Finding All Steady States in Biological Models

Jason Zwolak[1], John J. Tyson[2], and Layne T. Watson[1]
[1]Department of Computer Science and [2]Department of Biology
Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0106
jzwolak@vt.edu, Tel: 540-231-5958

Introduction

ODE based models of biological systems are used frequently by theoretical biologists. These systems can have stable and unstable steady states and oscillations. This poster presents a method to find all steady state solutions to a restricted class of ODE models. The right hand sides of the models are restricted to linear combinations of rational functions of variables and parameters. The method converts the steady state equations into a system of polynomials equations and runs POLSYS_PLP [3] to find all the roots of the system of polynomials. All the steady state solutions will be present as roots of the polynomial equations. There may be some roots that do not correspond to steady state solutions because of the conversion. Therefore, the roots of the polynomial equations must be checked in the steady state equations. The stabilities of the steady states are not revealed. This poster explains the methods used and gives the results of an example problem.

As an example, all the steady states were found for a model of frog egg extracts [2]. The steady states were verified independently in XPP [1].
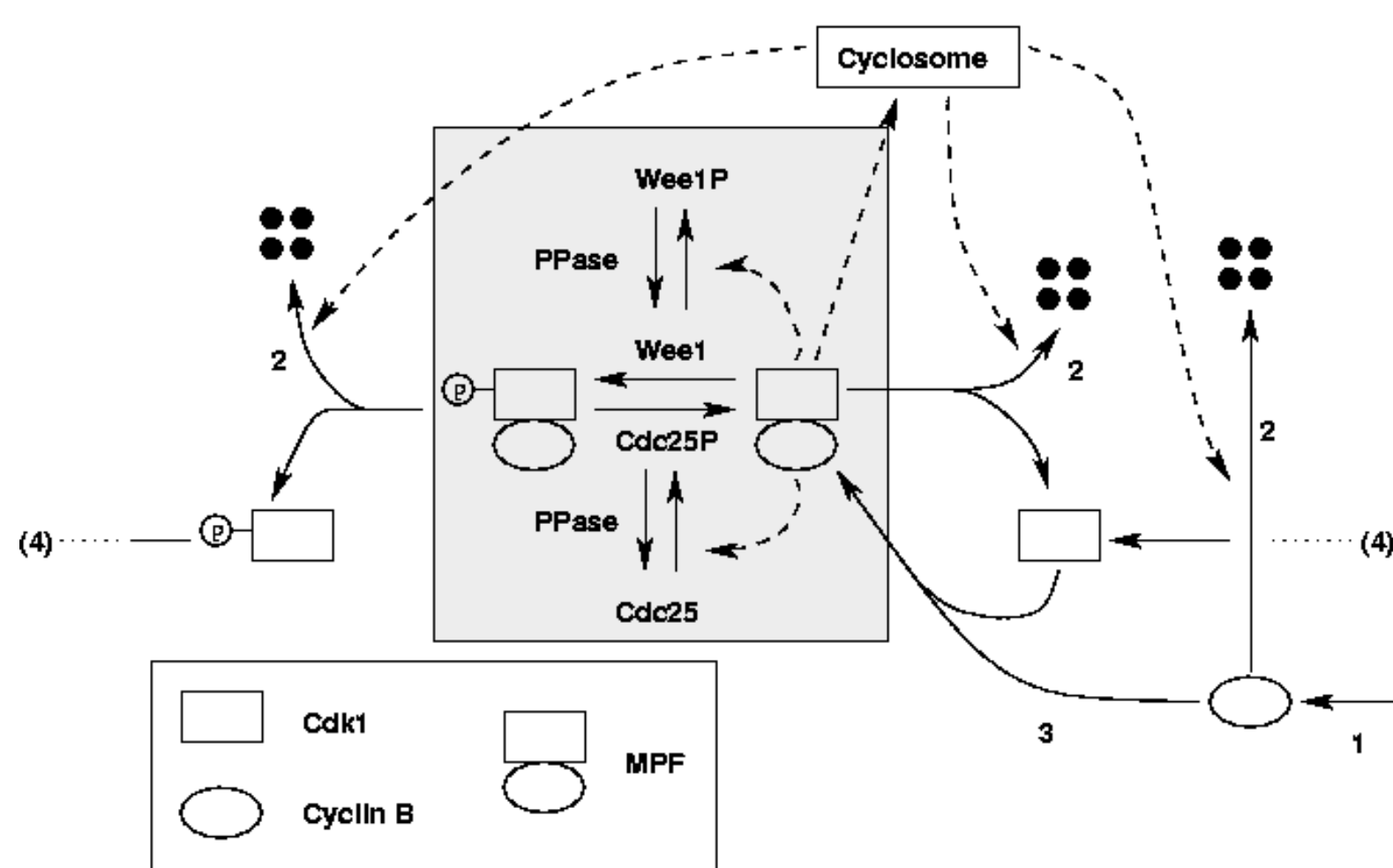

Fig. 1. Network of proteins for the example frog egg model.

Example Problem

Fig. 1 shows the network of the example problem from Marlovits [2]. The equations derived from that network are

$$\frac{dM}{dt} = (v'_d(1-D) + v''_d D)(C_T - M) - (v'_w(1-W) + v''_w W)M,$$

$$\frac{dD}{dt} = v_d\left(\frac{M(1-D)}{K_{md}+(1-D)} - \frac{\rho_d D}{K_{mdr}+D}\right),$$

$$\frac{dW}{dt} = v_w\left(-\frac{MW}{K_{mw}+W} + \frac{\rho_w(1-W)}{K_{mwr}+(1-W)}\right),$$

where M is active MPF, D is active Cdc25, and W is active Wee1.

Method

First, set up the steady state equations by setting the right hand sides equal to zero. Second, multiply each equation by a constructed polynomial to eliminate the denominators. Note here that the steady states are preserved because multiplying a zero will give you a zero. Multiplying through by a polynomial may add extraneous steady states. Third, run POLSYS_PLP to find the roots of the resulting system of polynomial equations. Last, verify the steady states by plugging the answers into the steady state equations.

The example problem yields the steady state equations

$$0 = (v'_d(1-D) + v''_d D)(C_T - M) - (v'_w(1-W) + v''_w W)M,$$

$$0 = v_d\left(\frac{M(1-D)}{K_{md}+(1-D)} - \frac{\rho_d D}{K_{mdr}+D}\right),$$

$$0 = v_w\left(-\frac{MW}{K_{mw}+W} + \frac{\rho_w(1-W)}{K_{mwr}+(1-W)}\right),$$

and the constructed polynomials are

$$1,$$
$$(K_{md}+(1-D))(K_{mdr}+D),$$
$$(K_{mw}+W)(K_{mwr}+(1-W)),$$

respectively. The resulting system of polynomial equations is

$$0 = (v'_d(1-D) + v''_d D)(C_T - M) - (v'_w(1-W) + v''_w W)M,$$

$$0 = v_d(M(1-D)(K_{mdr}+D) - \rho_d D(K_{md}+(1-D))),$$

$$0 = v_w(-MW(K_{mwr}+(1-W)) + \rho_w(1-W)(K_{mw}+W)).$$

POLSYS_PLP was run on the above system of polynomials with constant parameters as given in Marlovits [2]. C_T was varied from –1 to 1 and all the steady states were returned with a step size of 0.002.

Results

Several steady states were verified by plugging the results into the steady state equations. All the steady states checked were verified.

All the steady state values for active MPF were verified by tracking the steady states using XPPAUT [1]. The graphs of the steady states found by the methods described in this poster are in Fig. 2. On top of those steady states are the steady states tracked by XPPAUT. Note that XPPAUT cannot find a steady state. XPPAUT must start with a steady state, and it can then track the steady state over the range of a parameter (i.e., C_T).
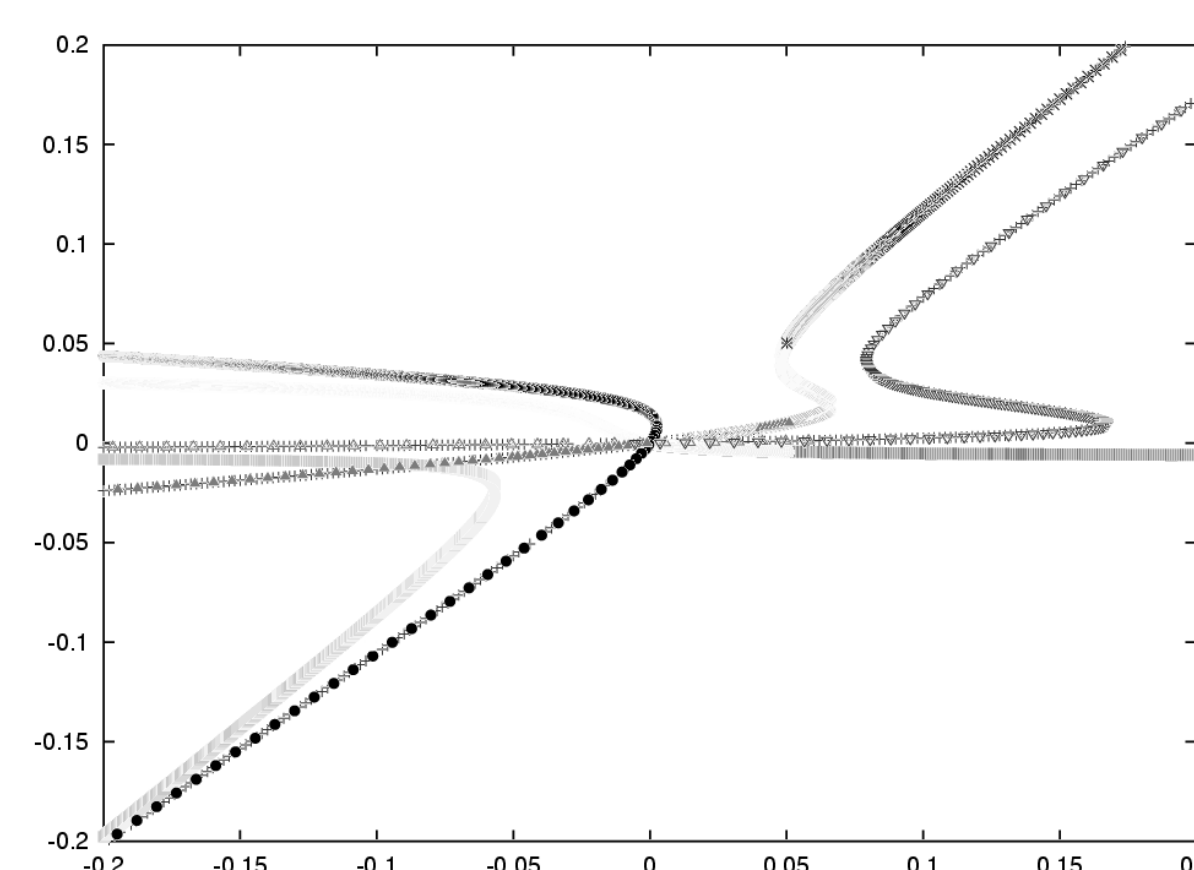

Fig. 2. The values of MPF that solve the steady state equations for a range of total cyclin values.

Conclusion

An algorithm was developed for finding all steady states in a restricted class of ODE based models. The algorithm was tested and verified to work on a test case. The algorithm will be tried on many more real world models and the steady states checked.

This is a revolutionary tool for theoretical biologists. It provides a means to show a model has steady states not previously known or to prove there are no more steady states than the ones known. This tool is very powerful in so far as it allows the theoretician to make powerful predictions that can be tested experimentally.

POLSYS_PLP

POLSYS_PLP finds solutions to systems of polynomial equations using homotopy mapping. POLSYS_PLP creates a system of polynomials with known roots and "morphs" that system into the system specified by the user. The "morphing" is a process of tracking curves (homotopy curves) from the start system to the target system. Each curve represents a solution, therefore the start system must be similar to the target system (i.e., the two systems must have the same number of roots).

References

[1] Ermentrout, B. Simulating, Analyzing, and Animating Dynamical Systems: A Guide to XPPAUT for Researchers and Students. SIAM, Philadelphia, 2002.
[2] Marlovits, G., Tyson, C. J., Novak, B., and Tyson, J. J. Modeling M-phase Control in Xenopus Oocyte Extracts: the Surveillance Mechanism for Unreplicated DNA. Biophys. Chem., 1998, 72, 169-184.
[3] Wise, S. M., Sommese, A. J., and Watson, L. T. POLSYS_PLP: A Partitioned Linear Product Homotopy Code for Solving Polynomial System of Equations. ACM Transactions on Mathematical Software, 2000, Vol. 26, 1:176-200.