example, solve

$$k_{12} + k_{21} = value_1, \quad V_e = value_2, \quad k_{21}\, V_e = value_3.$$

In general, beware algebraic relations amongs blocks !

$$k_{21} = value_1, \quad V_e = value_2, \quad k_{21}\, V_e = value_3.$$

Summary

A very interesting collaboration

We work with applied maths researchers who carry out real examples
We develop a BLAD based software implementing their methods.
Article at ICPSS04 (w. L. Denis–Vidal, T. Henin, F. Lemaire).

Strong connection with non commutative algebra

Identification Systématique de Modèles (black vs white box)

LÉPISME continues with the ANR project proposal IMAGES

Investigation du Métabolisme végétal des Acides Gras par Étude Systémique
Project led by a biological lab. *Génie Enzymatique et Cellulaire*.
Modelling the biosynthesis of fatty acids and oil in oilseed embryos.

A pluridisciplinary working group

The Institut de Recherche Interdisciplinaire.
The two leaders of the pluridisciplinary working group.

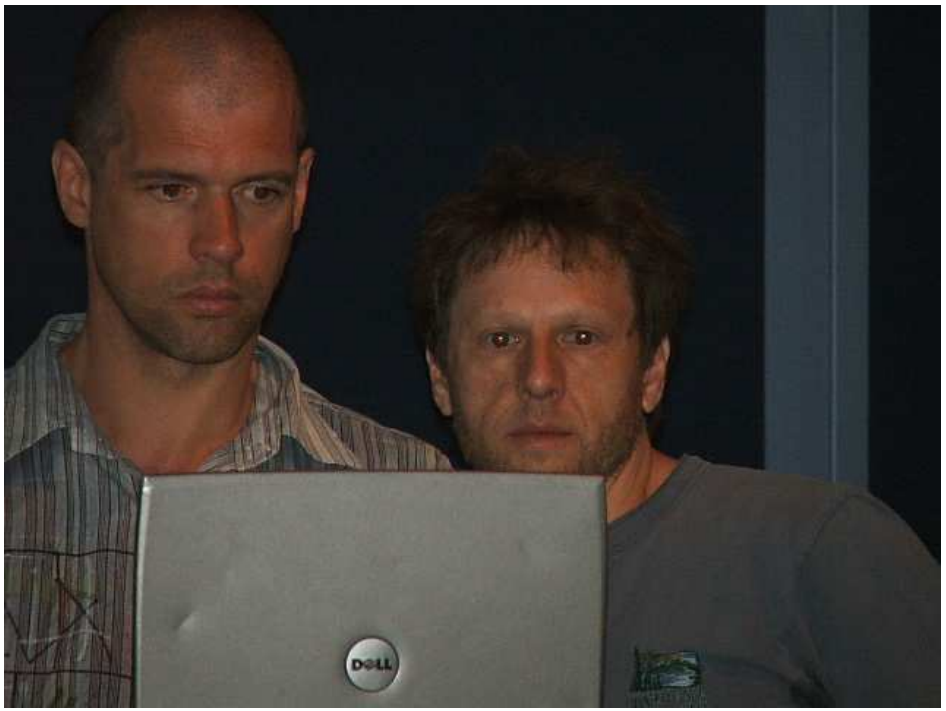
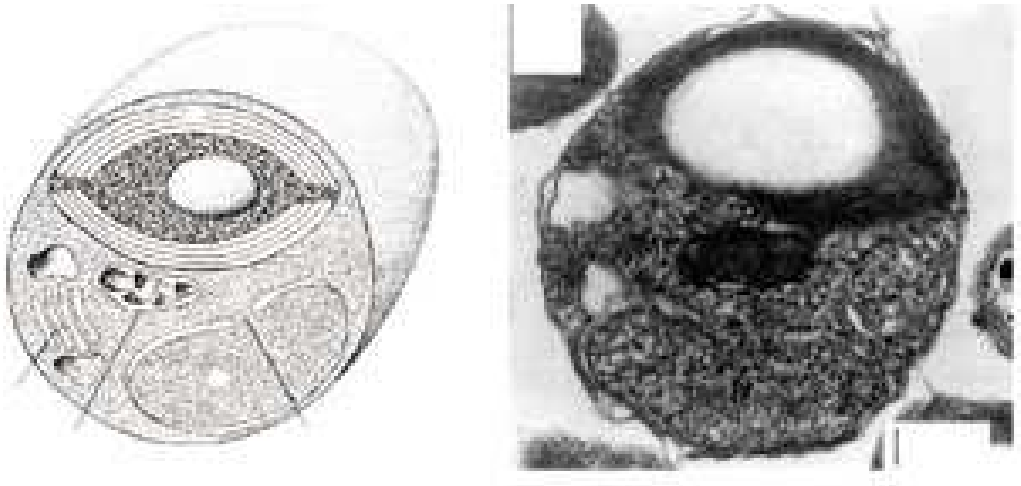


Fig.: F.-Y. Bouget (Banyuls) and M. Lefranc (Lille).

Ostreococcus tauri



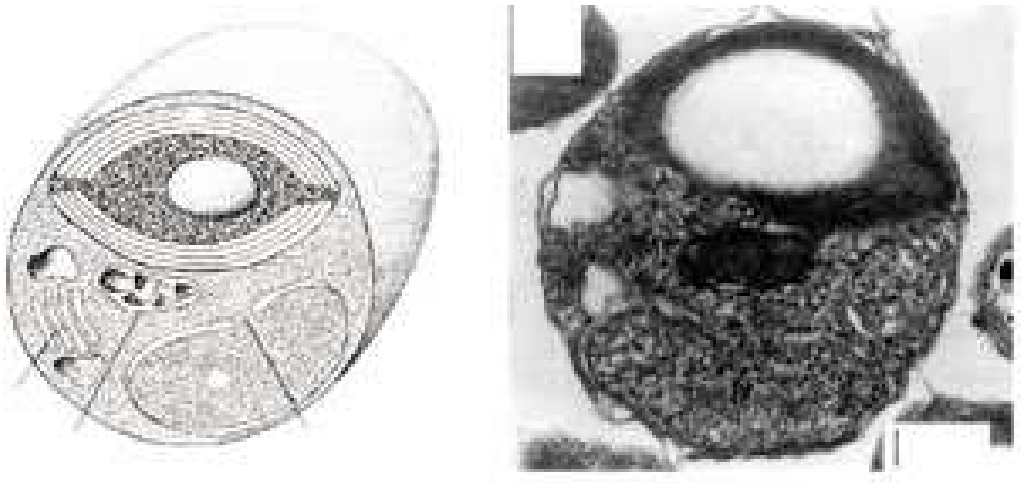
A green alga discovered in 1994 in the Etang de Thau.
The simplest non parasitic organism known. A circadian clock.
Genom completely sequenced in 2006 (Evelyne Derelle et *al.*).

Given a model as a system of parametric ODE

Ranges of parameters w.r.t. which the model oscillates ?

- Differential elimination + Poincaré–Bendixson theorem
- Algebraic elimination + Routh–Hurwitz (Hopf bifurcation)

Ostreococcus tauri



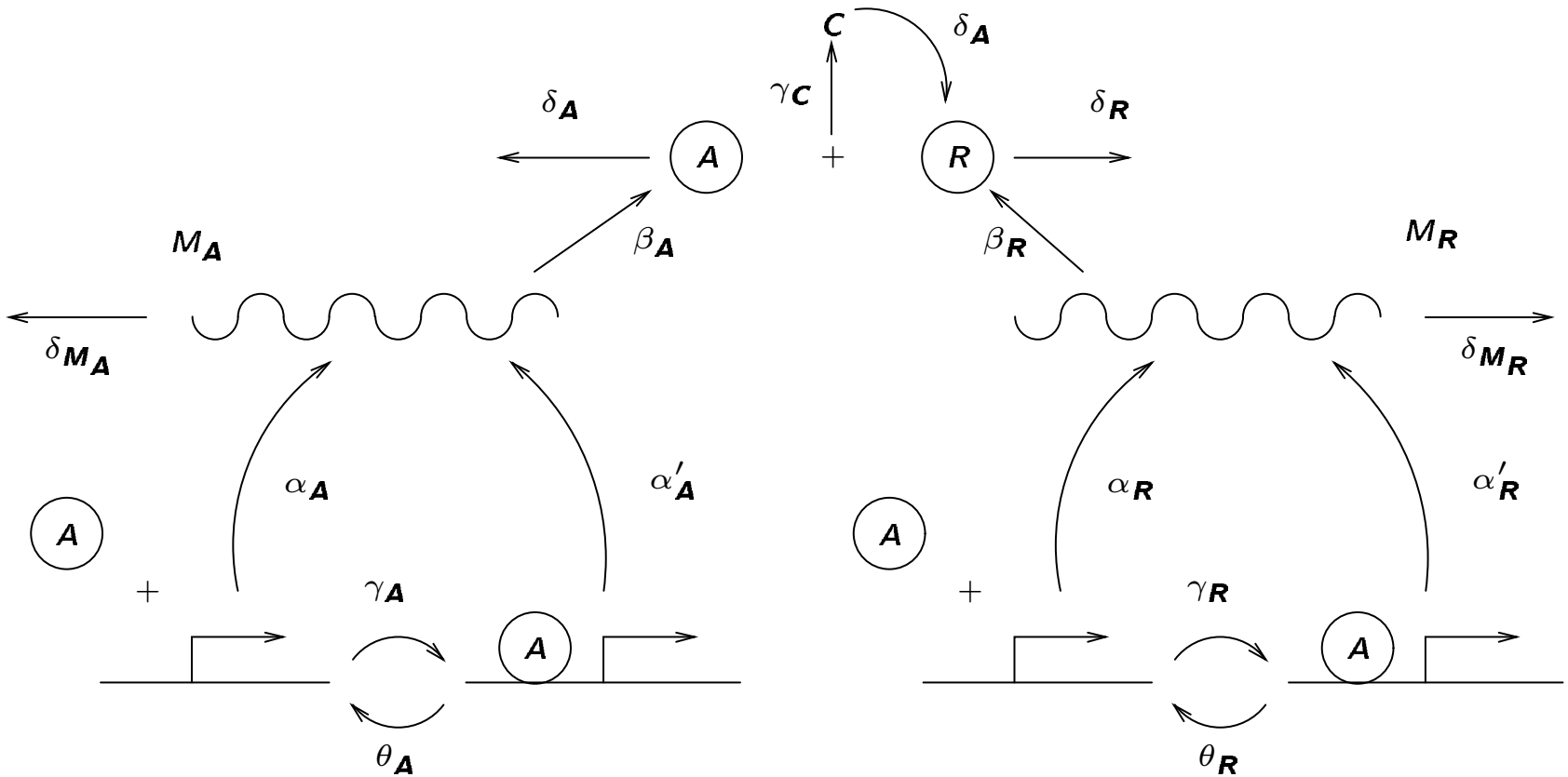
A green alga discovered in 1994 in the Etang de Thau.
The simplest non parasitic organism known. A circadian clock.
Genom completely sequenced in 2006 (Evelyne Derelle et *al.*).

Given a model as a system of parametric ODE

Ranges of parameters w.r.t. which the model oscillates ?

- Differential elimination + Poincaré–Bendixson theorem
- Algebraic elimination + Routh–Hurwitz (Hopf bifurcation)

Differential elimination + Poincaré–Bendixson



2 genes : an activator and a repressor.

7 variables and 15 parameters.

The VKBL idea : searching oscillations using the Poincaré–Bendixson theorem

- Poincaré–Bendixson. It only holds in the two variables setting. It gives sufficient conditions so that oscillations occur.
- Model reduction. If $g(x, y) \gg f(x, y)$ then a system

$$\dot{x}(t) = f(x(t), y(t)), \quad \dot{y}(t) = g(x(t), y(t))$$

may possibly be approximated by

$$\dot{x}(t) = f(x(t), y(t)), \quad \cancel{\dot{y}(t)}\ 0 = g(x(t), y(t)).$$

- Get rid of 5 of the 7 variables.
  Get parameters values w.r.t. which Poincaré–Bendixson holds.
  Check these values produce oscillations for the initial system.

Differential elimination for realizing VKBL idea

$$
\begin{aligned}
\dot{D}_A(t) &= \theta_A(1 - D_A(t)) - \gamma_A\, D_A(t)\, A(t),\\
\dot{D}_R(t) &= \theta_R\,(1 - D_R(t)) - \gamma_R\, D_R(t)\, A(t),\\
\dot{M}_A(t) &= \alpha'_A\,(1 - D_A(t)) + \alpha_A\, D_A(t) - \delta_{M_A}\, M_A(t),\\
\dot{M}_R(t) &= \alpha'_R\,(1 - D_R(t)) + \alpha_R\, D_R(t) - \delta_{M_R}\, M_R(t),\\
\dot{A}(t) &= \beta_A\, M_A(t) + \theta_A\,(1 - D_A(t)) + \theta_R\,(1 - D_R(t)) - A(t)\,(\cdots),\\
\dot{R}(t) &= \beta_R\, M_R(t) - \gamma_C\, A(t)\, R(t) + \delta_A\, C(t) - \delta_R\, R(t),\\
\dot{C}(t) &= \gamma_C\, A(t)\, R(t) - \delta_A\, C(t).
\end{aligned}
$$

Differential elimination for realizing VKBL idea

$$
\begin{aligned}
\dot{D}_A(t) &= \theta_A(1 - D_A(t)) - \gamma_A\, D_A(t)\, A(t),\\
\dot{D}_R(t) &= \theta_R\,(1 - D_R(t)) - \gamma_R\, D_R(t)\, A(t),\\
\dot{M}_A(t) &= \alpha'_A\,(1 - D_A(t)) + \alpha_A\, D_A(t) - \delta_{M_A(t)}\, M_A(t),\\
\dot{M}_R(t) &= \alpha'_R\,(1 - D_R(t)) + \alpha_R\, D_R(t) - \delta_{M_R}\, M_R(t),\\
\dot{A}(t) &= \beta_A\, M_A(t) + \theta_A\,(1 - D_A(t)) + \theta_R\,(1 - D_R(t)) - A(t)\,(\cdots),\\
\dot{R}(t) &= \beta_R\, M_R(t) - \gamma_C\, A(t)\, R(t) + \delta_A\, C(t) - \delta_R\, R(t),\\
\dot{C}(t) &= \gamma_C\, A(t)\, R(t) - \delta_A\, C(t).
\end{aligned}
$$

Differential elimination for realizing VKBL idea

$$
\begin{aligned}
\dot{D}_A(t) &= \theta_A(1 - D_A(t)) - \gamma_A\, D_A(t)\, A(t),\\
\dot{D}_R(t) &= \theta_R\,(1 - D_R(t)) - \gamma_R\, D_R(t)\, A(t),\\
\dot{M}_A(t) &= \alpha'_A\,(1 - D_A(t)) + \alpha_A\, D_A(t) - \delta_{M_A(t)}\, M_A(t),\\
\dot{M}_R(t) &= \alpha'_R\,(1 - D_R(t)) + \alpha_R\, D_R(t) - \delta_{M_R}\, M_R(t),\\
\dot{A}(t) &= \beta_A\, M_A(t) + \theta_A\,(1 - D_A(t)) + \theta_R\,(1 - D_R(t)) - A(t)\,(\cdots),\\
\dot{R}(t) &= \beta_R\, M_R(t) - \gamma_C\, A(t)\, R(t) + \delta_A\, C(t) - \delta_R\, R(t),\\
\dot{C}(t) &= \gamma_C\, A(t)\, R(t) - \delta_A\, C(t).
\end{aligned}
$$

For some ranking, Rosenfeld–Gröbner gives the VKBL system :

$$
\begin{aligned}
\dot{R}(t) &= f(R(t),\, C(t),\, A(t))\\
\dot{C}(t) &= g(R(t),\, C(t),\, A(t))\\
A^2(t) &= h(R(t), C(t))
\end{aligned}
$$

Differential elimination for realizing VKBL idea

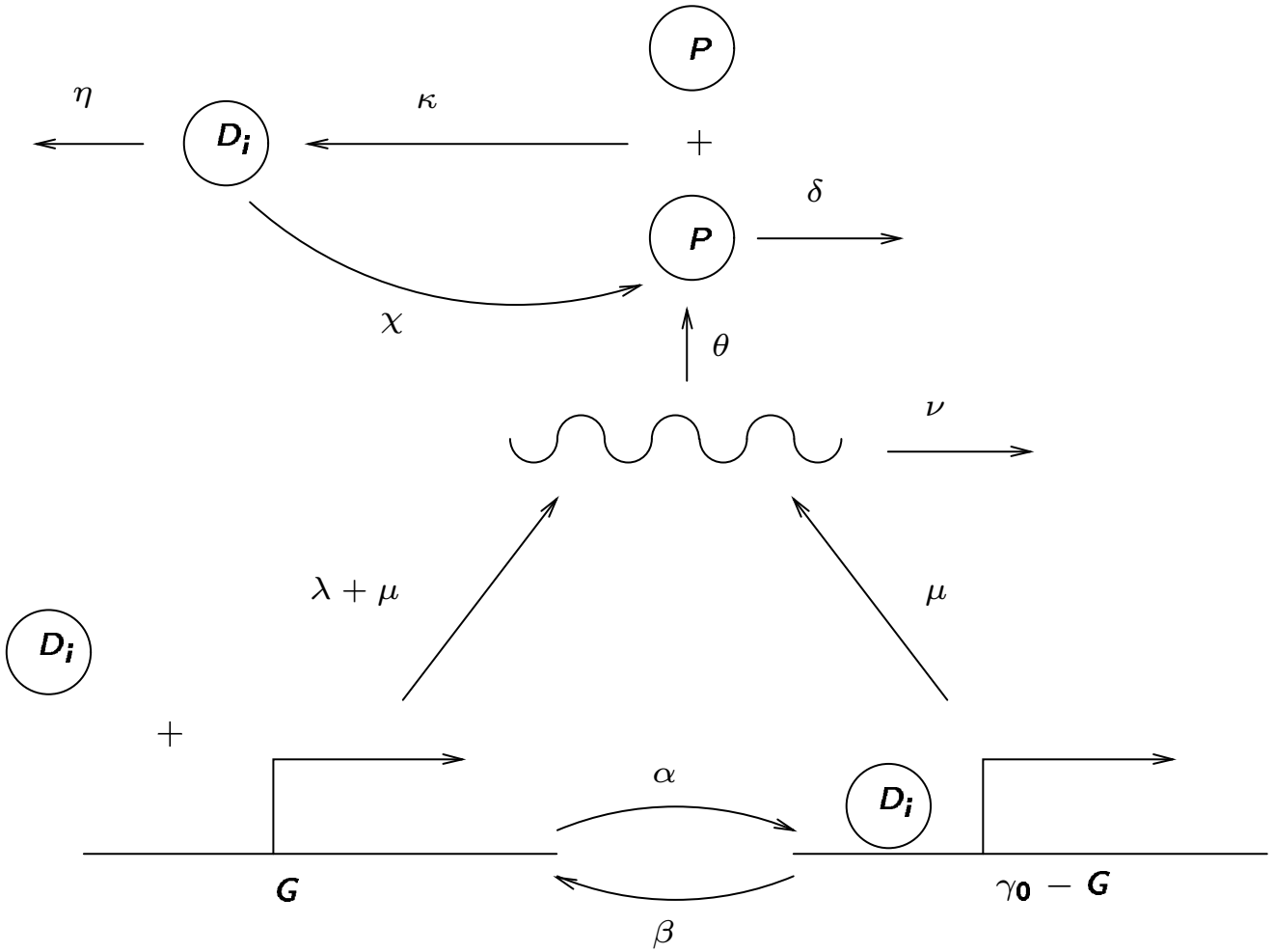
$$
\begin{aligned}
\dot{D}_A(t) &= \theta_A(1 - D_A(t)) - \gamma_A\, D_A(t)\, A(t),\\
\dot{D}_R(t) &= \theta_R\,(1 - D_R(t)) - \gamma_R\, D_R(t)\, A(t),\\
\dot{M}_A(t) &= \alpha'_A\,(1 - D_A(t)) + \alpha_A\, D_A(t) - \delta_{M_A(t)}\, M_A(t),\\
\dot{M}_R(t) &= \alpha'_R\,(1 - D_R(t)) + \alpha_R\, D_R(t) - \delta_{M_R}\, M_R(t),\\
\dot{A}(t) &= \beta_A\, M_A(t) + \theta_A\,(1 - D_A(t)) + \theta_R\,(1 - D_R(t)) - A(t)\,(\cdots),\\
\dot{R}(t) &= \beta_R\, M_R(t) - \gamma_C\, A(t)\, R(t) + \delta_A\, C(t) - \delta_R\, R(t),\\
\dot{C}(t) &= \gamma_C\, A(t)\, R(t) - \delta_A\, C(t).
\end{aligned}
$$

For some less intuitive ranking, Rosenfeld–Gröbner gives even better :

$$
\begin{aligned}
\dot{A}(t) &= f(R(t),\, C(t),\, A(t))\\
\dot{C}(t) &= g(R(t),\, C(t),\, A(t))\\
R(t) &= h(C(t), A(t))
\end{aligned}
$$

Yakoubsohn idea. Differential elimination may provide different sets of coordinates to describe constraint varieties.

Algebraic elimination + Routh–Hurwitz criterion



One gene regulated by a homodimer.

The model equations

The gene state G (γ_0 unbounded, 0 bounded to D_i).
The RNA concentration M.
The protein concentration P.
The homodimer concentration D_i.
Greek letters are parameters.
All variables and parameters are positive apart λ.

$$\begin{aligned}
\dot{G} &= \beta\,(\gamma_0 - G) + \alpha\, D_i\, G,\\
\dot{M} &= (\lambda + \mu)\, G + \mu(\gamma_0 - G) - \nu\, G,\\
\dot{P} &= \theta\, M - \delta\, P - 2\,\kappa\, P^2 + 2\,\chi\, D_i,\\
\dot{D_i} &= \kappa\, P^2 - \eta\, D_i - \chi\, D_i + \beta\,(\gamma_0 - G) + \alpha\, D_i\, G.
\end{aligned}$$

Proving the impossibility of Hopf bifurcation

When a Hopf bifurcation occurs, a stable steady point becomes unstable and a stable limit cycle arises around it (sufficient condition for an oscillation !).
One might look for Hopf bifurcations as follows :

1. compute the steady points
2. apply a variant of the Routh–Hurwitz criterion over each steady point.

The computer algebra way : do not compute the steady points. Apply the Routh–Hurwitz variant modulo the polynomial ideal which defines the steady points. Algebraic elimination.

Remark

The Routh–Hurwitz variant is surprisingly easy to apply.

Proving the impossibility of Hopf bifurcation

When a Hopf bifurcation occurs, a stable steady point becomes unstable and a stable limit cycle arises around it (sufficient condition for an oscillation !).
One might look for Hopf bifurcations as follows :

1. compute the steady points
2. apply a variant of the Routh–Hurwitz criterion over each steady point.

The computer algebra way : do not compute the steady points. Apply the Routh–Hurwitz variant modulo the polynomial ideal which defines the steady points. Algebraic elimination.

Remark

The Routh–Hurwitz variant is surprisingly easy to apply.

Conclusion

This talk showed different applications of computer algebra methods to biological modelling :

- parameters estimation (differential elimination)
- model reduction (differential elimination)
- proving the absence of Hopf bifurcation (algebraic elimination)

Cooperation with numerical methods is mandatory.
Software outside computer algebra systems have a role to play (BLAD libraries).