INVARIANT MANIFOLDS IN A DYNAMICAL MODEL FOR GENE TRANSCRIPTION

MARTIN CABERLIN *, MICHAEL MACKEY†, AND NILIMA NIGAM ‡

Abstract. We present some recent results concerning stiffness inthe Santillán-Mackey model of the tryptophan operon. In particular, we describe the existence of invariant manifolds in this system, and describe their biological significance.

1. Introduction. In this paper we present a numerical investigation of the Santillán-Mackey model of the tryptophan operon [9]. This negative-feedback system is modelled by four coupled delay differential equations (DDE). Sophisticated algorithms are used to simulate the behaviour of these DDE, and interesting phenomena are observed. These have both provocative biological significance, and suggest experiments which could verify the validity of the theoretical model.

Our numerical investigation revealed that Santillán-Mackey model exhibits *stiffness*; a characterisation of stiffness will be provided. In this particular situation, this phenomenon means that the dynamical system reduces from being four-dimensional (corresponding to four chemical concentrations) to three, and then to two. This dimension reduction has biological implications, which we conjecture upon and would like to investigate further.

Stiff dynamical systems are exceedingly difficult to integrate using standard *explicit methods*, and require very small time-steps in order to maintain numerical stability. The use of adaptive algorithms does not significantly alleviate this problem; the integrator selects an unreasonably small allowable step-size. Heuristically, the time-step is selected to resolve the behaviour of the fastest transient in the system. However, this transient may not be the dominant mode and may be contributing neglibily to the over-all dynamics of the system. The net result is a significant waste of computational effort. A well-known strategy to numerically integrate stiff systems is to use an *implicit method*. In Section 2, we first provide a working definition of stiffness, and then demonstate through simple examples the difference between using explicit and implicit methods.

Delay-differential systems also pose computational challenges, especially to any algorithm where the step-size is adaptively chosen. This is because one may require the value of the already-computed solution, but at a point other than a grid-point. Thus, any integration algorithm must now include a method for approximating solutions at intermediate values of time. We used an adaptive, 2nd-order explicit solver developed by Shampine and Thompson, as well as a fixed-step Runge-Kutta method (RK4 with linear-interpolation).

Given the two broad challenges presented by the Santillán-Mackey model – stiffness and delay – we began our investigation by first setting the delays to zero. This enabled us to compare explicit and implicit algorithms easily, without the added complication of the delays. We observe stiffness in this simplified system, and noted the presence of invariant manifolds. Subsequently, we integrated the full model using the explicit DDE methods, with a very small time-step. We again observed the existence of invariant manifolds. Our current research efforts are focussed on the implementation of a stiff delay-differential solver, which would be most suitable for the investigation of this model.

The organisation of this paper is as follows. We begin with a brief description of the model in question; for further details we refer the interested reader to [9, 8]. In Section 2, we

* Dept. of Mathematics and Statistics, McGill. email:caberlin@math.mcgill.ca
† Center for Nonlinear Dynamics, McGill. email:mackey@cnd.mcgill.ca
‡ Dept. of Mathematics and Statistics, McGill. email:nigam@math.mcgill.ca

describe stiffness and present examples which demonstrate the power of implicit methods. We also briefly describe the DDE solvers we used. In section 3, we present the results of our investigations of the zero-delay system, and then we describe the simulation of the full Santillán-Mackey model. We end this paper with tentative at a biological explanation for the observed phenomena, as well as suggested experiments to verify our findings.

1.1. The Santillan-Mackey Model of the Tryptophan Operon. Since the discovery of DNA as the genetic library of life, there have been extraordinary advances in experimental molecular biology, particularly in the understanding of gene control systems. These have spurred theoretical advances in the mathematical modelling of these systems, with a view to numerical simulation of expensive and time-intensive experiments. In case of the lactose and tryptophan operons, it is particularly remarkable that the veracity of a theoretical model can be checked using the vast amount of experimental data available

In 1965, not long after its discovery, Goodwin proposed the first mathematical model of the tryptophan operon [5]. Bliss *et al*. proposed a more detailed model in 1982, accounting for the system's repression, feedback inhibition and time delays [2]. With more recent experimental data, several other models, most notably the Sinha [12] and Santillan-Mackey [9] models, have become more sophisticated, attempting to explain better the observed phenomena. The tryptophan operon modelled here is a negative feedback system. The operon in question is found in the single chromosome of the bacterium *Escherichia coli* (*E. coli* for short). It consists of five clustered genes required for the production of the amino acid tryptophan, preceded by a short promotor region. The promoter region directs RNA polymerase to bind to DNA and begin synthesizing an RNA molecule. Within the promoter region lies the operator. This is the region to which the tryptophan repressor, *TrpR*, may bind to inhibit the initiation of transcription. Transcription is normally facilitated via the attachment of an mRNA polymerase (mRNAP) molecule to the promoter region. The differential in affinities between *TrpR* and mRNAP for the promoter site are crucial to the dynamics of this system.

The repression is impressive, to say the least – the repressor protein can bind to the promoter region only when *TrpR* is bound by two tryptophan molecules. This binding occurs sequentially and non-cooperatively at two independent sites of the repressor molecule.

Another important feature of the tryptophan operon is feedback inhibition. Although there are five different polypeptides produced in the expression of the *trp* operon, the molecule anthranilate synthase is the most important from a regulatory standpoint as it is the first molecule in the chemical pathway which converts chorismic acid to tryptophan. It is a complex of two *TrpE* and two *TrpD* proteins. Incidentally, these are the first two polypeptides encoded by the *trp* mRNA. Anthranilate synthase is also the molecule that is feedback inhibited by tryptophan. The inhibition takes place when two tryptophan molecules bind to each of the *TrpE* subunits of anthranilate synthase.

From a regulatory point of view, the essential features of the tryptophan operon are repression and feedback inhibition. However, there are many more features which affect the dynamics of the biochemical system, including chemical rate constants, natural delay processes, transcription attenuation, et cetera. These will be discussed in more detail in the following section.

1.2. The Model. In this subsection, we describe the Santillán-Mackey model of the tryptophan operon [9]. Although the model is oversimplified in many ways, it is the most intricate model of the tryptophan operon system available, and is considered sufficiently detailed by Santillán and Mackey to reproduce experimental results.

The system variables and symbols are found in Table 1.1. An explanation for the estimation of the parameters displayed in Table 1.3 can be found in supplemental section B of

TABLE 1.1
System variables and symbols

O	Total operon concentration
O_F	Free operon concentration
M_F	Free mRNA concentration
E	Total anthranilate synthase concentration
E_A	Active anthranilate synthase concentration
T	Tryptophan concentration
R	Total repressor concentration
R_A	Active repressor concentration
P	mRNA polymerase concentration
ρ	Ribosomal concentration
D	mRNA destroying enzyme concentration

TABLE 1.2
Santillán-Mackey Model Equations

$$\dot{O}_F = \frac{K_r}{K_r + R_A(T)}\{\mu O - k_p P[O_F(t) - O_F(t-\tau_p)e^{-\mu\tau_p}]\} - \mu O_F(t) \tag{1.1}$$

$$\dot{M}_F = k_p P O_F(t-\tau_m)e^{-\mu\tau_m}[1 - A(T)] - k_\rho \rho[M_F(t) - M_F(t-\tau_\rho)e^{-\mu\tau_\rho}] - (k_d D + \mu)M_F(t) \tag{1.2}$$

$$\dot{E} = \frac{1}{2}k_\rho \rho M_F(t-\tau_e)e^{-\mu\tau_e} - (\gamma + \mu)E(t) \tag{1.3}$$

$$\dot{T} = KE_A(E,T) - G(T) + F(T, T_{ext}) - \mu T(t) \tag{1.4}$$

$$R_A(t) := R\frac{T(t)}{T(t) + K_t} \tag{1.5}$$

$$A(T) := b(1 - e^{-T(t)/c}) \tag{1.6}$$

$$E_A(E,T) := \frac{K_i^{n_H}}{K_i^{n_H} + T^{n_H}(t)}E(t) \tag{1.7}$$

$$G(T) := g\frac{T(t)}{T(t) + K_g} \tag{1.8}$$

$$F(T, T_{ext}) := d\frac{T_{ext}}{e + T_{ext}[1 + T(t)/f]} \tag{1.9}$$

[9]. We should note that the parameters have been estimated from real experimental data, as opposed to curve fitting experiments with appropriately chosen parameters.

The Santillán-Mackey model considers four independent variables. These are the concentration of free operon (O_F), the concentration of *trp* mRNA molecules with free *TrpE*-related binding sites (M_F), and the concentrations of anthranilate synthase (E) and tryptophan (T).

The repression chemistry modelled here is assumed to be a first-order reversible process with forward and backward rate constants k_r and k_{-r}, respectively.

Relevant to the binding of mRNAP to the promoter region are the mRNAP concentration P and the rate constant k_p. After binding a free operon, the DNA-mRNAP complex undergo a series of chimeric changes before assembling the first mRNA nucleotide. After a time τ_p, the mRNAP has moved far enough along the DNA to free the operon, allowing another mRNAP or *TrpR* molecule to bind it.

This gives us the information required to model the dynamics of O_F. It is important to note that the binding of repressor molecules to free operons occurs at a rate roughly two orders of magnitude larger than that of mRNAPs. This is made evident by the estimated values of k_r, k_{-r} and k_p, and justifies a quasisteady-state assumption for the repression process.

The next process of the system is the translation of the mRNA into polypeptide. This

TABLE 1.3
Model parameters

$R = 0.8\,\mu\text{M}$	$O = 3.32 \times 10^{-3}\,\mu\text{M}$
$P = 2.6\,\mu\text{M}$	$k_{-r} = 1.2\,\text{min}^{-1}$
$\rho = 2.9\,\mu\text{M}$	$k_r = 460\,\mu\text{M}^{-1} \cdot \text{min}^{-1}$
$\tau_p = 0.1\,\text{min}$	$k_{-i} = 720\,\text{min}^{-1}$
$\tau_m = 0.1\,\text{min}$	$k_i = 176\,\mu\text{M}^{-1} \cdot \text{min}^{-1}$
$\tau_\rho = 0.05\,\text{min}$	$k_{-t} = 2.1 \times 10^4\,\text{min}^{-1}$
$\tau_e = 0.66\,\text{min}$	$k_t = 348\,\mu\text{M}^{-1} \cdot \text{min}^{-1}$
$\gamma = 0\,\text{min}^{-1}$	$k_p = 3.9\,\mu\text{M}^{-1} \cdot \text{min}^{-1}$
$k_d D = 0.6\,\text{min}^{-1}$	$k_\rho = 6.9\,\mu\text{M}^{-1} \cdot \text{min}^{-1}$
$n_H = 1.2$	$\mu = 1.0 \times 10^{-2}\,\text{min}^{-1}$
$b = 0.85$	$c = 4.0 \times 10^{-2}\,\mu\text{M}$
$K_g = 0.2\,\mu\text{M}$	$g = 25\,\mu\text{M} \cdot \text{min}^{-1}$
$e = 0.9\,\mu\text{M}$	$d = 23.5\,\mu\text{M} \cdot \text{min}^{-1}$
$f = 380\,\mu\text{M}$	$K = 126.4\,\text{min}^{-1}$
$K_r = k_{-r}/k_r$	$K_i = k_{-i}/k_i$
$K_t = k_{-t}/k_t$	

begins even during transcription, and since *TrpE* is the first protein encoded on the operon, it is the first to be translated. Recall, M_F is the concentration of free *TrpE* -related ribosome binding sites. Transcriptional attenuation is a mathematical complication, but a natural regulatory wonder. When there is a large concentration of charged tRNA^{Trp}, the premature termination of transcription becomes more likely. The amount of charged tRNA^{Trp} depends on the tryptophan concentration, so the model assumes the attenuation $A(T)$ is a function of T. There is also a delay τ_m associated with the time taken for an mRNAP to produce a functional *TrpE*-related ribosome binding site. With these considerations, the production rate of free *TrpE*-related ribosome binding sites is equal to the rate of mRNAPs that bound free operons a time τ_m ago $[k_p P O_F(t - \tau_m)]$ times a dilution factor for the exponential growth of the culture, times the probability that a newly bound mRNAP will produce a functional mRNA $[1 - A(T)]$. Yet there remain further considerations for M_F.

After a ribosome binds a free *TrpE*-related binding site, and before assembling the first peptide bond, the mRNA-ribosome complex must undergo conformational changes. Santillán and Mackey assume this takes place with a rate proportional to M_F and to the ribosomal concentration ρ with rate constant k_ρ. The ribosome binding site is freed a time τ_ρ later. In this time, the free mRNA concentration rate increases by an amount equal to the rate of ribosomal binding and translation initiation a time τ_ρ ago, times the dilution that occurred during that time. The free mRNA concentration rate decreases by the rate of ribosome binding and initiation of translation at time t. These account for the second term of (1.2).

This completes our discussion of the Santillán-Mackey model of the tryptophan operon. The system has a unique steady state which is stable [9], but from this system of DDE, we are unable, by simple examination, to say much further regarding the dynamics. However, several questions remained unanswered. Does the system rapidly attain its equilibrium point? Are all the system variables considered equally active throughout the reaction? Motivated by these, and similar, questions we used several algorithms to simulate the dynamics of the system. This led to the discovery of the presence of stiffness. We thus digress briefly to discuss stiff problems, and the computational complications that arise as a consequence. An analysis of the interesting numerical properties of the model are discussed in Sections 3 and 4.

2. Numerical stiffness, and DDE solvers. We shall begin this section with a sample dynamical system which exhibits stiffness, and seek to convince the audience that merely

using higher-order or adaptive step-size algorithms does not alleviate the problem, as long as these methods are explicit. This motivates the discussion on numerical stiffness, and we describe well-known fixes for the problem. Readers who are familiar with the issues of stiffness and delay-differential solvers are encouraged to skip to Section 3.

To begin with, we are interested in computing the solution $y(x)$ to the system of ODE

$$\begin{aligned} y'(x) &= f(x, y(x)), \\ y(a) &= \alpha, \quad x \in [a, b]. \end{aligned} \tag{2.1}$$

We note that the IVP may be a single equation, or a system. In the latter case the solution $y(x)$ is a vector. We also wish to study delay differential equations (DDE) of the form

$$\begin{aligned} y'(x) &= f(x, y(x), y(x - \tau_j)), \quad x \in [a, b], \\ y(a) &= \alpha, \quad y(x) = g(x) \text{ for } x < a. \end{aligned} \tag{2.2}$$

where $\tau_j \geq 0$, $j = 1, \ldots, M$. In Henrici's notation [6], the recursion used to advance the solution of (2.1) from x_n to x_{n+1} is

$$y_{n+1} = y_n + h_n \Phi(x_n, y_n, h_n), \tag{2.3}$$

where y_n is the numerical approximation to the exact solution $y(x_n)$. The function Φ is called the *increment function* of the method. For example, when $\Phi(x_n, y_n) = f(x_n, y(x_n))$, the algorithm obtained is the usual Forward Euler. Note that for DDE, the increment function may also depend on the history, *ie.* the solution at values $x < x_{n+1}$.

2.1. Stiffness- what is it?. We begin this section with a simple example to illustrate the phenomenon of "stiffness".

EXAMPLE 1. *Consider the IVP*

$$y' = -10000y - e^t + 10000, \qquad y(0) = 1, \qquad t \in [0, 5]. \tag{2.4}$$

The exact solution of this problem is given by

$$y(t) = -\frac{1}{10001}e^t + 1 + \frac{1}{10001}e^{-10000t}.$$

We see from the expression above that the last exponential term $\frac{1}{10001}e^{-10000t}$ should become negligible in comparison to 1, indeed, when $t \approx .072$, this term contributes less than 10^{-15} to the overall computation.

In order to compare the relative efficiency of various algorithms in the example above, we required that the relative error of the computed solution stayed within 10^{-6} during the interval $[0, 5]$. For each algorithm, we reduced the step-size or error tolerance until the desired accuracy was achieved. We present the step-size used, as well as the number of time-steps taken to integrate over the interval $[0, 5]$. The algorithms used were:

1. **Forward Euler:** The simplest explicit integrator possible, low-order, and fixed step-length.
2. **ODE23:** Matlab's adaptive step-size, second-order algorithm
3. **RK4:** A fixed step-length, explicit fourth-order algorithm
4. **ODE45:** Matlab's adaptive step-size, fourth-order algorithm

In Table 2.1, we compile some results using various standard explicit numerical algorithms. All *explicit solvers*, with or without variable step size, work in the same way. That is, the computation of the approximate solution at x_{n+1} depends on the values of the solution already

algorithm	no. of steps	max step	cputime	Y(5)
Forward Euler	100000	5×10^{-5}	339.8	.9851609
ode23	19902	3.803×10^{-4}	67.37	.9851602
RK4	50000	1×10^{-4}	91.31	.9851616
ode45	60273	1.088×10^{-4}	109.41	.9851602

TABLE 2.1

Results of using explicit numerical methods on example 1.The true solution at $t = 5$ is $y(5) = .9851601681\ldots$

computed. All these experiments were conducted in MATLAB, on a Linux workstation with a 1.3GHz Athlon processor. The cputimes reported are in seconds. The algorithms would blow up, yielding NaN, if the stepsizes were much larger. We see that the algorithms are chosing a step size small enough to accurately resolve the fastest transient - in other words, the insignificant term $\frac{1}{10001}e^{-10000t}$ governs the choice of step size!! This is neither a problem of accuracy, nor of adaptivity. The poor performance of *all* these algorithms on the example points to a deeper issue. Merely using higher and higher order algorithms, even if they are adaptive, does not suffice; the dynamical system 2.4 exhibits behaviour that confounds each of these methods. In particular, all of these algorithms have *stability* problems; if the step size is not excruciatingly small, the computed solutions "blow up". We describe this system as **stiff**.

Heuristically, one useful definition of stiff dynamical systems is : A system in which one or several parts of the solution vary rapidly (as an exponential, e^{-kx} for example, with k large), while other parts of the solution vary much more slowly (as an oscillator, or linearly) is characterized as a stiff system. These are prevalent in the study of damped oscillators, chemical reactions and electrical circuits. Essentially, the derivative or parts thereof change very quickly.

The precise definition of numerical stiffness is difficult to state; however, the phenomenon is one that is widely observed while numerically approximating solutions of differential equations (as seen above). For our purposes, we adopt the following definition of stiffness due to Lambert ([7]):

DEFINITION 1. *If a numerical method with a finite region of absolute stability, applied to a system with any initial conditions, is forced to use in a certain interval of integration a step length which is excessively small in relation to the smoothness of the exact solution in that interval, then the system is said to be* **stiff** *in that interval.* This definition allows us to concede that the stiffness of a system may vary over the interval of integration. We must clarify the term 'excessively small'. In regions where the fast transient is still alive, we expect the step length to remain small. However, in regions where the fast (or slow) transients have died, we would expect the step length to increase to a size reasonable to the problem.

A "cure" for stiffness is the use of implicit algorithms. These are algorithms where the approximation Y_{n+1} to the true solution y at $t_n + h$ is given by

$$Y_{n+1} = Y_n + h\Phi(Y_{n+1}, t),$$

where Φ is chosen according to the system being solved, the accuracy desired, etc. Note that the update Y_{n+1} is now the solution of a (typically) non-linear system. Thus, each step of our algorithm requires a nonlinear solve, an expensive proposition. However, since the domain of stability of implicit algorithms is much larger than that of explicit algorithms, most state-of-the-art stiff solvers employ adaptive implicit algorithms. For example, the built-in MATLAB solvers `ode23s` and `ode23tb` are implicit Runge-Kutta methods with variable step size control. The saving in terms of computational time is well worth the effort of solving the

algorithm	no. of steps	max step	cputime	Y(5)
Backward Euler	2500	.002	.11	.9851898
ode23s	179	0.1094	0.61	0.9851601

TABLE 2.2
Some implicit algorithms used to compute example 1. Compare these with similar explicit algorithms in Table ().

nonlinear problem. To drive this message home, let us return to the example we began this section with, and use a couple of implicit solvers:

1. **Backward Euler :** the implicit analog of Forward Euler, this method requires a fixed step-size.
2. (**MATLAB's ODE23s:) this method uses an adaptive step-size, and is a second-order algorithm**

Table 2.1 summarizes the performance of these implicit algorithms. Despite the nonlinear solve, these algorithms outperform those in Table 2.1 by a couple of orders of magnitude both in terms of number of time-steps, and cputime taken.

2.2. DDE solvers. The built-in MATLAB solvers `ode23` and `ode45` are explicit Runge-Kutta methods with variable step size. These solvers control the error using the lower order formula (`2` and `4`, resp.), and advance the integration using the higher order formula (`3` and `5`, resp.). The variable step size control allow the methods to advance very quickly in regions where $f(x, y)$ changes slowly. Variable step size ERK are very efficient for what are called *non-stiff* ODE. For *stiff* ODE, however, we must examine the implicit methods.

2.3. Delay Differential Equations (DDE). New methods exist for solving stiff and non-stiff delay differential equations (2.2),

$$y'(x) = f(x, y(x), y(x - \tau_j)), \quad x \in [a, b],$$
$$y(a) = \alpha, \;\; y(x) = g(x) \text{ for } x < a,$$

where $\tau_j \geq 0, j = 1 \ldots, M$. These methods are clever extensions of the ideas used to solve ODE. The key difference between these algorithms and those for ODE is the need to approximate intermediate values of the solution; having obtained the approximate solution $S(x_n) \equiv y_n$ and $S(x_{n-1}) \equiv y_{n-1}$, one may require $S(x)$ for $x \in (x_{n-1}, x_n)$. This is computed using interpolation, which introduces an error.

2.3.1. Fixed step-size DDE Solver. For the purpose of this study, we implemented a fixed-step DDE solver based on the four stage Runge-Kutta method of order four. We call this solver `RK4d`. The algorithm is designed to approximate the solution of (2.2) with positive, constant delays τ_j and constant history function, $g(x) \equiv c$.

The explicit algorithm `RK4d` uses a constant step size, and cannot advance the integration if the step size is larger than any of the delays, $h > \tau_j$. Hence, the step length must be chosen smaller than $\tau = \min_j \{\tau_j\}$.

Any delay solver must interpolate the solution between mesh points. For a simple discourse, we assume there is only one delay τ. Suppose the integration has reached x_n and we require the solution at $(x_{n+1} - \tau) \in [x_{n-k}, x_{n-k+1}]$, where the numerical solution is already known at the mesh points x_{n-k} and x_{n-k+1}. To estimate the solution at the intermediate steps of the RK4 method, $y(x_{n+1} + \frac{h}{2} - \tau)$, linear interpolation is used. More specifically, we define $z(x) = y(x - \tau)$ and the numerical approximation $z_n = y_{n-k} \approx y(x_n - \tau)$. Then the intermediate quantity can be approximated linearly by $y(x_{n+1} + \frac{h}{2} - \tau) \approx \frac{1}{2}(z_n + z_{n+1})$.

In the notation of Definition **??** suitably adapted for delays, we write

$$\begin{aligned} k_1 &= f(x_n, y_n, z_n) \\ k_2 &= f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1, \frac{1}{2}(z_n + z_{n+1})) \\ k_3 &= f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2, \frac{1}{2}(z_n + z_{n+1})) \\ k_4 &= f(x_n + h, y_n + k_3, z_{n+1}) \\ y_{n+1} &= y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) = y_n + h\Phi(x_n, y_n, z_n) \end{aligned} \tag{2.5}$$

We prove this algorithm is order 2; for details please see ([4]).

2.3.2. An adaptive-step DDE solver. The recently developed MATLAB solver, `dde23`, of Shampine and Thompson, uses an explicit Runge-Kutta method with variable step size control and dense output to integrate ordinary and delay differential equations [10]. It approximates solutions of (2.2) with constant delays τ_j such that $\tau = \min(\tau_1, \ldots, \tau_k) > 0$. The discussion in this section is based on the technical report of Shampine and Thompson [10]. The solver `dde23` is based closely on the ODE solver `ode23`. It implements a triple of s stages involving three formulas - the Runge-Kutta triple BS(2,3) of Bogacki and Shampine [11, 3]. For the general advancement, we have an approximation y_n to the true solution $y(x_n)$ and wish to compute an approximation at $x_{n+1} = x_n + h_n$. For $i = 1, \ldots, s$, the stages $f_{ni} = f(x_{ni}, y_{ni})$ are defined in terms of $x_{ni} = x_n + c_i h_n$ and

$$y_{ni} = y_n + h_n \sum_{j=1}^{i-1} a_{ij} f_{nj}. \tag{2.6}$$

The approximation used to advance the integration is

$$y_{n+1} = y_n + h_n \sum_{i=1}^{s} b_i f_{ni} = y_n + h_n \Phi(x_n, y_n). \tag{2.7}$$

The exact solution nearly satisfies this formula, save the local truncation error, lte_n,

$$y(x_{n+1}) = y(x_n) + h_n \Phi(x_n, y(x_n)) + lte_n.$$

For sufficiently smooth f and $y(x)$ this error is $O(h^{p+1})$. The triple also includes the lower order formula,

$$y^*_{n+1} = y_n + h_n \sum_{i=1}^{s} b^*_i f_{ni} = y_n + h_n \Phi^*(x_n, y_n), \tag{2.8}$$

whose local truncation error, lte^*_n, is $O(h^p)$. This formula is used to select the step size. The third formula is a continuous extension of the first one (2.7),

$$y_{n+\sigma} = y_n + h_n \sum_{i=1}^{s} b_i(\sigma) f_{ni} = y_n + h_n \Phi(x_n, y_n, \sigma),$$

where the coefficients $b_i(\sigma)$ are polynomials in σ, so that the formula represents a polynomial approximation to $y(x_n + \sigma h_n)$ for $0 \le \sigma \le 1$. The extension gives the value y_n when $\sigma = 0$

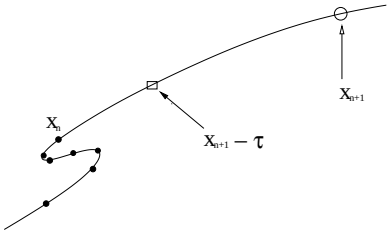

FIG. 2.1. *dde23 when* $h > \tau_j$.

and y_{n+1} when $\sigma = 1$. We assume that the order of the extension is the same as that of the formula, that is $O(h^{p+1})$. The local truncation error of the continuous extension is given by

$$y(x_n + \sigma h_n) = y(x_n) + h_n \Phi(x_n, y(x_n), \sigma) + lte_n(\sigma).$$

All delay solvers must distinguish the situations where the step size is smaller than the smallest delay and where the step size is bigger than at least one of the delays. We let $S(x)$ be the already available approximation to the solution $y(x)$ for all $x \leq x_n$ (recall, for $x \leq a$, we have $S(x) = g(x)$). First, assume $h_n \leq \tau$. Here, all the $x_{ni} - \tau_j \leq x_n$ and all the formulas are explicit. After taking the step to x_{n+1}, the continuous extension is used to define $S(x)$ on $[x_n, x_{n+1}]$ as $S(x_n + \sigma h_n) = y_{n+\sigma}$. One can prove, under suitable hypotheses, convergence as $h_n \to 0$.

The second situation arises when $h_n > \tau_j$ for some j. In this case, the history term falls in the interval of the current step, where it is not yet known (see Figure 2.1). Here, all of the formulas are implicit, and are evaluated by iteration. Up to x_n, $S(x)$ is known, and we must extend its definition to $(x_n, x_n + h_n]$. The resulting function is called $S^0(x)$. A typical stage of the iteration begins with the approximation $S^m(x)$. The next iterate is computed with the explicit formula

$$S^{m+1}(x_n + \sigma h_n) = y_n + h_n \Phi(x_n, y_n, \sigma; S^m(x)). \tag{2.9}$$

One can prove that this is a contraction map for h_n small enough. The `dde23` code chooses $S^0(x)$ to be the constant y_0 for the first step. After the first step, the continuous extension of the preceding step is used as $S^0(x)$ for the current step. The step size adjustment algorithms of `dde23` are those of `ode23` changed to deal with the implicit formulas. The convergence test for the iteration is that $\|y_{n+1}^{m+1} - y_{n+1}^m\|$ is no more than one tenth the accuracy required of y_{n+1}. Since each iteration is as expensive as an explicit step, any h_n with $\tau < h_n < 2\tau$ is reduced to τ to make the formulas explicit. Up to five iterations have been allowed when $h_n \geq 2\tau$. If convergence is not attained, h_n is halved and the step repeated. The solver does not crash due to convergence failures because it will eventually reduce the step size until all the formulas are explicit.

3. Numerical simulations of the Santillan-Mackey model. Several exciting results were obtained for the Santillán-Mackey model with regard to stiffness and invariant manifolds. We found Lambert's Definition 1 suitable for the delay-free Santillán-Mackey system.

It also applied well to the full delay system. We were able to compare the performance of two explicit DDE solvers, `dde23` and our RK4-based delay solver, `RK4d`. Our current research effort is directed at implementing a stiff delay-differential solver which will be suitable for this, and other, biological systems.

The observations for sets of experiments – the models with and without delay – are similar. Using an explicit integrator, the time step necessary to compute the solution seems

too small, even in regions where the fast transient has died. We verified this observation in the case of zero delay by comparing results with an implicit solver. The difference in computational effort between the implicit and explicit methods bears out our conjecture.

Another observation that suggests stiffness in the Santillan-Mackey model is the presence of a fast transient manifold and a slow invariant manifold. After a very rapid transitional stage, the dynamics unfold on a two dimensional manifold on which there is a functional relationship between two pairs of variables. We found that the equations of the invariant manifold depend on the parameter T_{ext}. We are able to fit the numerical data and can a reasonable estimate of the invariant manifold for a large interval of T_{ext}. This is particularly useful as it gives predictions of the model which can be tested by experimentalists. In particular, we invite comment from the experimental community regarding the existence of these meta-stable states for the concentration variables. Given the starting T_{ext} values, we are able to predict the nature of these invariants; an excellent verification of the model would follow if these predicted invariants are found in practice.

3.1. The algorithms used. The *explicit* methods we prepared for this work using MATLAB are the Euler method, the fourth order Runge-Kutta method (`RK4`) , and the built-in MATLAB solvers `ode45`, .

The delay solvers we used were the fixed-step `RK4d`and Shampine and Thompson's adaptive-step `dde23` [1, 10].

The *implicit* methods used in this study were the built-in MATLAB solvers `ode23s`, `ode23tb` , [1]. The source code is available from MATLAB .

All of these are *one-step* methods. One-step methods compute the approximation at x_{n+1} using only the value of the solution at the immediately preceding mesh point x_n. There also exist *multistep* methods whose computation at x_{n+1} relies on the solution at more than one of the previous mesh points.

3.2. The zero-delay setting. We present first our observations on the Santillan-Mackey system in the case when all four delays are set to zero. In order to refer to the following numerical experiment, which is conducted to simulate Figure 2 of [9], we define

EXPERIMENT 1. *The model is run with the parameters as in Table 1.3, experimentally determined initial conditions in Table 3.1 and* $T_{ext} = 400\overline{T}$. *The system is solved numerically to obtain new steady state values for the variables,* S_0. *With* S_0 *as the new initial conditions, the experiment is re-executed with* $T_{ext} = 0$ *and the same parameters as above. The system is then allowed to reach its steady state.* Initially, setting $T_{ext} = 400\overline{T}$ simulates placing the bacteria in a minimal plus tryptophan medium. The second run with $T_{ext} = 0$ simulates the transfer of the bacteria from the minimal plus tryptophan medium to a minimal medium.

Santillán and Mackey visualize the response of the operon by plotting the enzyme activity $KE_A(t)$ as a function of time; this function represents the number of tryptophan molecules produced per unit time [9]. We shall do the same. Variations on Experiment 1 were performed in order to better understand the dynamics of the system. These shall be discussed below.

TABLE 3.1
Steady State values of the model in a minimal medium without tryptophan

$\overline{O_F}$	$1.54 \times 10^{-4}\,\mu$M
$\overline{M_F}$	$3.78 \times 10^{-4}\,\mu$M
$\overline{E}$	$0.378\,\mu$M
$\overline{T}$	$4.1\,\mu$M

Throughout our experiments, we attempted to use similar families of solvers, ie, we

employ the adaptive solver `ode23`, the adaptive stiff solver `ode23s`, `ode23tb` and the adaptive delay solver `dde23`. We also make use of `ode45`, which is a solver often considered a good 'first try for most problems' [1]. Each of these solvers is used with the default MATLAB tolerances, RelTol $= 10^{-3}$ and AbsTol $= 10^{-6}$, unless otherwise noted. In a biological setting, where experimental errors are relatively large (as opposed to stringent error estimates necessary for engineering purposes) these crude tolerances suffice. We also allowed the solvers to determine a suitable initial step size and maximum step size, unless otherwise noted.

We begin by describing our results for the Santillán-Mackey model without delay, and then proceed to the full delay system.

$$\dot{O}_F = \frac{K_r}{K_r + R_A(T)}\mu O - \mu O_F(t) \tag{3.1}$$

$$\dot{M}_F = k_p P O_F(t)[1 - A(T)] - (k_d D + \mu) M_F(t) \tag{3.2}$$

$$\dot{E} = \frac{1}{2} k_\rho \rho M_F(t) - (\gamma + \mu) E(t) \tag{3.3}$$

$$\dot{T} = K E_A(E, T) - G(T) + F(T, T_{ext}) - \mu T(t) \tag{3.4}$$

With the delays set to zero, the Santillán-Mackey model 1.1 - 1.9 simplifies to 3.1 - 3.4 with 1.5 - 1.9 unchanged. We study this system with the same parameter values as are reported in Table 1.3, modulo the parameters corresponding to delay. The justification for examining the above system is that the dynamics are very similar to the full Santillán-Mackey model, as is seen Figure 3.1. Qualitatively, the zero-delay system attains its steady state earlier than the full delayed system. These plots are typical of the experiment described above.

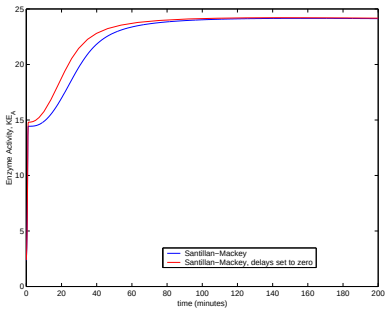

FIG. 3.1. *Comparison of Santillán-Mackey model with and without delays*

3.2.1. Characterizing stiffness in the zero-delay system. We will now present the data accumulated while studying the system 3.1 - 3.4 which suggests genuine stiffness in this system. The simplest method for detecting stiffness numerically is to apply both an explicit code and a similar implicit method for stiff systems, and observe the results.

First we ran Experiment 1 with `ode45`. Much like the full model, the steady state is plainly visible after integrating for 200 minutes. The solver `ode45` takes 335 steps to accomplish this. The time step starts out fairly small at about 0.15, about which it oscillates as low as ~ 0.0025 until the initial fast transient has died after about 15 steps. From there the time step increases slowly to about 1, where it oscillates until the end of the integration (see Figure 3.2). This moderate value of $h = 1$ may seem large. However, after $t = 80$ the

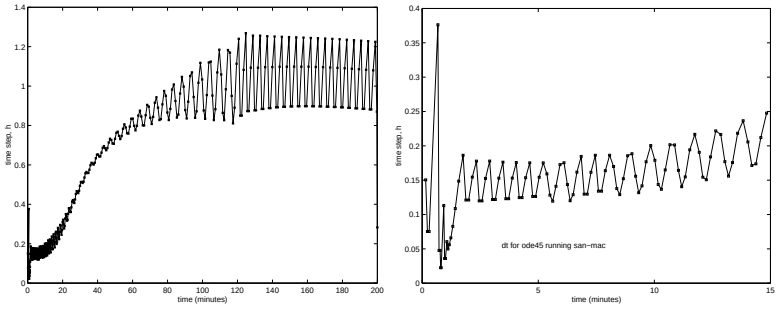

FIG. 3.2. left*: Time step of the zero delay system using* `ode45`. right*: Close up of the first 50 time steps.*

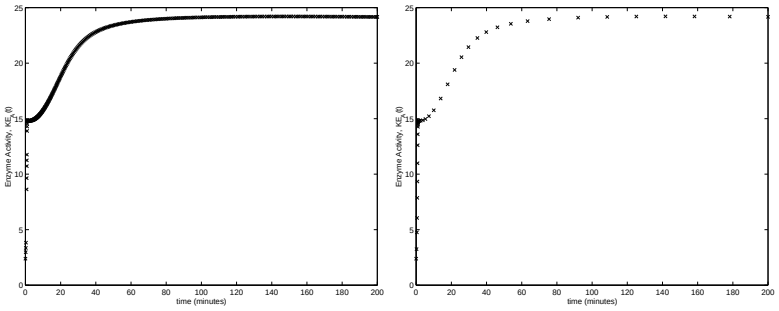

FIG. 3.3. *Zero-delay system,* left*:* `ode45`*,* $N = 335$ right*:* `ode23tb`*,* $N = 45$.

system is very near its constant steady state, and an integrator should not be expending much computational effort here. The small step sizes and gross effort of `ode45` can be seen by the density of crosses in Figure 3.3. Performing a little worse, although similarly, is `ode23` taking 434 steps. The behaviour of the time step is also similar to `ode45`. Compare this to the stiff solver `ode23tb`, which takes only 45 steps to integrate the system. The time step for `ode23tb` starts out at about 0.2, then decreases to about 0.02 around $t = 1$, where the solution hits the first invariant manifold (to be discussed). After that, the time step increases to its maximum value just under 22 (see Figure 3.4). This is clearly quite different behaviour from `ode45` and `ode23`, and certainly more desirable.

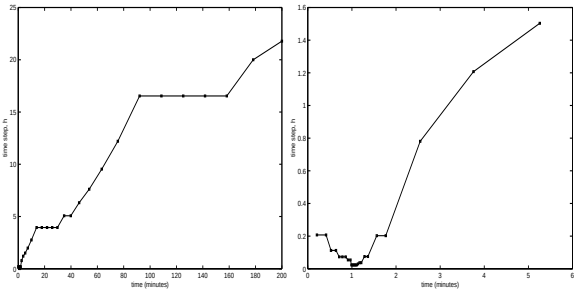

FIG. 3.4. left*: Time step of the zero-delay system using* `ode23tb`. right*: Close up of the first 25 time steps.*

3.2.2. Invariant manifolds in the zero-delay system. Invariant manifolds were discovered in both the zero delay and full Santillán-Mackey models. We will discuss these results

more fully with respect to the full model, however we give a brief description for the zero-delay situation.

After only 2.5 minutes we observe the variable $T(t)$ decrease from its initial value $T(0) \approx 16.8\,\mu$M to $T(2.5) \approx 0.291\,\mu$M. In the time T changes by two orders of magnitude, the other variables change by very little (see Table 3.2). At this point, $E(t)$ and $T(t)$ become linearly related, taking the system down from dimension four to dimension three. A further decrease in dimension occurs around $t = 46$ minutes, when $O_F(t)$ and $M_F(t)$ become linearly related. The system quickly moves from four dimensions to three and then again down to two. The dynamics continue to evolve on this two dimensional manifold until the solution reaches its steady state. We found that the linear relationship between the respective variables depends on T_{ext}, and were able to fit this dependence well for the full system.

TABLE 3.2
Comparison of initial values of the variables and the values at $t = 2.5$ minutes

	$t = 0$	$t = 2.5$
$\overline{O_F}$	$4.89 \times 10^{-5}\,\mu$M	$6.93 \times 10^{-5}\,\mu$M
$\overline{M_F}$	$1.21 \times 10^{-4}\,\mu$M	$1.41 \times 10^{-4}\,\mu$M
$\overline{E}$	$0.122\,\mu$M	$0.122\,\mu$M
$\overline{T}$	$16.8\,\mu$M	$0.291\,\mu$M

An important observation with respect to the invariant manifolds is that `ode45` has just as much difficulty to integrate the solution after reaching the first invariant manifold. It takes 313 steps to integrate from here. Even after the fast transient has died, using the initial conditions in the second column of Table 3.2, `ode45` still takes step lengths smaller than the smoothness of the solution would suggest. This is somewhat reminiscent of Example **??** of Chapter **??**, where the fast transient was not even present in the general solution, yet still dictated the step size.

3.2.3. The Source of Stiffness. In our study, we have determined that one source of stiffness in the zero-delay system lies in the parameter T_{ext}. We varied the parameter $T_{ext}/\overline{T}$ from 0 to 250 and recorded the number of steps required of `ode45`. The fascinating result is shown in Figure 3.5. It is clear from this plot that the stiffness of the system is very much

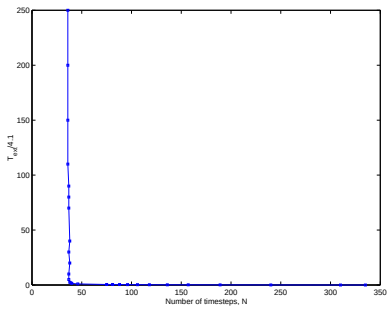

FIG. 3.5. *Number of time steps required of `ode45` versus $T_{ext}/\overline{T}$.*

affected by the value of $T_{ext}/\overline{T}$. The system becomes much less stiff, even not stiff at all, when the ratio $T_{ext}/\overline{T}$ moves just slightly from zero. At $T_{ext}/\overline{T} = 0.5$, the system is solved in 75 steps, and at $T_{ext}/\overline{T} = 1.5$, the system is solved in 40 steps and is effectively not stiff at all! We should note that there is a small concession to be made here. It is that the systems

were all run with the same time span $t \in [0, 200]$. This does not allow enough time for each system (*ie.* with different T_{ext}) to reach its steady state. We maintain that the comparison strongly suggests that T_{ext} is a source of stiffness in the system, and we note that running these experiments to their steady states produce *very* similar results, particularly since the system is more stiff near $t = 0$, as is suggested below.

We also tried to determine whether there was a time interval where the solution was more stiff than others. This is a little more difficult, and the results less clear. We ran Experiment 1, changing the end point of integration. That is, we integrated from $t \in [0, T_f]$ and varied T_f from 1/10 minutes to 200 minutes. The slope of the plot in Figure 3.6 is initially large, which implies that the system is more stiff at the beginning of the integration, between $t = 0$ minutes and $t = 20$ minutes, say. Meanwhile, the slope of the remainder is nearly 1, which implies that the system is less stiff there. Keep in mind that a slope of 1 does *not* imply that the system is *not* stiff!

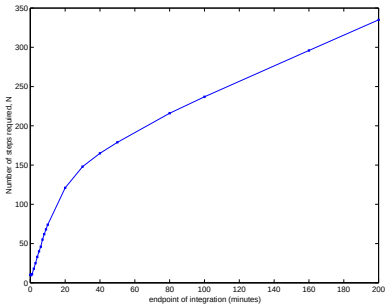

FIG. 3.6. *Relationship between T_f and the number of steps N for* `ode45`.

The possibility exists that the difference in slope between these two regions may not imply a decrease in stiffness. The large initial slope may be due to the small time step chosen by the integrator during the fast transient phase. A similar approach was taken by Lambert [7] in which the change is much more dramatic. In this example, Lambert presents a system which is not stiff for a particular interval of integration, and then becomes very stiff on a neighbouring interval. Our contention here is that the zero delay Santillán-Mackey model *is* stiff, irrespective of the interval. It is the degree to which it is stiff that is in question.

Having seen certain aspects which reveal stiffness in the zero delay system, we now look for similar features in the full Santillán-Mackey model.

3.3. The full Santillán-Mackey model. Here, we review our study of the full Santillán-Mackey model 1.1 - 1.9. To show stiffness in the system, we rely on Definition 1 of Lambert and the rapid transition through the transient phase to the invariant manifold. Unfortunately, a source of stiffness in the parameter T_{ext} is not as clear as it is for the zero delay system. However, we do find that the equations of the invariant manifolds still depend on T_{ext} and we are able to fit this dependance well.

3.3.1. Characterizing stiffness for the full model. We began our study of this system by running Experiment 1 on our fourth order RK method adapted for DDEs, `RK4d`. Using the time step $h = 0.01$ reported by Santillán and Mackey, `RK4d` nicely reproduces the figures seen in [9]. This time step seems unsatisfactory though, especially when the system approaches its steady state. The total number of steps required to reach the steady state is 20000, which amounts to 80000 function evaluations with four evaluations per step. Our hope was that this huge computational effort could be avoided with a more suitable algorithm. At

this point in the study, we discovered the MATLAB DDE solver of Shampine and Thompson, `dde23` [10]. Figure 3.7 shows that `RK4d` and `dde23` produce the same results up to graphical resolution.

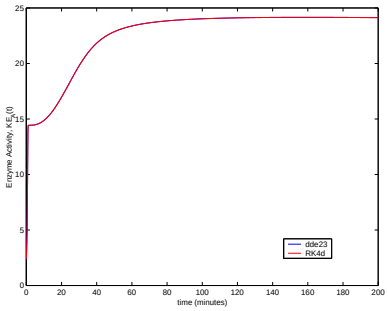

FIG. 3.7. *Experiment 1 with* `RK4d` $N = 20000$ *and* `dde23`.

As we have already seen, `dde23` is an explicit, adaptive step length method. With no options set, `dde23` completes the integration in 1523 steps, but with 1330 failed attempts, resulting in over 32000 function evaluations. It takes nearly as many failed steps as successful ones! The problem is that its estimate for the step length is often too large for the next step. It computes the step, calculates a large error and then decreases the step size to recompute the same step. This is very expensive here. To combat this, we set the maximum step length to 0.15 using the `ddeset` option `MaxStep`. Now `dde23` completes the integration in 1386 steps with only 19 failed attempts making about 8300 function evaluations. Even with this improvement, the step length of `dde23` remains rather small (see Figure 3.8). The step length is still very small after the transients have died, and the constant steady state solution appears. The time step begins around 0.025 and oscillates about 0.1 with an amplitude of about 0.05 until $t = 10$ minutes. From there it increases to its maximum value 0.15, where it remains for the rest of the integration. The step length without the `MaxStep` option behaves similarly. Here, the time step oscillates at a smaller value, and settles into a time step just below 0.14 for the majority of the integration (hence the greater number of steps taken) (see Figure 3.9). However, it peaks at certain times well above 0.15, which leads to concerns of absolute stability. We believe that with no options set, `dde23` solves the system on the

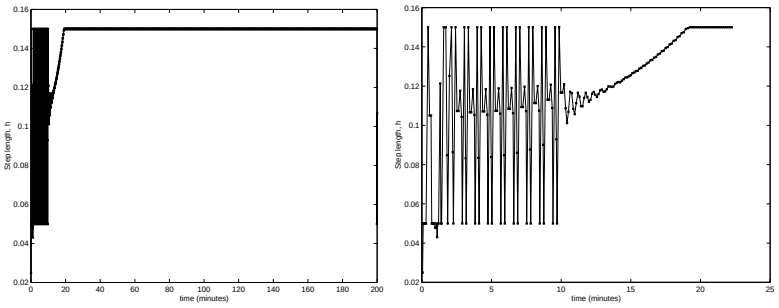

FIG. 3.8. left*: Experiment 1, step length versus time,* `dde23`*,* `MaxStep` $= 0.15$. right*: Close up for* $t \in (0, 25)$.

border of its stability region. We observe oscillations, beginning in the $M_F(t)$ variable for $T_{ext} = 0$, which become more pronounced as we increase the ratio $T_{ext}/\overline{T}$. Figure 3.10 shows quite clearly that the oscillations begin precisely when the time step shoots up from

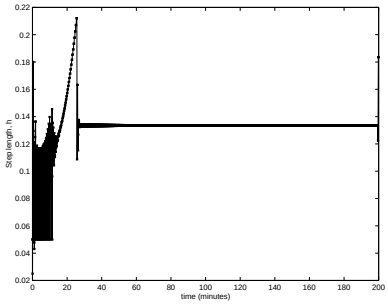

FIG. 3.9. *Experiment 1, step length versus time,* `dde23`*, no options.*

its value at 0.05 to just above 0.3 in only two steps. This occurs around $t = 2$ minutes. Finally, the solution becomes so unstable for some $T_{ext}/\overline{T} \in (54, 55]$ that the solution blows

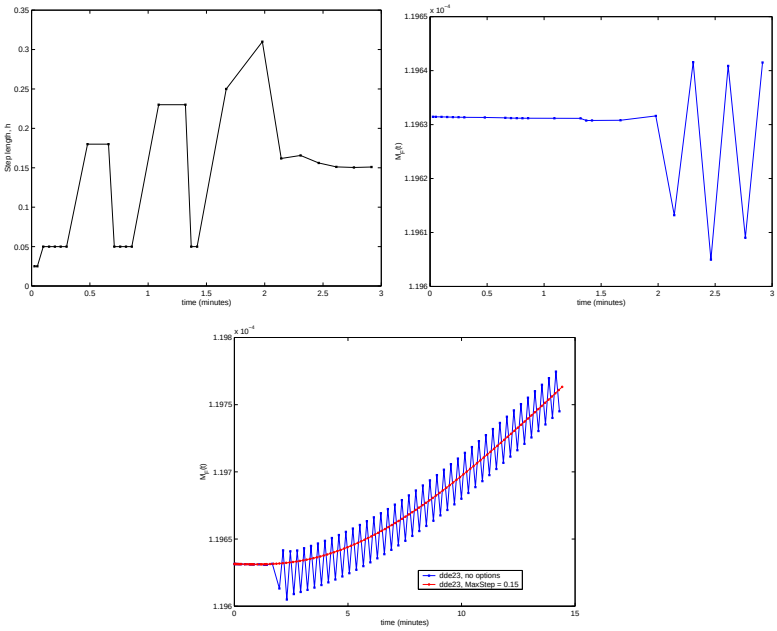

FIG. 3.10. top l*: Step length versus time, Experiment 1 with* $T_{ext} = 50\overline{T}$*,* `dde23`*, no options.* top r*:* $M_F(t)$ *versus time for the same experiment.* bottom*:* $M_F(t)$ *comparison* `MaxStep` *= 0.15 and no options.*

up at the 20th step. We observe that the step length jumps at this step - it is at precisely the same step as in the top left plot of Figure 3.10. To remedy this problem, we set the `ddeset` option `MaxStep` to 0.15, keeping the integration within the region of absolute stability. The dramatic improvement is seen in the bottom plot of Figure 3.10. The behaviour of the model with respect to T_{ext} suggests this parameter may be related to the stiffness of the system. An attempt to reproduce Figure 3.5 for the full model failed, so T_{ext} as a source of stiffness is less clear here.

Like the zero-delay system we observe a saturation of crosses in Figure 3.11. Once more, this saturation is evident even after the solution reaches its steady state.

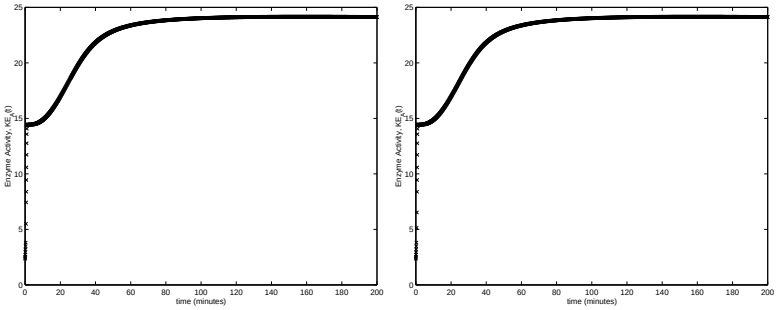

FIG. 3.11. *Experiment 1,* `dde23`. left*: no options, N=1523.* right*:* `MaxStep` $= 0.15$, $N = 1386$.

3.3.2. Invariant manifolds in the full system. As in the zero-delay system, we detected invariant manifolds in the full Santillán-Mackey model. The coupling of the variables is the same - $E(t)$ pairs up with $T(t)$ and $O_F(t)$ pairs with $M_F(t)$. The time scales, though, are slightly different. The system moves from a four dimensional manifold to a three dimensional manifold when E and T pair up somewhere around 1 or 2 minutes, depending on the initial conditions. The solution then moves from this three dimensional manifold to a two dimensional manifold anywhere from 9 to 62 minutes later, depending on the initial conditions. Figure 3.12 shows the E-T invariant manifold for four plots with different initial data. Similarly, Figure 3.13 shows the invariant O_F-M_F manifold.

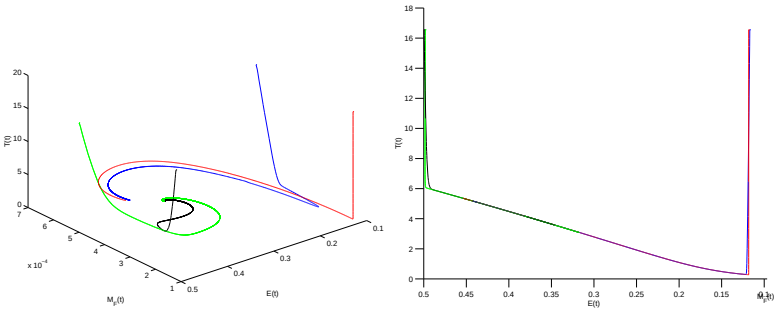

FIG. 3.12. left*: Invariant E-T manifold.* right*: Rotation in ET-plane.*

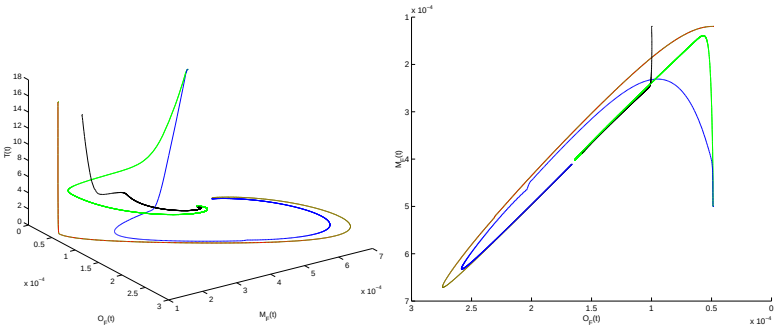

FIG. 3.13. left*: Invariant O_F-M_F manifold.* right*: Rotation in $O_F M_F$-plane.*

We found that the relationships between the variables becomes linear at these times, and

TABLE 3.3
Nonlinear fit of invariant manifold data with $Z = T_{ext}/\overline{T}$

$E(t)$-$T(t)$ plane fitting

$$m_1(Z) = 63.18\tanh(-1.440Z) - \frac{39.37}{Z} + \frac{9.548Z}{1000} + 151.1 \tag{3.6}$$

$$b_1(Z) = 2.027\tanh(\frac{7.135Z}{100}) - \frac{3.758}{Z} + \frac{1.861Z}{1000} + 3.753, \tag{3.7}$$

$M_F(t)$-$O_F(t)$ plane fitting

$$m_2(Z) = \frac{0.2001}{(2.157Z - 0.8117)} + \frac{0.01456}{Z} + \frac{4.100Z}{10000} + 2.546 \tag{3.8}$$

$$b_2(Z) = -1.210\tanh(19.41Z) - \frac{1.431}{10^5 Z} - \frac{1.682Z}{10^8} + 1.210 \tag{3.9}$$

were able to fit this relationship well for $T_{ext}/\overline{T} \in [0, 100]$. Since each relationship is linear, we determined the coefficients of the equation

$$Y(t) = m(T_{ext})X(t) + b(T_{ext}) \tag{3.5}$$

numerically. $(Y(t), X(t))$ represent $(M_F(t), O_F(t))$ and $(T(t), E(t))$, respectively. The coefficients m and b are functions of T_{ext}, or more precisely, we fitted the numerical data as functions of $Z = T_{ext}/\overline{T}$. Figure 3.14 shows the nonlinear fits in Table 3.3.

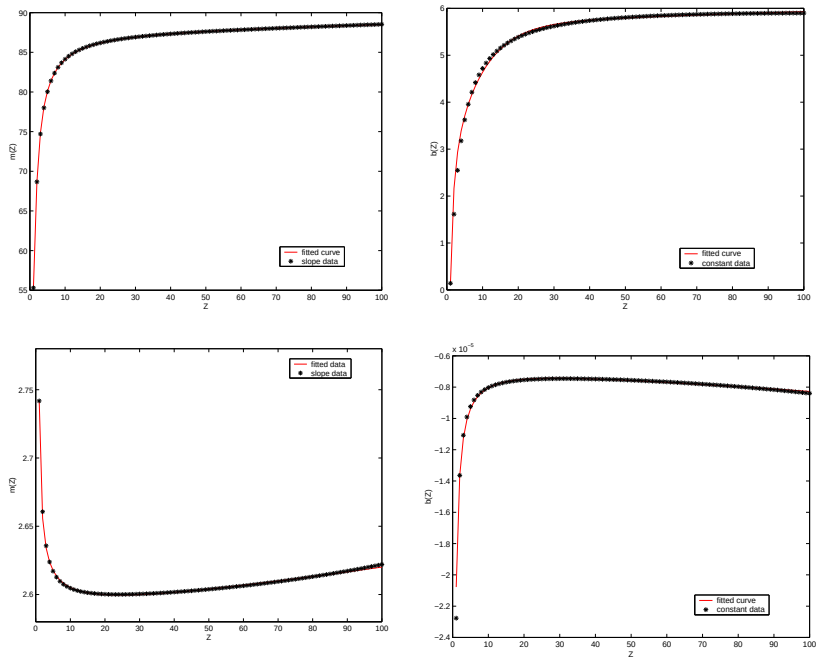

FIG. 3.14. top*: Nonlinear fit of* E-T *manifold.* top l*: slope,* $m_1(Z)$. top r*: constant,* $b_1(Z)$. bottom*: Nonlinear fit of* O_F-M_F *manifold.* bottom l*: slope,* $m_2(Z)$. bottom r*: constant,* $b_2(Z)$.

3.4. Up to here: A summary of results. We observe that the Santillán-Mackey model describes dynamics which quickly converge to a two dimensional manifold. We find that the model can be represented by the following equations,

For $0 \leq t < t_1$

$$\dot{O}_F = \frac{K_r}{K_r + R_A(T)}\{\mu O - k_p P[O_F(t) - O_F(t-\tau_p)e^{-\mu\tau_p}]\} - \mu O_F(t) \tag{1.1}$$

$$\dot{M}_F = k_p P O_F(t-\tau_m)e^{-\mu\tau_m}[1 - A(T)] - k_\rho \rho[M_F(t) - M_F(t-\tau_\rho)e^{-\mu\tau_\rho}] - (k_d D + \mu)M_F(t) \tag{1.2}$$

$$\dot{E} = \frac{1}{2}k_\rho \rho M_F(t-\tau_e)e^{-\mu\tau_e} - (\gamma + \mu)E(t) \tag{1.3}$$

$$\dot{T} = KE_A(E,T) - G(T) + F(T, T_{ext}) - \mu T(t). \tag{1.4}$$

For $t_1 < t < t_2$, $T(t) = m_1(Z)E(t) + b_1(Z)$,

$$\dot{O}_F = \frac{K_r}{K_r + R_A(T)}\{\mu O - k_p P[O_F(t) - O_F(t-\tau_p)e^{-\mu\tau_p}]\} - \mu O_F(t)$$

$$\dot{M}_F = k_p P O_F(t-\tau_m)e^{-\mu\tau_m}[1 - A(T)] - k_\rho \rho[M_F(t) - M_F(t-\tau_\rho)e^{-\mu\tau_\rho}] - (k_d D + \mu)M_F(t)$$

$$\dot{E} = \frac{1}{2}k_\rho \rho M_F(t-\tau_e)e^{-\mu\tau_e} - (\gamma + \mu)E(t).$$

For $t > t_1$, $T(t) = m_1(Z)E(t) + b_1(Z)$ and $M_F(t) = m_2(Z)O_F(t) + b_2(Z)$,

$$\dot{O}_F = \frac{K_r}{K_r + R_A(T)}\{\mu O - k_p P[O_F(t) - O_F(t-\tau_p)e^{-\mu\tau_p}]\} - \mu O_F(t)$$

$$\dot{E} = \frac{1}{2}k_\rho \rho M_F(t-\tau_e)e^{-\mu\tau_e} - (\gamma + \mu)E(t).$$

The functional values of m_1, b_1, m_2 and b_2 are given in Table 3.3.

4. Conclusion. We wish to examine these dynamics in more detail from a biological perspective. Recall that Experiment 1 is designed to take a bacterial culture from a medium of high external tryptophan concentration and put it into a medium of low external tryptophan concentration. Biologically, we see that the total anthranilate synthase concentration (E) and the tryptophan concentration (T) rapidly reach a state in which their rates of change are directly proportional. Running the first part of Experiment 1 with $T_{ext} = 400\overline{T}$, we find that the tryptophan concentration reaches a high steady state value, while the steady state enzyme concentration (E) is small, as are steady states of the free operon (O_F) and free mRNA (M_F) concentrations. Switching to the minimal medium, with $T_{ext} = 0$ (or a medium with $T_{ext} < 400\overline{T}$), the tryptophan concentration (T) decreases remarkably quickly, yet the anthranilate synthase concentration (E) remains relatively unchanged. It is likely that any tryptophan bound to anthranilate synthase causing inhibition is released during this time, after which we begin to see a direct correlation between the enzyme concentration and the tryptophan concentration. As the catalysts' rate constant for tryptophan synthesis is large, $K = 126.4\ \text{min}^{-1}$, we expect the tryptophan concentration to be very sensitive to the enzyme concentration. Any enzyme freed from inhibition may instantly begin to catalyze the reaction to make tryptophan. Meanwhile, the total delay between mRNAP binding at the promoter and enzyme production is the sum of all the delays $\tau_p + \tau_m + \tau_\rho + \tau_e = 0.91$ minutes. This is approximately the time we see the solution reach the first invariant manifold, at about 1 minute. Though this may be coincidental, it seems unlikely. It remains a question for further study: an appropriate test of this hypothesis involves changing the delays such that their sum is not 0.91 min and observing the time at which the solution reaches the first invariant manifold.

The solution reaches the second invariant manifold tens of minutes after reaching the first invariant manifold. Here the free operon concentration (O_F) and (M_F) become linearly

related. Speculation as to the tens of minutes taken to reach this manifold may be that the tryptophan binding to the repressor molecules is particularly strong. From the estimated parameter $K_t = 60.3\ \mu$M, we see that this *is* the largest of the reaction rate constants, validating this hypothesis. From a biological perspective, strong binding here is beneficial: transcribing DNA into RNA and making proteins are energetically expensive cellular functions. If the binding of tryptophan to the repressor is not particularly strong, then minor fluctuations in the tryptophan concentration may cause the cell to produce enzyme unnecessarily, and expend a lot of energy doing it. Again, it is not surprising that the free operon concentration and the free mRNA concentration should be related. After all, there is little in the way of the mRNAP binding a free operon and making an mRNA molecule. Interestingly, however, we see that the free mRNA concentration is about 2.6 times *larger* than the free operon concentration on the invariant manifold. This is also observed in the steady state conditions in Table 3.1, where $\overline{M_F}/\overline{O_F} \approx 2.45$. Presumably, the reason that the free mRNA concentration is larger than the free operon concentration is that several mRNAP molecules can transcribe mRNA molecules at the same time, freeing the mRNA even during transcription. We may hypothesize, given that the slope of the O_F-M_F manifold is consistently around 2.6, that approximately 2-3 mRNAP molecules transcribe the anthranilate synthase gene on the DNA before freeing up the promoter region for the next mRNAP. It is quite possible that this is a saturation level. With these thoughts, we have yet another prediction of the model which may be experimentally tested.

REFERENCES

[1] MATLAB 6. The Math Works Inc., 3 Apple Hill Dr., Natick, MA 01760, 2001. computer software.

[2] R.D. Bliss and A.G. Marr. "Role of feedback inhibition in stabilizing the classical operon". *J. Theor. Biol.*, 97, 1982.

[3] P. Bogacki and L.F. Shampine. "A 3(2) pair of Runge-Kutta formulas". *Appl. Math. Lett.*, 2, 1989.

[4] M. Caberlin. Stiff ordinary and delay differential equations in biological systems. Master's thesis, McGill University, July 2002.

[5] B. Goodwin. "Oscillatory behaviour in enzymatic control process". *Adv. Enz. Regul.*, 3, 1965.

[6] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving ordinary differential equations. I.* Springer-Verlag, Berlin, second edition, 1993. Nonstiff problems.

[7] J.D. Lambert. *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*. John Wiley & Sons, Toronto, 1991.

[8] M. Santillán and M.C. Mackey. "Dynamic behaviour in mathematical models of the tryptophan operon". *Chaos*, 11(11), 2001.

[9] M. Santillán and M.C. Mackey. "Dynamic regulation of the tryptophan operon: A modeling study and comparison with experimental data". *PNAS*, 98(4), 2001.

[10] L. F. Shampine and S. Thompson. Solving DDEs in MATLAB. *Appl. Numer. Math.*, 37(4):441–458, 2001.

[11] L.F. Shampine. *Numerical Solution of Ordinary Differential Equations*. Chapman & Hall, New York, 1994.

[12] S. Sinha. "Theoretical study of tryptophan operon: Application in microbial technology". *Biotechnol. Bioeng.*, 31, 1988.